

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

на тему: «Розробка підсистеми торгів електронної біржі»

САДИКОВ ПАВЛО ВАСИЛЬОВИЧ

(прізвище, ім'я та по батькові студента повністю)

Київ 2023 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

„___” _____ 2023 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

на тему: " Розробка підсистеми торгів електронної біржі "

Виконав: студент II-го курсу, групи КНм-II

Спеціальності: 122 «Комп'ютерні науки» .

(шифр і назва напрямку підготовки, спеціальності)

Садиков П.В. .

(прізвище та ініціали)

Керівник к.т.н., доц. Шабала Є.Є. .

(прізвище та ініціали)

Рецензент к.т.н., доц. Баліна О.І. .

(прізвище та ініціали)

Київ, 2023 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій

Освітній рівень: «магістр за ОПП»

Спеціальність: 122. «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

„___” _____ 2023 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

_____ Садиков Павло Васильович _____

1. Тема роботи: Розробка підсистеми торгів електронної біржі
затверджена наказом ректора КНУБА № _____ від «_____»
2023 р.
2. Керівник роботи: Шабала Євгенія Євгенівна, к.т.н, доцент
кафедри КБКІ
3. Строк подання студентом роботи до захисту: грудень 2023 рік
4. Зміст пояснювальної записки за розділами:
 - P.1. Аналіз предметної області та постановка задачі
 - P.2. Концептуальне та функціональне призначення
 - P.3. Проектування інформаційного забезпечення
 - P.4. Алгоритмічне забезпечення системи
 - P.5. Тестовий приклад програми
5. Інформаційні слайди:
 - C.1 Розробка архітектури системи
 - C.2. Розробка БД системи
 - C.3. Дерево цілей, дерево функцій

С.5. Програмне забезпечення системи. Тестовий приклад програми

6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області та постановка задачі	Вересень 2023 р.
Р. 2. Концептуальне та функціональне призначення	Жовтень 2023 р.
Р. 3. Проектування інформаційного забезпечення	Жовтень 2023 р.
Р. 4. Алгоритмічне забезпечення системи	Листопад 2023 р.
Р. 5. Тестовий приклад програми	Листопад 2023 р.
Остаточне оформлення роботи	Грудень 2023 р.
Направлення роботи на рецензування, перевірку на плагіат	02 грудня 2023 р.
Попередній захист роботи на кафедрі	05 грудня 2023 р.

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		дата	підпис
Розділ 1.	к.т.н., доц. Баліна О.І.	05.12.2023 р.	
Розділ 2.			
Розділ 3.			
Розділ 4.			
Розділ 5.			
Розділ 6.			

8. Дата видачі завдання: 05 вересня 2023 року

Керівник

(підпис)

Шабала Є.Є.

(прізвище та ініціали)

Магістрант

(підпис)

Садиков П.С.

(прізвище та ініціали)

АНОТАЦІЯ

Садиков П.С. «Розробка підсистеми торгів електронної біржі».

Атестаційна випускна робота магістра за спеціальністю: 122. «Комп'ютерні науки» – Київський національний університет будівництва та архітектури. – Київ, 2023.

Основними причинами розвитку внутрішньої торгівлі товарами є: здійснення структурних економічних змін у бік посилення конкуренції та зменшення частки тіньового сектора; законодавчі зміни в регулюванні біржової діяльності, що сприяють уніфікації бірж, пропозиції ними сучасних послуг, уніфікації біржових товарів, розвитку електронної біржової торгівлі; активізація діяльності бірж для залучення більшої кількості клієнтів за рахунок збільшення доступності електронної біржової торгівлі, організації торгівлі деривативами, надання гарантій виконання контрактів.

Ключові слова: інформаційна система, база даних, функціональне та концептуальне призначення системи.

SUMMARY

Sadikov P.S. "Development of the trading subsystem of the electronic exchange".

Master's attestation thesis in the specialty: 122. "Computer Science" - Kyiv National University of Construction and Architecture. - Kyiv, 2023.

The main reasons for the development of domestic trade in goods are: implementation of structural economic changes in the direction of increased competition and reduction of the share of the shadow sector; legislative changes in the regulation of exchange activity, which contribute to the unification of exchanges, their offer of modern services, unification of exchange goods, development of electronic exchange trading; activation of stock exchanges to attract more customers by increasing the availability of electronic stock trading, organizing derivatives trading, providing guarantees of contract performance.

Keywords: information system, database, functional and conceptual purpose of the system.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Аналіз предметної області.....	10
1.1.1 Загальна характеристика електронної біржі	10
1.1.2 Характеристика електронної торгової платформи.....	11
1.1.3 Опис діючої інформаційної системи	15
1.1.4 Опис недоліків діючих торгових платформ.....	17
1.1.5 Обґрунтування необхідності розширення діючої системи торгівлі	18
1.2 Загальні вимоги до підсистеми торгів.....	20
1.3 Постановка задачі.....	22
РОЗДІЛ 2 РОЗРОБКА АРХІТЕКТУРИ ПІДСИСТЕМИ ТОРГІВ	24
2.1 Дерево цілей системи.....	25
2.2 Дерево функцій системи.....	28
2.3 Функціональні моделі	31
2.4 DFD діаграми потоків даних.....	36
РОЗДІЛ 3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ ЕЛЕКТРОННОЇ БІРЖІ	42
3.1 Визначення сутностей БД системи.....	43
3.2 Концептуальна модель бази даних	45
3.3 Логічна модель бази даних.....	46
3.3 Фізична модель бази даних	49
РОЗДІЛ 4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПІДСИСТЕМИ ТОРГІВ ЕЛЕКТРОННОЇ БІРЖІ.....	58
4.1 Вибір сучасних інформаційних технологій	59
4.2 Архітектура програми та опис класів і модулів	64
4.3 Алгоритм виконання правил.....	71
РОЗДІЛ 5 ПРИКЛАД ВИКОРИСТАННЯ ПІДСИСТЕМИ ТОРГІВ ЕЛЕКТРОННОЇ БІРЖІ.....	74
5.1 Розгляд роботи GUI підсистеми	75
5.2 Розгляд та приклад роботи Rules Script.....	88
СПИСОК ЛІТЕРАТУРИ	

ВСТУП

Інформаційні технології є найбільш важливою складовою процесу використання інформаційних ресурсів суспільства. До теперішнього часу вони пройшли кілька еволюційних етапів, зміна яких визначалася головним чином рівнем розвитку науково-технічного прогресу, появою більш досконалих засобів обробки інформації. Сучасний науково-технічний прогрес неможливо уявити без широкого впровадження і використання обчислювальної техніки в освіті, виробництві, управлінні та наукових дослідженнях. Сьогодні на основі засобів обчислювальної техніки розробляються і впроваджуються різні автоматизовані інформаційні системи.

Метою даної роботи є розробка підсистеми торгів електронної біржі – так званої електронної торгової площадки. Ця система дозволить проводити торги на електронних біржах шляхом їх розширеного функціоналу:

- Створення відкладених ордерів
- Додавання правил та умов продажу/купівлі
- Відображення «стакану» біржі з загальною сумою
- Підрахунок комісії та цін беззбитку
- Створення ордерів з визначеним прибутком
- Можливість використання декількох бірж
- Калькулятор прибутку
- Загальне підвищення зручності користування елементами біржі
- Локалізація на українській мові

Вигода від створення такої підсистеми полягає в тому, що користувач матиме змогу більш ефективно розпоряджатись своїми грошима та часом.

У першому розділі проводиться аналіз предметної області, розглядається діюча інформаційна система торгів, її недоліки. Описується поставка задачі, визначаються загальні цілі та функції.

У другому розділі проводиться розробка підсистеми торгів, розглядається «дерево цілей» та «дерево функцій», також будуються IDEF0 моделі та DFD діаграми потоків даних.

У третьому розділі проектується база даних електронної біржі, а саме складається список сутностей та атрибутів, створюється концептуальна, даталогічна та фізичні моделі бази даних.

У четвертому розділі розглядаються застосовані у розробці підсистеми інструментальні засоби, структура системи, а також опис основних класів та алгоритмів.

У п'ятому розділі розглядається приклад роботи підсистеми торгів електронної біржі, створюється новий профіль користувача, розглядаються елементи програми, створення ордерів та правил виконання.

У шостому розділі виконано розрахунок кошторисної вартості розробки програмного продукту.

У сьомому розділі приводяться заходи, які необхідно провести для забезпечення охорони працюючих з даним програмним продуктом.

**РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ
ТА ПОСТАНОВКА ЗАДАЧІ**

1.1 Аналіз предметної області

1.1.1 Загальна характеристика електронної біржі

Електронна біржа (*e-біржа, e-exchange*) - Купівля-продаж різноманітних товарів на біржі в межах електронного бізнесу, коли є багато продавців і багато покупців. Зацікавлений в конкретних пропозиціях клієнт, вигравши контракт, перераховує продавцю платню і одержує потрібний продукт протягом обумовленого відповідного терміну. Історія електронної біржової торгівлі нараховує кілька десятиліть. Це одне з новітніх явищ у розвитку техніки проведення операцій на міжнародних товарних біржах. По суті, це принципово новий підхід до процесу ведення біржового торгу.

Інтернет-біржа - це насамперед купівля-продаж різних товарів на біржі в рамках електронного бізнесу, коли є багато продавців і багато покупців. Зацікавлений у конкретному реченні клієнт, вигравши контракт, переводить продавцеві плату і отримує потрібний продукт протягом певного часу. І, відповідно, Інтернет-біржа - біржа, ведуча торги з використанням інформаційної мережі. Операції купівлі продажу здійснюються користувачами за допомогою абонентських систем, включених в інформаційну мережу. Аж до недавнього часу комп'ютерна технологія використовувалася біржами тільки для висвічування на електронних табло і передачі по засобах зв'язку біржових цін, що формуються в «кільці», і цін на біржові товари найбільших дилерів, а також для обліку біржових угод та контролю і регулювання портфеля активів у рамках самих фірм-членів біржі.

Предметами електронної біржової торгівлі є, в основному, фінансові контракти, тобто ф'ючерси та опціони на валюти, фондові індекси, процентні ставки і т.п. В нашому випадку торги на біржах проводяться між звичайними, фіатними грошима (USD, EUR) та електронними (bitcoin, litecoin тощо).

Bitcoin - це мережа глобального консенсусу, яка створює нову платіжну систему і повністю цифрові гроші. Це перша децентралізована р2р платіжна мережа, яка обслуговується її ж користувачами, без центральних органів управління або посередників. З точки зору користувачів, біткойн дуже схожий на готівку для інтернету.

Основне завдання інтернет біржі - забезпечити можливість використання Інтернету як зручного та доступного інструменту, реально допомагає здійснювати професійну діяльність. В даний час в світі діють самі різні системи електронної біржової торгівлі, однак всі ці системи будуються, по суті, за однією і тією ж схемою: центральний комп'ютер з розгалуженою мережею, що забезпечує роботу певної кількості терміналів. Будь електронна біржова система включає в себе три головні компоненти:

1. Робоча станція члена біржі - учасника торгівлі (трейдера) (Trader Station) - за допомогою неї брокери беруть участь у торгівлі, отримуючи ринкову інформацію і вводячи в систему свої накази на покупку або продаж; В даній роботі розробляється саме ця частина електронної біржі.

2. Лінія зв'язку (Communications Link) - засіб для здійснення діалогу між брокером і центральною біржею;

3. Центральна біржова система (Central Exchange System) - забезпечує зведення всієї ринкової інформації та виконання наказів брокерів.

1.1.2 Характеристика електронної торгової платформи

Електронна торгівельна платформа - комп'ютерна система, яка може використовуватися, щоб розмістити замовлення (купівлю / продаж) фінансових продуктів, таких як акції, облігації, валюти, біржовий товар та деривативи з фінансовим посередником, такими як брокери, дилери, маркетмейкери або фондові біржі по мережі . Такі платформи дозволяють

використовувати електронну торгівлю користувачами з будь-якого місця розташування на відміну від традиційної кімнатної торгівлі (Floor trading) і торгівлі, що базується на телефонній лінії.

В даній роботі розглядатимуться три електронні платформи:

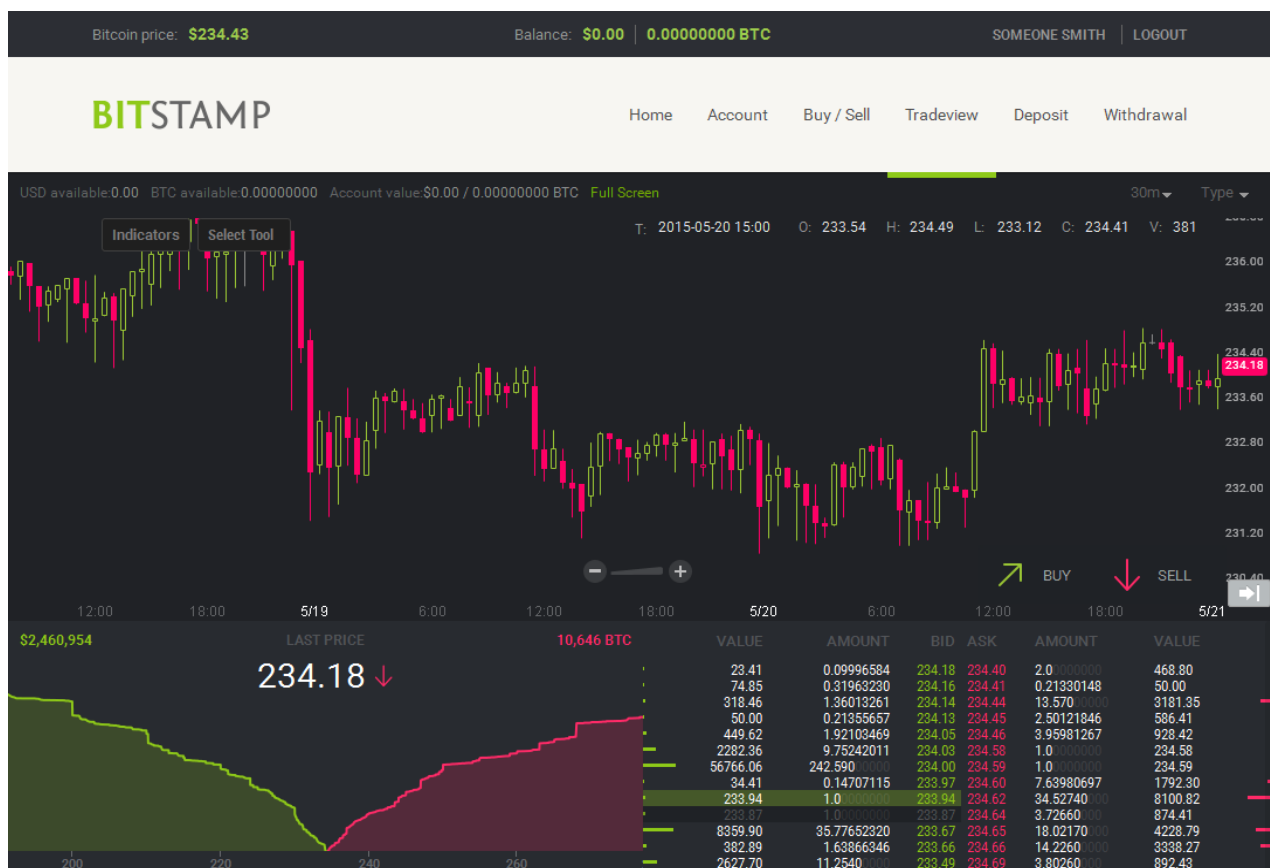



Рисунок 1.1 bitstamp.net



BITFINEX
INVEST IN THE FUTURE

Sapien ⚙

<p>Bitcoin last price: \$273.99 Daily Change: 0.25 (0.09%)</p>	<p>Litecoin last price: \$1.9098 Daily Change: 0.0058 (0.3%)</p>	<p>Darkcoin last price: \$3.3999 Daily Change: -0.0161 (-0.47%)</p>
--	--	---

YOUR BALANCE		\$1,064.51
deposit	exchange	Total
0.00 USD	1,064.51 USD	1,064.51 USD

Exchange
Total Return Swaps

BTCUSD
LTCUSD
LTCBTC
DRKUSD
DRKBTC

My active orders

#	Pair	Type	Original Amount	Amount	Price	Placed	Status	Cancel all
No active orders.								

▼ 273.99

Order Type

Limit

273.99

BUY

Order size (BTC)

How much?

Limit

273.99

Buy all

Hide order

Sell all

▼ 273.93

Order Type

Limit

273.93

SELL

Order size (BTC)

How much?

Limit

273.93

Buy all

Hide order

Sell all

Order Book

Account overview

exchange

USD
1,064.511

Deposit

Withdraw

Manage Wallets

API access

Market Statistics

Bids				+ / -	Asks			
Order Count	Bid Price	Amount	Total Amount		Total Amount	Amount	Ask Price	Order Count
1	\$273.9	12.38	12.38		0.57	0.57	\$274.0	1
4	\$273.8	60.16	72.54		5.76	5.19	\$274.1	3
5	\$273.7	17.82	90.36		11.23	5.47	\$274.2	3
2	\$273.5	7.51	97.86		24.03	12.79	\$274.3	3
1	\$273.3	4.1	101.97		43.86	19.83	\$274.4	4
3	\$273.2	30.67	132.63		93.01	49.15	\$274.5	7
9	\$273.1	79.51	212.15		149.29	56.28	\$274.6	7
4	\$273.0	45.71	257.85		217.8	68.5	\$274.7	6
4	\$272.9	53.88	311.73		280.8	63.0	\$274.8	7
4	\$272.8	84.19	395.92		367.64	86.84	\$274.9	7
4	\$272.7	40.0	435.92		479.75	112.11	\$275.0	7
3	\$272.6	53.94	489.86		520.81	41.06	\$275.1	6
5	\$272.5	47.67	537.53		601.93	81.12	\$275.2	6
3	\$272.4	30.0	567.53		668.05	66.12	\$275.3	7
5	\$272.3	61.8	629.32		713.05	45.0	\$275.4	4
2	\$272.2	1.25	630.57		782.96	69.91	\$275.5	6
6	\$272.1	121.87	752.44		827.71	44.75	\$275.6	6
3	\$272.0	84.35	836.79		872.71	45.0	\$275.7	4
1	\$271.9	4.52	841.31		922.63	49.93	\$275.8	6
3	\$271.8	5.03	846.33		949.72	27.08	\$275.9	6
1	\$271.7	0.03	846.36		959.72	10.0	\$276.0	1

[Home](#)
[Announcements](#)
[Security](#)
[How it works](#)
[Verification](#)
[Support](#)
[Our fees](#)
[Our API](#)

Copyrights © 2013 - 2015 Bitfinex
[Terms and conditions](#)

Рисунок 1.2 bitfinex.com

BTC

Last Price: **272.073 USD** Low: **269.2219 USD** High: **276.5369 USD**
 Volume: **5373 BTC / 1466255 USD** Server Time: **08.03.15 17:22**

сapten сообщ. (0) | уведомл. (0)
 Профиль | Финансы | Редактировать | Выйти
790.8228 USD **0 BTC**

Торги
Новости
Правила
FAQ
РАММ
Support

Новости проекта:

05/03/15 Снижение комиссии на пополнение счета через банковской перевод

19/01/15 Закрытие торгов по TRC, FTC, XPM.

BTC/USD	BTC/RUR	BTC/EUR	BTC/CNH	BTC/GBP	LTC/BTC	LTC/USD	LTC/RUR	LTC/EUR	LTC/CNH
272.073	16295	268.7	1483	172.5	0.00882	0.008999	113.3	174	11.62
LTC/GBP	NMC/BTC	NMC/USD	NVC/BTC	NVC/USD	USD/RUR	EUR/USD	EUR/RUR	USD/CNH	GBP/USD
1.322	0.00181	0.487	0.00251	0.684	65	1.09888	65.28	6.082	1.546
PPC/BTC	PPC/USD								
0.00143	0.392								

Купить BTC

Ваши средства: **790.8228 USD**

Мин Цена: **272.066 USD**

Количество BTC:

Цена за BTC: USD

Всего: **0 USD**

Комиссия: **0 BTC**

Нажмите **подсчитать**, чтобы рассчитать сумму в соответствии с ордерами.

Продать BTC

Ваши средства: **0 BTC**

Макс Цена: **272.066 USD**

Количество BTC:

Цена за BTC: USD

Всего: **0 USD**

Комиссия: **0 USD**

Нажмите **подсчитать**, чтобы рассчитать сумму в соответствии с ордерами.

Ордера на продажу Всего: 11116.05 BTC

Цена	BTC	USD
272.066	0.00278786	0.75853767
272.098	0.3699234	84.05743929
272.221	0.0202	5.4988542
272.231	0.0101	2.7495231
272.255	0.013972	3.80394686
272.261	0.0101	2.7498361
272.282	0.01497	4.07606154
272.284	0.010978	2.98913375
272.309	0.0385	10.4838965
272.31	1.06127221	288.9950355
272.317	0.010978	2.98949602
272.321	0.0505	13.7522105
272.326	0.07391575	20.12918053
272.335	0.013972	3.80506462
272.391	0.0696182	18.96337111
272.392	0.0566998	15.44457192
272.393	0.01497	4.0772321
272.425	0.013972	3.8063221
272.482	0.01497	4.0795554
272.487	1.49	406.00563
272.494	3	817.482

Ордера на покупку Всего: 3088078.41 USD

Цена	BTC	USD
272.066	0.011	2.992726
272.064	0.253735	69.03215904
272.063	0.11629544	31.63968629
272.046	0.011	2.992506
272.02	14.415	3921.1683
271.7	0.8990395	244.26903215
271.446	0.099	26.873154
271.436	8.7070547	2363.40809954
271.366	0.101	27.407966
271.364	0.101	27.407764
271.012	0.28029755	75.96399962
271	0.13314575	36.08249825
270.908	0.01109059	3.00452955
270.9	0.18725	50.726025
270.804	0.01838	4.97737752
270.7	2.9374	795.15418
270.595	0.01109059	3.0010582
270.53	0.20299078	54.91509571
270.522	0.1005	27.187461
270.521	0.01	2.70521
270.5	3.08702428	835.04006774

Ваши активные ордера:

Тип	Цена	Кол-во BTC	Всего USD	Дата	Действия
Нет активных ордеров в данный момент.					

Комиссия на сделки составляет 0.2%.

История сделок:

Дата	Тип	Цена	Кол-во (BTC)	Всего (USD)
08.03.15 17:22	Buy	272.086 USD	0.00812114 BTC	2.2340832 USD
08.03.15 17:22	Buy	272.078 USD	0.011 BTC	2.992858 USD
08.03.15 17:22	Buy	272.073 USD	0.011 BTC	2.992803 USD
08.03.15 17:22	Buy	272.07 USD	0.011 BTC	2.99277 USD
08.03.15 17:21	Buy	272.073 USD	0.011 BTC	2.992803 USD
08.03.15 17:20	Sell	272.063 USD	0.005 BTC	1.360115 USD
08.03.15 17:20	Sell	272.063 USD	0.014 BTC	3.808882 USD
08.03.15 17:17	Buy	271.83 USD	9.795 BTC	2662.57485 USD
08.03.15 17:17	Sell	271.7 USD	0.03 BTC	8.151 USD
08.03.15 17:17	Sell	271.94 USD	1 BTC	271.94 USD
08.03.15 17:16	Buy	271.94 USD	0.183864 BTC	49.9997616 USD
08.03.15 17:16	Buy	271.891 USD	0.011 BTC	2.990801 USD
08.03.15 17:16	Sell	271.7 USD	0.00097 BTC	0.263549 USD
08.03.15 17:16	Sell	271.793 USD	0.011 BTC	2.989733 USD
08.03.15 17:16	Sell	272 USD	0.16796 BTC	45.44032 USD
08.03.15 17:16	Sell	272.001 USD	0.0257918 BTC	7.0159859 USD
08.03.15 17:15	Buy	272.03 USD	0.349444 BTC	95.1136732 USD
08.03.15 17:15	Sell	272 USD	0.01197 BTC	3.25584 USD
08.03.15 17:15	Sell	272.033 USD	0.011 BTC	2.992363 USD
08.03.15 17:14	Buy	272.098 USD	0.01 BTC	2.72098 USD
08.03.15 17:14	Buy	272.098 USD	0.055028 BTC	14.97300874 USD

Copyright © 2011-2013 BTC-e.com. All rights reserved.
 Public API | Trade API | Support

Рисунок 1.3 btc-e.com

1.1.3 Опис діючої інформаційної системи

У кваліфікаційній роботі, підсистема торгової платформи розглядається на прикладі двох електронних бірж. В подальшому можливе додавання будь-якої кількості площадок по мірі їх утворення та розвитку.

Діюча система торгів на площадці btc-e.com не є достатньо ефективною. Вона надає дуже обмежений функціонал(рисунок 1.4).

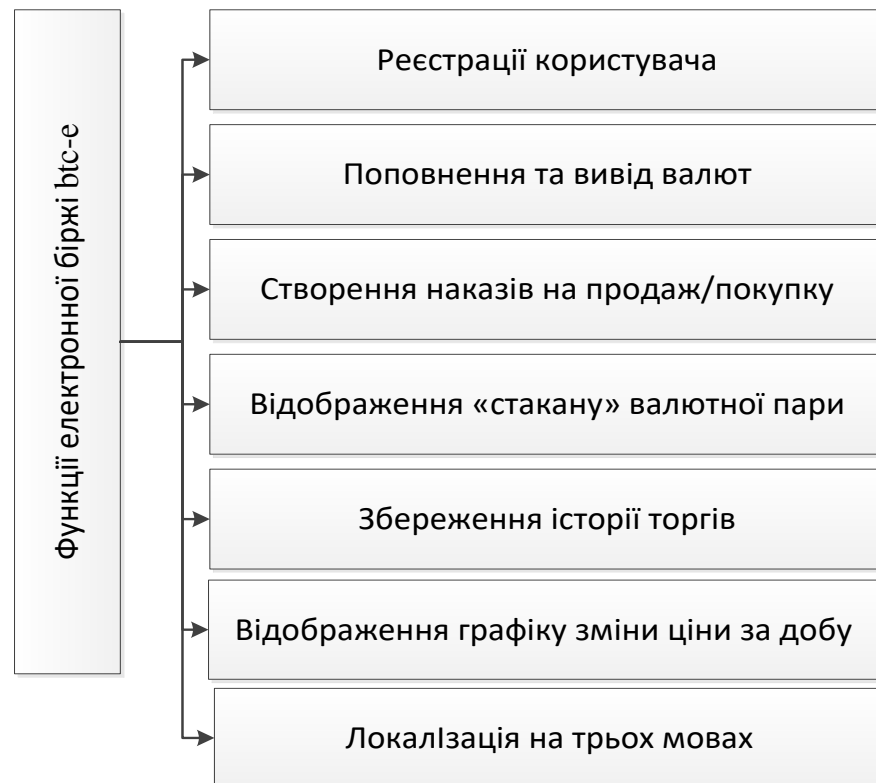


Рисунок 1.4 Функції біржі btc-e.com

В свою чергу торгова площадка bitfinex.com надає такий функціонал:

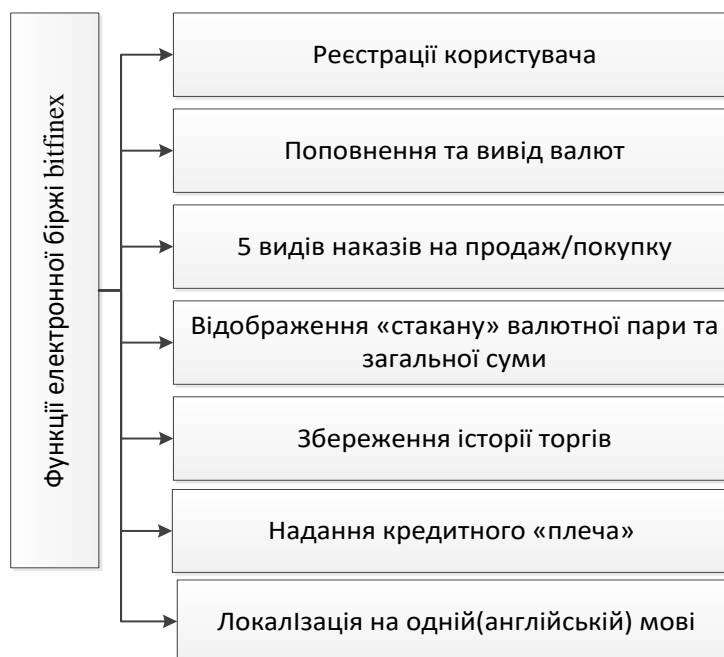


Рисунок 1.5 Функції біржі bitfinex.com

Функціонал bitstamp.net:

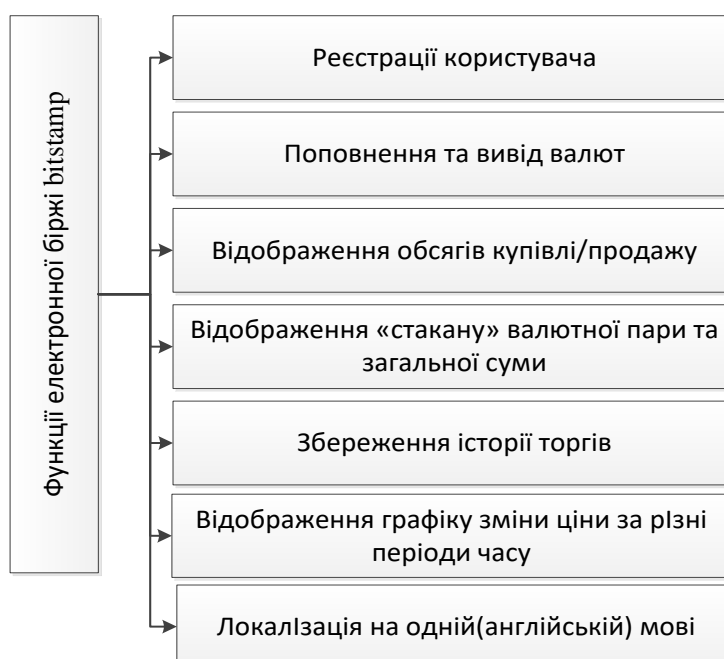


Рисунок 1.6 Функції біржі bitstamp.net

1.1.4 Опис недоліків діючих торгових платформ

Діюча інформаційна система торгів має багато недоліків, які в основному пов'язані з мінімальним функціоналом бірж та неможливість додаткових автоматичних розрахунків для ефективної торгівлі.

Основні недоліки існуючої системи btc-e такі:

- Неможливість створення відкладених ордерів(наказів)
- Відсутність:
 - 1) Створення правил та умов продажу/купівлі
 - 2) Підрахунок комісії та цін без збитку
 - 3) Калькулятора прибутку
 - 4) Створення ордерів з визначеним прибутком
 - 5) Локалізації на українській мові
- Відображення неповного «стакану» біржі
- Загальна незручність користування елементами біржі

Основні недоліки існуючої системи bitfinex такі:

- Локалізація тільки на англійській мові
- Відсутність:
 - 1) Графічного відображення зміни ціни
 - 2) Створення правил та умов продажу/купівлі
 - 3) Створення ордерів з визначеним прибутком
 - 4) Калькулятора прибутку

Основні недоліки існуючої системи bitstamp такі:

- Локалізація тільки на англійській мові
- Відсутність:
 - 1) Створення правил та умов продажу/купівлі
 - 2) Підрахунок комісії та цін без збитку
 - 3) Калькулятора прибутку
 - 4) Створення ордерів з визначеним прибутком
 - 5) Локалізації на українській мові
- Відображення неповного «стакану» біржі

1.1.5 Обґрунтування необхідності розширення діючої системи торгівлі

Діючі системи електронної торгівлі хоча й надають якісний базовий функціонал для користувача, але для щоденної та більш ефективної роботи з ними цього не зовсім достатньо. Відсутність більшості аналітичних функцій можуть не задовольняти усіх потреб користувача.

Використання сучасних інформаційних технологій для створення торгової платформи може вирішити всі ці недоліки, підвищити ефективність та якість користувацького інтерфейсу, а також збільшити її функціональність.

Удосконалення діючої системи торгів, зокрема створення єдиного функціоналу, інтерфейсу для декількох бірж поліпшить умови праці безпосередньо користувачів. Всі операції будуть проводитись через одну програму, яка забезпечить:

- Захист інформації користувача;
- Можливість швидкого відстеження всіх проведених операцій;

- Моментальне створення ордерів по назначеним параметрам, що дасть можливість швидко реагувати на ситуацію на біржі;
- Створення спеціальних правил, які автоматично виконуватимуться без участі користувача;
- Можливість розрахунку прибутковості на основі декількох значень;
- Налаштування звукових оповіщень щодо зміни ціни або виконання ордерів чи правил;
- Можливість відстеження об'єму торгів за останній період часу;
- Розташування всіх елементів програми у довільному порядку/будь-якому місці на екрані (за допомогою «відкріплення») для забезпечення зручного користування у кожному окремому випадку;
- Загальне підвищення користувацького досвіду роботи завдяки доданим елементам управління та вводу даних.

На ринку програмного забезпечення відсутні типові рішення, які б врахували особливості роботи з електронними біржами. В свою чергу, більшість таких платформ зосереджені на представленні легких та безпечних методів зачислення/зняття коштів, забезпеченні інформаційною безпекою для усіх користувачів, і тільки в останню чергу зручність та функціонал. Проте усі вони мають відкрите API для сторонніх розробників і тому постає питання в розробці та впровадженні такої підсистеми торгів.

1.2 Загальні вимоги до підсистеми торгів

У процесі виконання роботи потрібно обстежити об'єкт автоматизації та сформулювати вимоги до системи у відповідності зі специфікою виконуваних в торгівлі задач. Необхідно розглянути основні задачі, функції та класи інформаційної системи та виділити специфічні ознаки майбутньої підсистеми торгів.

Важливим в даному випадку є питання технологій розробки торговою платформи. Поставлена задача дослідження сучасних напрямків розвитку методологій побудови систем даного типу та визначення й обґрунтування технологій для розробки нашої торгової системи. Повинні бути чітко сформувані вимоги до майбутнього проекту та виділені його специфічні ознаки та особливості, визначена його структура.

Будь-яка, у тому числі розроблювана нами, автоматизована система повинна відповідати вимогам, які зазначені нижче(рис 1.5).

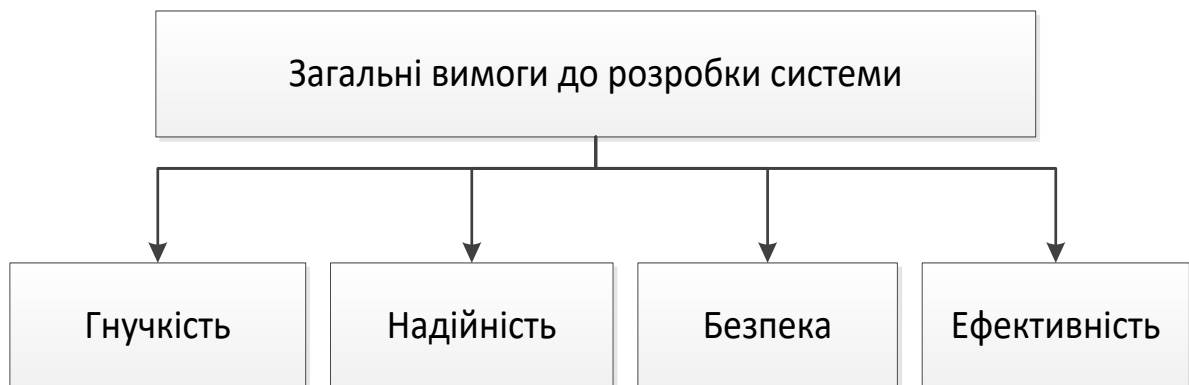


Рисунок 1.5 Загальні вимоги до розробки

Властивість гнучкості (здатності до адаптації і подальшого розвитку) вказує на можливість пристосування інформаційної системи до нових умов, нових потреб користувача. Витримка цієї властивості можливо, якщо на етапі розробки інформаційної системи використовувалися загальноприйняті засоби й методи документування, так що після певного часу збережеться можливість розібратися в структурі системи та внести до неї відповідні зміни, навіть

якщо всі розробники або їх частина з яких-небудь причин не зможуть продовжити роботу.

Надійність автоматизованої системи має на увазі її функціонування без спотворення інформації, втрати даних з «технічних причин». Вимога надійності забезпечується створенням резервних копій інформації, що зберігається, виконання операцій протоколювання, підтриманням якості каналів зв'язку та фізичних носіїв інформації, використання сучасних програмних і апаратних засобів. Сюди ж слід віднести захист від випадкових втрат інформації внаслідок недостатньої технічної грамотності користувача.

Ефективність системи забезпечується оптимізацією даних і методів їх обробки, застосуванням оригінальних розробок, ідей, методів проектування.

Не слід забувати про те, що працювати із системою будуть не лише фахівці, але й звичайні люди, які також хочуть зайнятись електронною торгівлею, але вони володіють вельми середніми навичками в роботі з комп'ютерами. Інтерфейс програми повинен бути інтуїтивно зрозумілий. У свою чергу, розробник-програміст повинен розуміти характер виконуваних кінцевим користувачем операцій. Рекомендаціями в цьому випадку можуть служити підвищення ефективності управління розробкою інформаційних систем, поліпшення інформованості розробників про предметну область.

Необхідно дослідити та побудувати архітектуру підсистеми, взаємодію з якою буде проводити користувач. Невід'ємною частиною даної роботи є опис програмної реалізації майбутньої системи з точки зору вхідних та вихідних даних, введення даних, пошуку інформації.

В результаті успішного виконання роботи ми повинні отримати повноцінно функціонуючу підсистему торгів на електронних біржах, яка здатна надійно, ефективно та в повному обсязі виконувати задачі поставлені перед нею. Програма повинна бути гнучкою до змін особливостей роботи окремих бірж, не потребувати частих супровідних робіт по модифікації та

адаптації. Повинні бути встановлені вимоги до програмного та апаратного забезпечення, виконання яких є необхідним для ефективної роботи проекту.

1.3 Постановка задачі

Мета:

Метою даної кваліфікаційної роботи є розробка зазначеної підсистеми торгівлі на електронних біржах, що дозволить ефективно відстежувати та реагувати на будь-які ситуації на ринку, своєчасно та менш енерго- та часозатратно проводити торгові операції.

Об'єкт вдосконалення:

Електронні системи біржової торгівлі криптовалютою.

Загальні цілі розробки:

- Використати сучасні інформаційні технології для ефективної роботи систем електронної торгівлі, що повинно вирішити недоліки, які притаманні об'єкту вдосконалення, а саме забезпечення вузького спектру функціоналу;
- Підвищити ефективність і якість роботи інформаційної системи торгів;
- Розширити базовий функціонал шляхом додавання спеціальних можливостей в програму;
- Автоматизована система електронних торгів повинна дотримуватися вимог надійності, гнучкості, ефективності, а також система повинна забезпечувати цілісність даних і безпечний доступ до інформаційної системи.
- Створити єдиний програмний продукт для декількох торгових площадок, та забезпечити подальше легке додавання нових бірж по мірі необхідності користувачів.

Задача та функції підсистеми:

Задачею системи, що створюється, є додавання, або в деяких випадках розширення наступних функцій:

- Додавання користувацьких ключів доступу до бірж
- Створення наказів на покупку/продаж (редагування та видалення)
- Створення правил автоматичного додавання/зняття наказів (редагування та видалення)
- Розрахунок мінімальної зміни ціни для беззбиткових торгів
- Можливість розрахунку бажаного прибутку враховуючи певні фактори
- Локалізація на українській мові
- Зручний перегляд власної історії торгів, можливість повтору вже проведених операцій
- Відображення повного «стакану» з додатковою інформацією
- Безпечне зберігання користувацьких даних

РОЗДІЛ 2 РОЗРОБКА АРХІТЕКТУРИ ПІДСИСТЕМИ ТОРГІВ

2.1 Дерево цілей системи

Дерево цілей - це графічне зображення взаємозв'язку і підпорядкованості цілей, що відображає розподіл місії і мети на цілі, під цілі, завдання та окремі дії.

Основна ідея щодо побудови дерева цілей - декомпозиція.

Декомпозиція - це метод розкриття структури системи, при якому за однією ознакою її поділяють на окремі складові. Декомпозиція використовується для побудови дерева цілей, щоб пов'язати генеральну мету зі способами її досягнення, що сформульовані у вигляді завдань окремим виконавцям.

Основне правило побудови дерева цілей - це повнота редукції - процес зведення складного явища, процесу або системи до більш простих складових. Для реалізації цього правила використовують такий системний підхід:

- ціль вищого рівня є орієнтиром, основою для розробки (декомпозиції) цілей нижчого рівня;
- цілі нижчого рівня є способами досягнення мети вищого рівня і мають бути представлені так, щоб їхня сукупність зумовлювала досягнення початкової цілі;

Дерево цілей підсистеми торгів електронної біржі зображено на Рисунку 2.1.

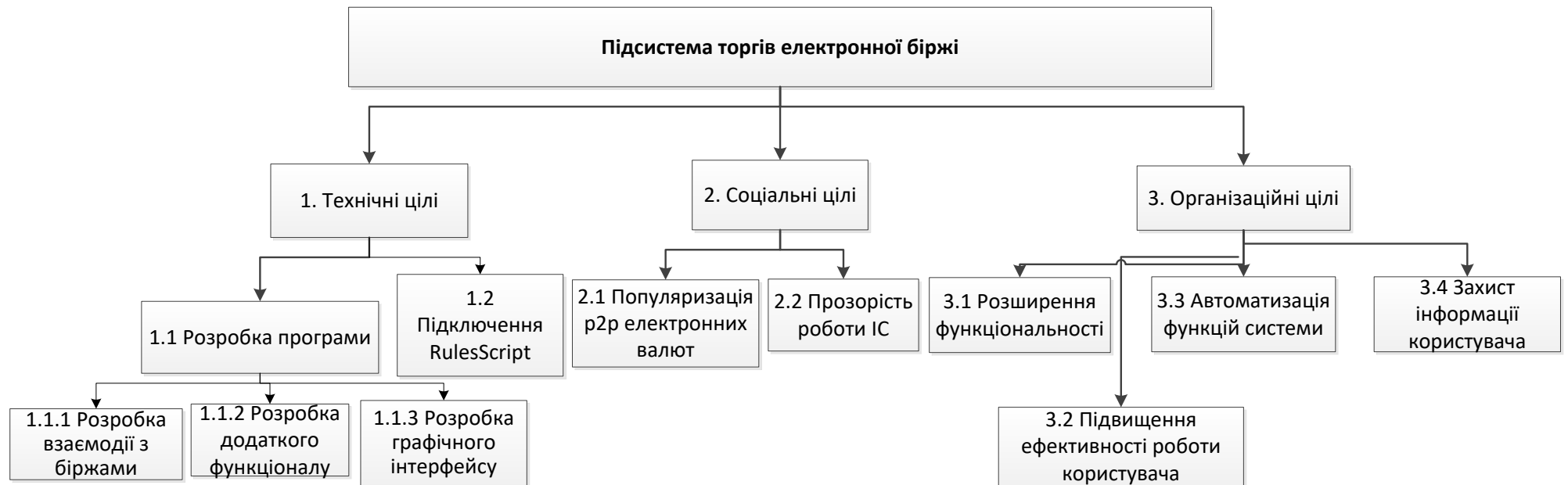


Рисунок 2.1 Дерево цілей підсистеми торгів

Детальний опис цілей системи:

1. Технічні цілі – це поставлені цілі, які необхідні для технічного забезпечення підсистеми торгів.
 - 1.1 Розробка програми – необхідно розробити програму, яка буде виконувати роль робочої станції торгівельних платформ.
 - 1.1.1 Розробка взаємодії з біржами – для функціонування системи, необхідно розробити методи взаємодії програми з електронними біржами.
 - 1.1.2 Розробка додаткового функціоналу – необхідно розробити додаткові функції та інформаційні елементи для підвищення ефективності проведення торгів користувача.
 - 1.1.3 Розробка графічного інтерфейсу – необхідно створити зручний та зрозумілий інтерфейс для легкого доступу до всіх доступних функцій системи.
 - 1.2 Підключення RulesScript – необхідно впровадити в систему спеціальну мову правил та стратегій RulesScript.
2. Соціальні цілі – цілі для роботи з користувачами електронних бірж.
 - 2.1 Популяризація р2р електронних валют – поширення програмного продукту не тільки серед користувачів бірж електронних валют, але і серед більш традиційних валютних ринків (Forex та інші).
 - 2.2 Прозорість роботи системи – кожен користувач має змогу отримати ті чи інші дані роботи системи, таких як статистика проведених торгів, виставлених поточних наказів та інше.
3. Організаційні цілі – це цілі, які сприяють покращенню роботи торгової системи.
 - 3.1 Розширення функціональності – при впровадженні підсистеми торгів електронної біржі, функціональна частина біржових платформ збільшиться.

- 3.2 Підвищення ефективності роботи користувача – при впровадженні підсистеми торгів електронної біржі, ефективність роботи користувача підвищиться, так як більша частина проведення торгів може виконуватись у автоматичному режимі без постійної присутності користувача.
- 3.3 Автоматизація функцій системи – при впровадженні підсистеми торгів електронної біржі, більша частина функцій повністю автоматизуються.
- 3.4 Захист інформації користувача – інформація буде зберігатися в конфігураційному файлі програми, який шифрується надійними криптографічними алгоритмами та запобігає використанню даних злоумисниками.

2.2 Дерево функцій системи

У процесі управління, органи управління виконують певні функції, сукупність яких охоплює весь склад завдань управління і характеризує зміст управління. Ця сукупність являє собою систему взаємопов'язаних функцій управління. Така система (сукупність) функцій може бути розчленована на більш прості сукупності завдань управління і представлена деревом функцій.

Функціональна декомпозиція. Декомпозиція базується на аналізі функцій системи. При цьому ставиться питання що робить система, незалежно від того, як вона працює. Підставою розбиття на функціональні підсистеми служить спільність функцій, виконуваних групами елементів.

Дерево функцій підсистеми торгів електронної біржі зображено на Рисунку 2.2.

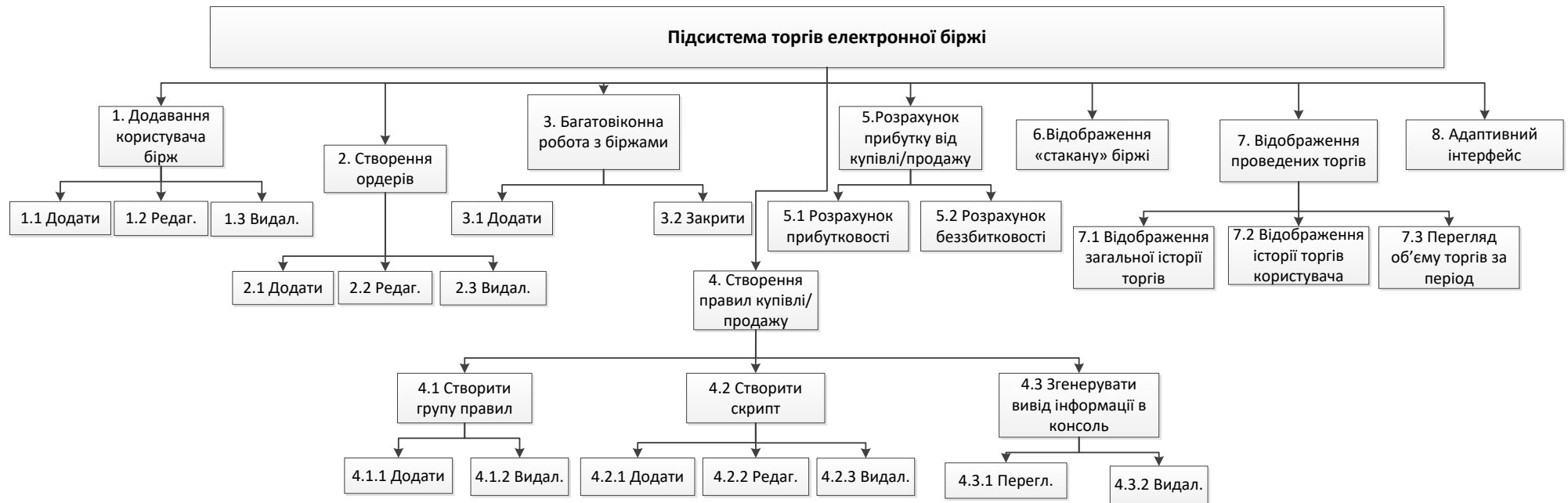


Рисунок 2.2 Дерево функцій підсистеми торгів електронної біржі

Детальний опис функцій системи:

1. Користувач – функція для роботи з користувачами програми.
 - 1.1 Додати – додавання користувача певної біржі.
 - 1.2 Редагування – редагування даних користувача біржі.
 - 1.3 Видалення – видалення даних користувача біржі.
2. Ордер – функція для маніпуляції з ордерами на купівлю/продаж.
 - 2.1 Додати – створення ордера на купівлю/продаж.
 - 2.2 Редагування – редагування вже створеного ордеру.
 - 2.3 Видалення – видалення ордеру до його виконання.
3. Багатовіконна робота – це функція для одночасного запуску декількох користувачів з різних бірж.
 - 3.1 Додати – Відкрити додаткове вікно з іншим користувачем.
 - 3.2 Закрити – Закрити непотрібні вікна.
4. Правила купівлі/продажу – це функція створення правил виконання на будь-які операції на біржі.
 - 4.1 Група правил – створення правил одного типу для досягнення одної конкретної цілі.
 - 4.1.1 Додати – Створити групу правил.
 - 4.1.2 Видалення – Видалити всю групу правил.
 - 4.2 Скрипт – створення одного правила на виконання.
 - 4.2.1 Додати – створити правило.
 - 4.2.2 Редагування – редагування правила.
 - 4.2.3 Видалення – видалення правила.
 - 4.3 Вивід інформації – відображення стадії виконання правил у спеціальній консолі.
 - 4.3.1 Перегляд – переглянути інформацію по правилу.
 - 4.3.2 Видалення – видалення повідомлень вже виконаної частини правил.

- 5 Прибуток від купівлі/продажу – калькулятор розрахунку прибутковості торгів.
 - 5.1 Прибутковість – розрахунок прибутковості проведення операцій.
 - 5.2 Беззбитковість – Розрахунок мінімальних цін купівлі/продажу для беззбиткових торгів.
- 6 «Стакан» біржі – відображення загальних виставлених ордерів на біржі.
- 7 Проведені торги – функції, які відображують історію торгів.
 - 7.1 Загальна історія – останні операції купівлі/продажу для усієї біржі.
 - 7.2 Історія користувача – останні проведені операції користувача.
 - 7.3 Об'єми проведених торгів – розрахунковий об'єм проведених торгів за останній період часу.
- 8 Адаптивний інтерфейс – можливість користувача відкріпляти/закріпляти деякі елементи програми у довільному порядку.

2.3 Функціональні моделі

IDEF0 — Function Modeling — методологія функціонального моделювання і графічного описання процесів, призначена для формалізації і опису бізнес-процесів. Особливістю IDEF0 є її акцент на ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області. В IDEF0 розглядаються логічні зв'язки між роботами, а не послідовність їх виконання в часі (WorkFlow).

Так само відображаються всі сигнали управління. Така модель є однією з найпрогресивніших моделей і використовується в організації бізнес проектів і проектів, що базуються на моделюванні всіх процесів як адміністративних, так і організаційних.

У даному підрозділі будуть розглянуті моделі тільки головних функцій підсистеми торгів електронної біржі, а саме: додавання користувача, створення ордерів, створення правил та калькулятор прибутку.

IDEF0 модель «Користувач»



Рисунок 2.3 Користувач

Вхідні дані:

Назва біржі, API ключ, ключ «Secret», ім'я профілю, пароль.

Механізм:

ОПР, ЕОМ.

Управління:

Права адміністратора, запит на створення користувача.

Вихідні дані:

Запис про користувача.

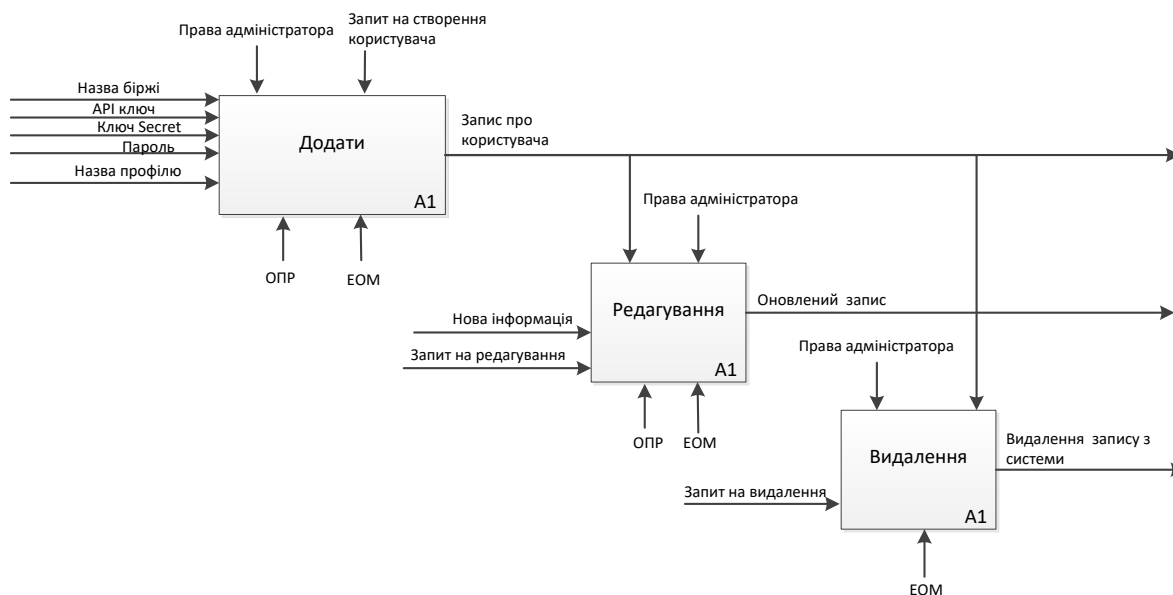


Рисунок 2.4 Декомпозиція користувача

IDEF0 модель «Ордер»

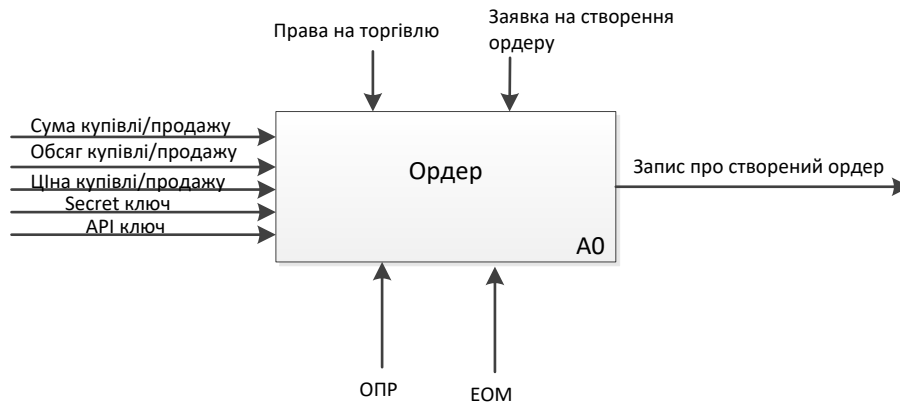


Рисунок 2.5 Ордер

Вхідні дані:

Сума, Обсяг, Ціна, API ключ, ключ «Secret».

Механізм:

ОПР, ЕОМ.

Управління:

Права на торгівлю, заявка на створення ордеру.

Вихідні дані:

Запис про створений ордер.

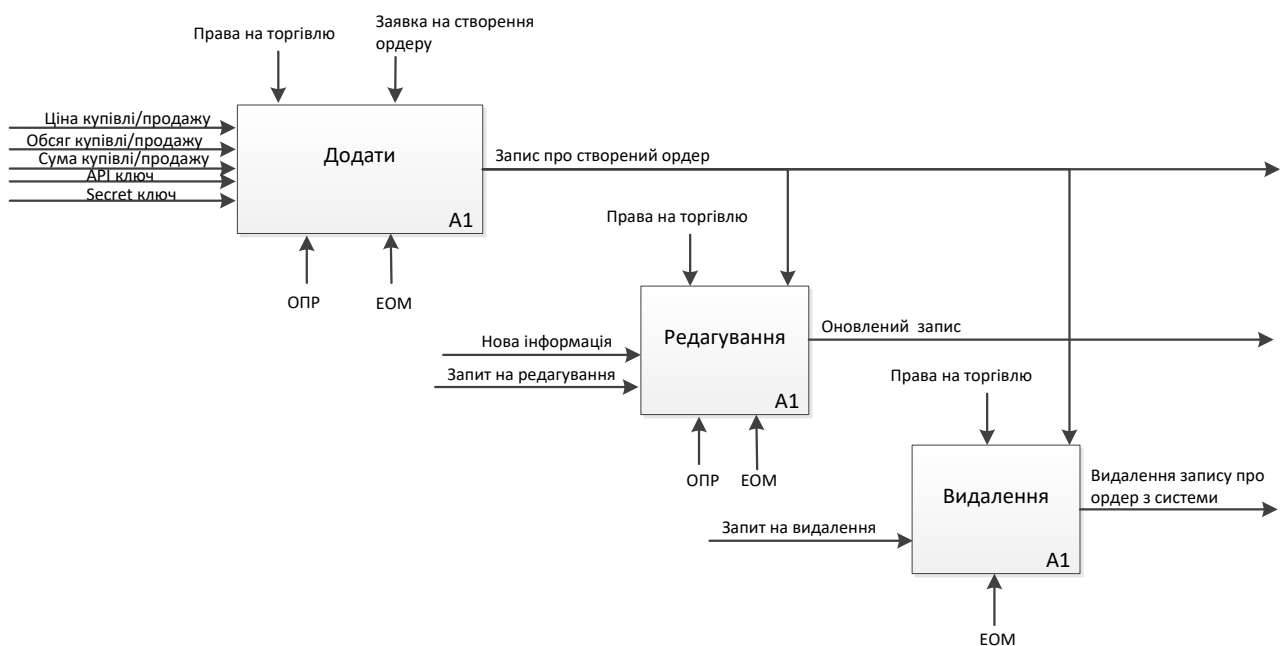


Рисунок 2.6 Декомпозиція ордеру

IDEF0 модель «Правила»

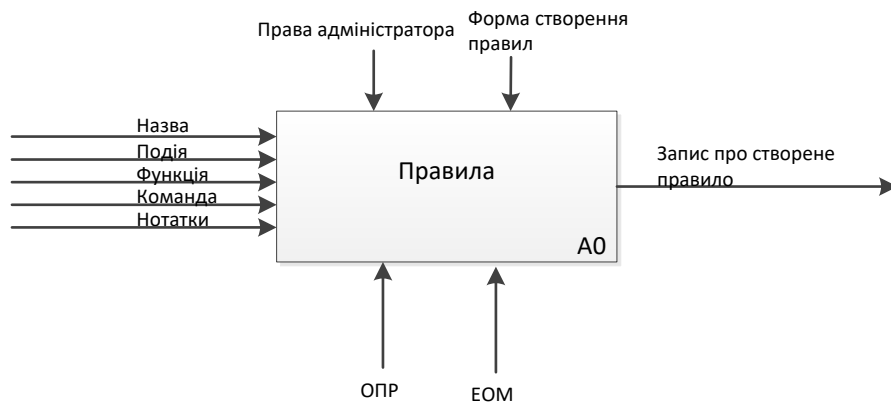


Рисунок 2.7 Правила

Вхідні дані:

Назва, подія, функція, команда, нотатки.

Механізм:

ОПР, ЕОМ.

Управління:

Права адміністратора, форма створення правил.

Вихідні дані:

Запис про створене правило.

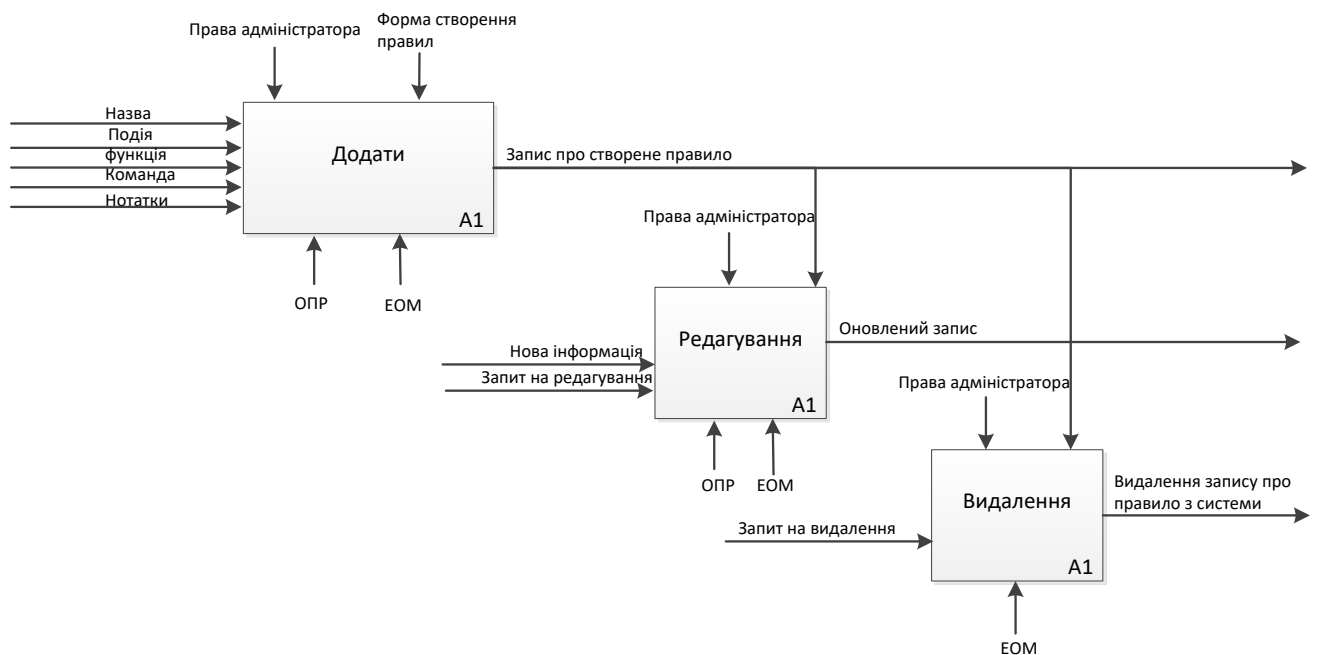


Рисунок 2.8 Декомпозиція правил

IDEF0 модель «Калькулятор прибутку»

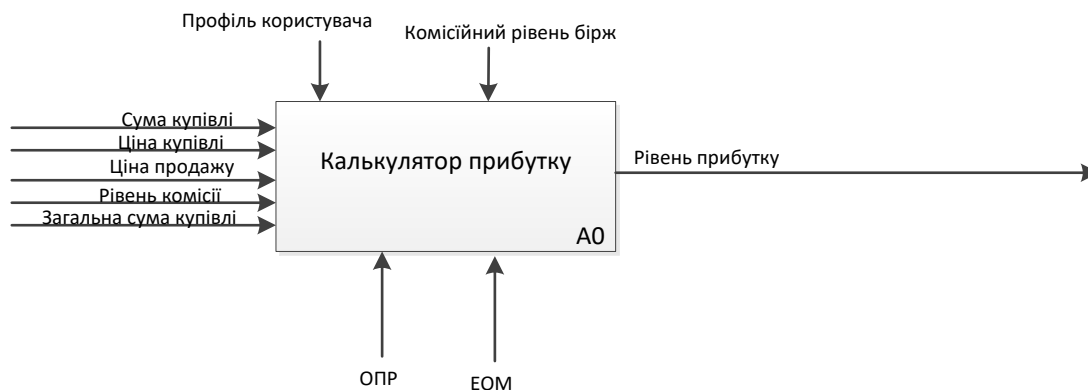


Рисунок 2.9 Калькулятор прибутку

Вхідні дані:

Сума купівлі, ціна купівлі, ціна продажу, рівень комісії, загальна сума.

Механізм:

ОПР, ЕОМ.

Управління:

Профіль користувача, комісійний рівень бірж.

Вихідні дані:

Рівень прибутку.

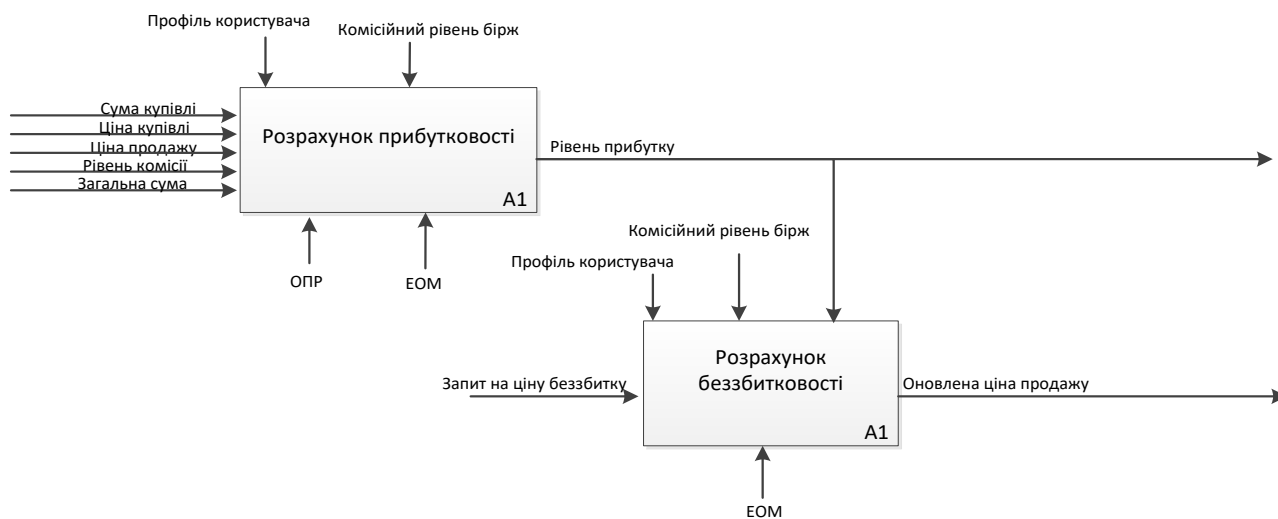


Рисунок 2.10 Декомпозиція калькулятора прибутку

2.4 DFD діаграми потоків даних

Діаграма потоків даних (англ. Data Flow Diagram) — модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма потоків даних також може використовуватись для візуалізації процесів обробки даних (структурне проектування).

Діаграми потоків даних відображають послідовність перетворення даних в ході виконання процесу, крім того, включають підпроцеси перегляду інформації та джерела її уточнення.

Діаграми потоків даних містять чотири типи графічних елементів:

- процеси - являють собою трансформацію даних в рамках описуваної системи;
- сховища даних;
- зовнішні по відношенню до системи сутності;
- потоки даних між елементами трьох попередніх типів.

У даному підрозділі DFD моделі будуть побудовані для деталізації, попередньо розглянутих у розділі 2.3, IDEF0 моделей.

DFD діаграма «Користувач»

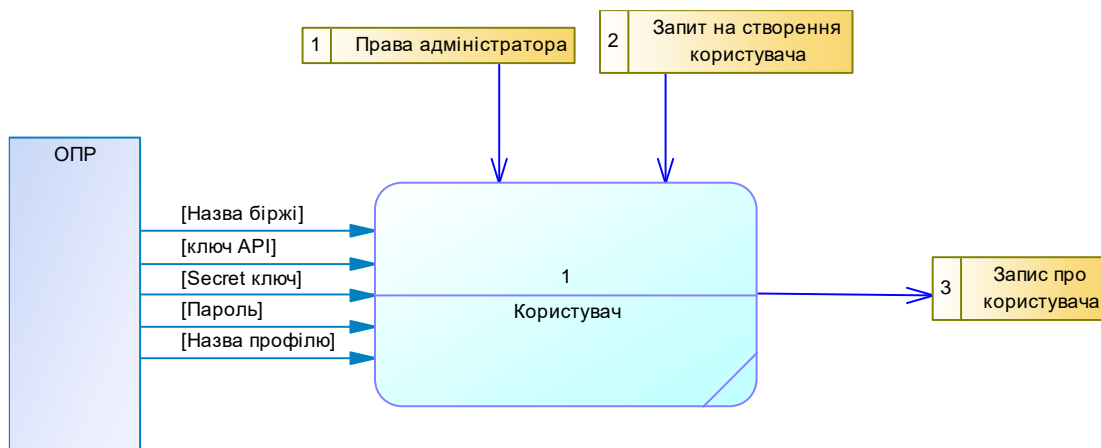


Рисунок 2.11 Діаграма користувача

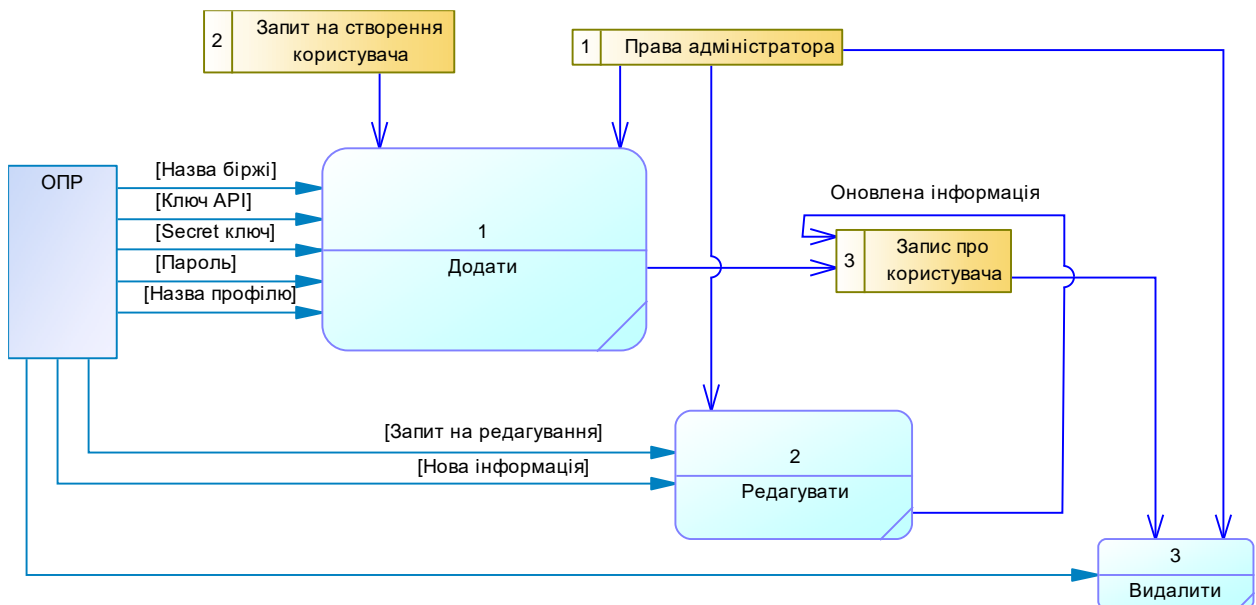


Рисунок 2.12 Декомпозиція діаграми користувача

Опис потоків даних

@ Назва	@ Тип	@ БНФ
Назва біржі	Дискретний	«Назва біржі»
Ключ API	Дискретний	«Ключ API користувача»
Secret ключ	Дискретний	«Секретний ключ користувача»
Пароль	Дискретний	«Пароль користувача»
Назва профілю	Дискретний	«Назва профілю»
Запит на створення користувача	Дискретний	«Запит на створення користувача»
Права адміністратора	Дискретний	«Доступ до системи управління»
Запис про користувача	Дискретний	«Запис даних користувача у конфіг. файлі»

Таблиця 2.1 Опис потоків даних функції Користувач

DFD діаграма «Ордер»

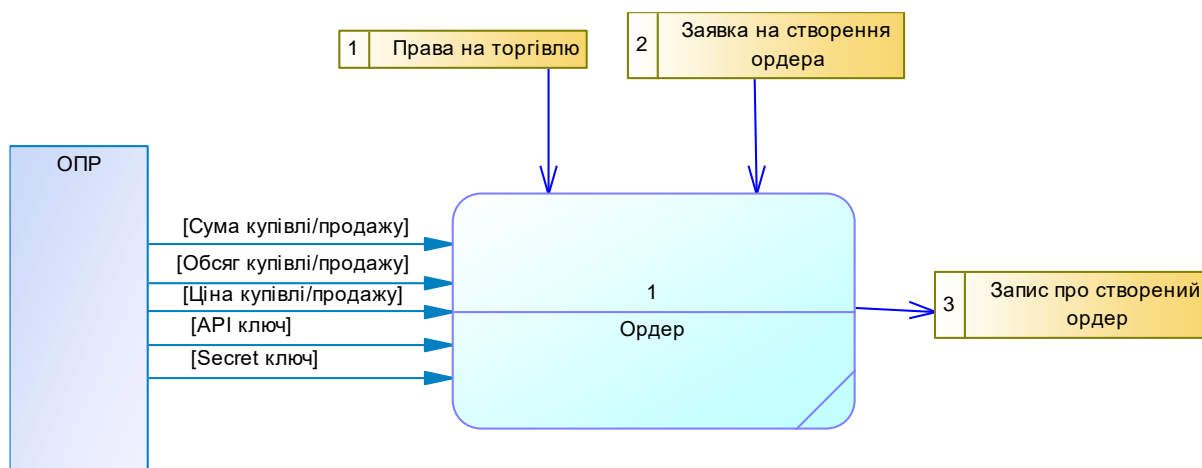


Рисунок 2.13 Діаграма ордера

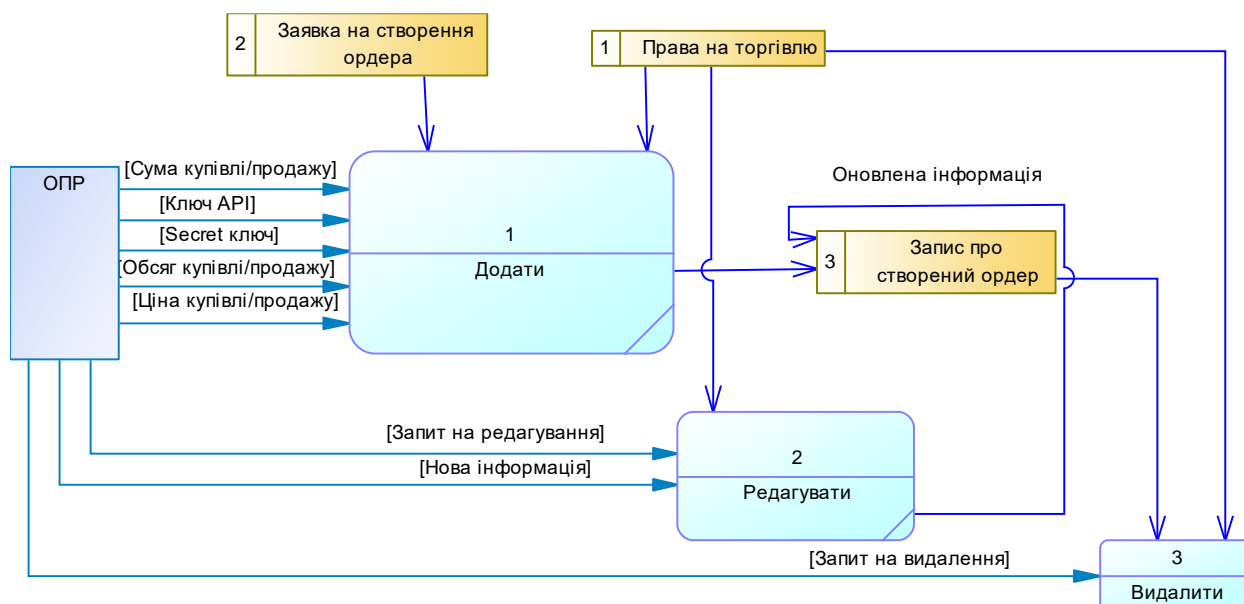


Рисунок 2.14 Декомпозиція діаграми ордера

Опис потоків даних

@ Назва	@ Тип	@ БНФ
Сума купівлі/продажу	Дискретний	«Сума коштів для купівлі/продажу»
Обсяг купівлі/продажу	Дискретний	«Обсяг валюти для купівлі/продажу»
Ціна купівлі/продажу	Дискретний	«Ціна для купівлі/продажу»
Ключ API	Дискретний	«Ключ API»
Secret ключ	Дискретний	«Secret ключ»
Права на торгівлю	Дискретний	«Доступ до проведення торгів користувача»
Заявка на створення ордера	Дискретний	«Заявка на створення ордера»
Запис про створений ордер	Дискретний	«Запис про створений ордер на біржі»

Таблиця 2.2 Опис потоків даних функції Ордер

DFD діаграма «Правила»

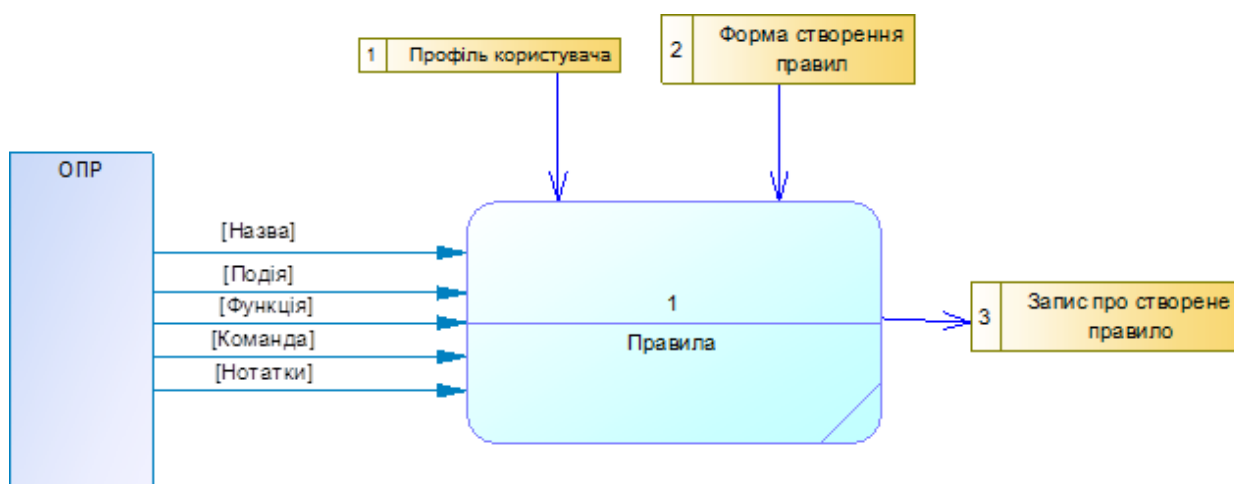


Рисунок 2.15 Діаграма правил

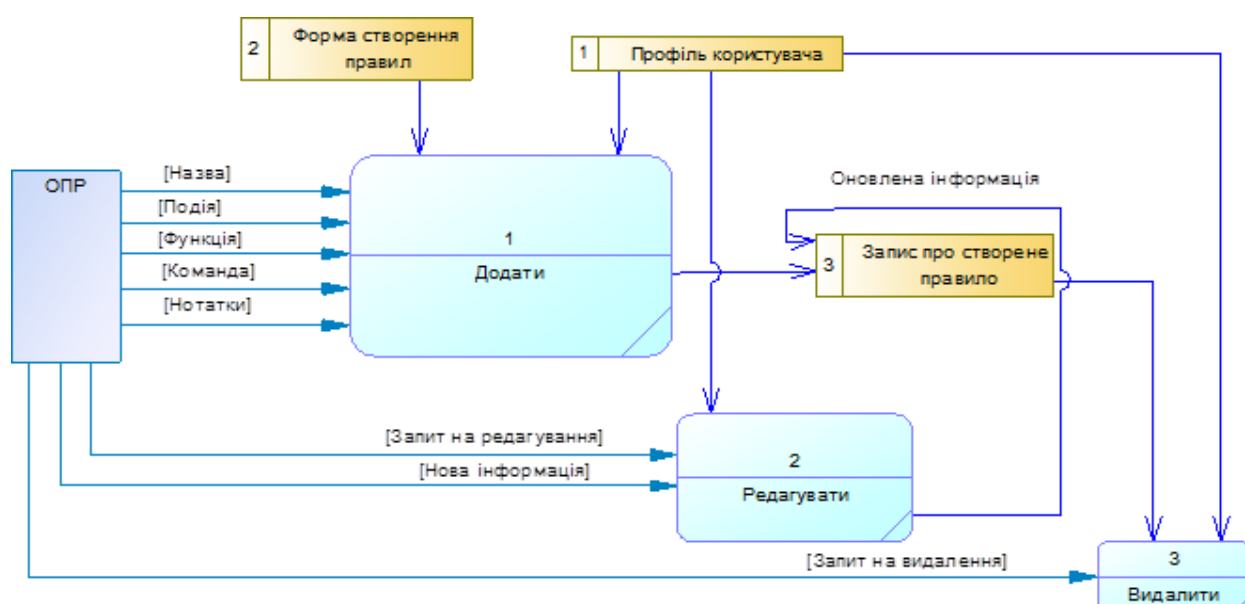


Рисунок 2.16 Декомпозиція діаграми правил

Опис потоків даних

@ Назва	@ Тип	@ БНФ
Назва	Дискретний	«Назва правила»
Подія	Дискретний	«Подія для виконання правила»
Функція	Дискретний	«Функції правила»
Команда	Дискретний	«Команда при виконанні правила»
Нотатки	Дискретний	«Нотатки для користувача»
Форма створення правил	Дискретний	«Форма для створення правила»
Профіль користувача	Дискретний	«Доступ до системи управління»
Запис про створене правило	Дискретний	«Запис про створене правило у конфіг. файлі»

Таблиця 2.3 Опис потоків даних функції Правила

DFD діаграма «Калькулятор прибутку»

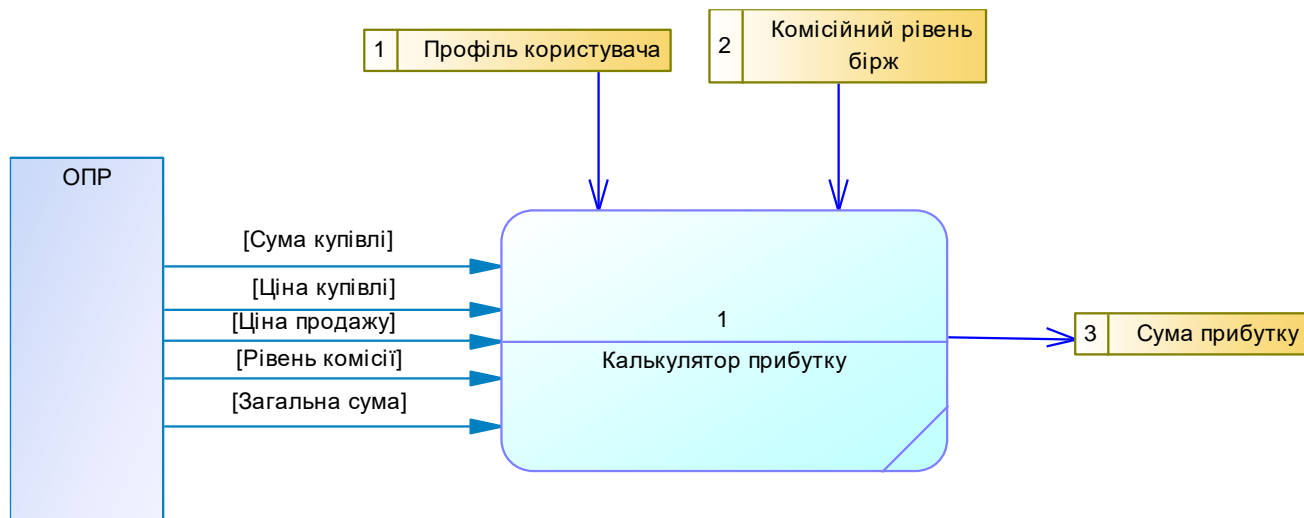


Рисунок 2.17 Діаграма калькулятора прибутку

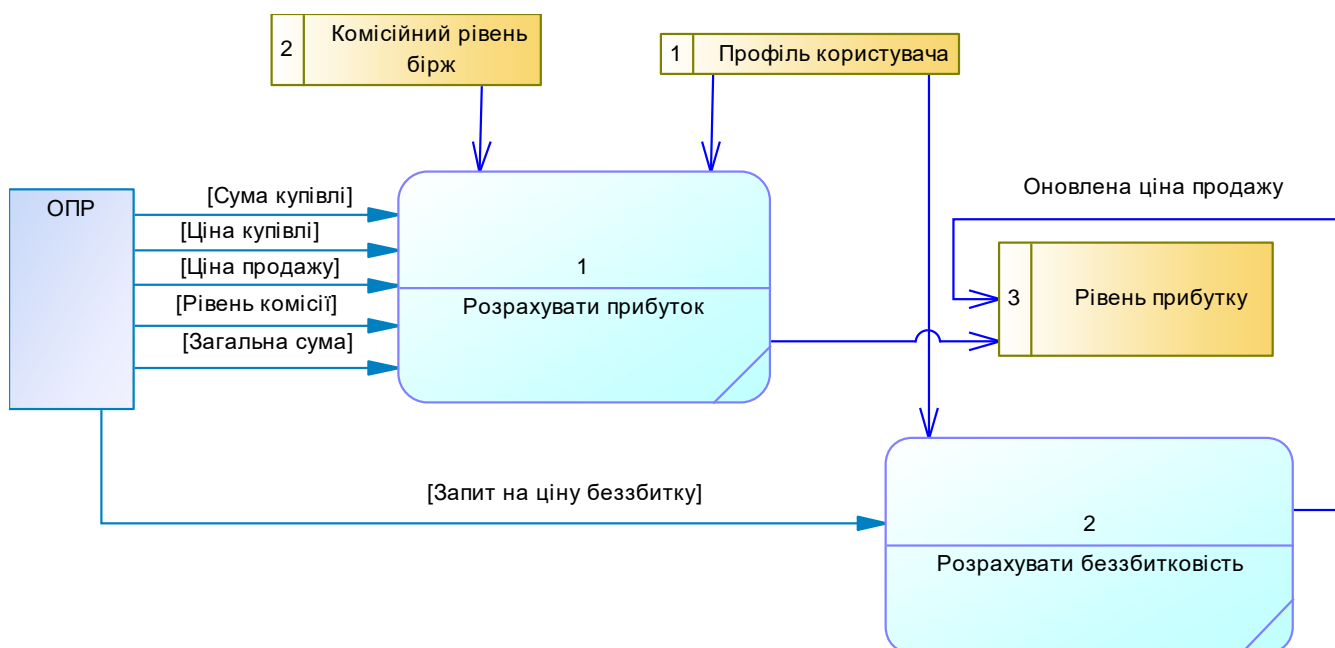


Рисунок 2.18 Декомпозиція діаграми Калькулятор прибутку

Опис потоків даних

@ Назва	@ Тип	@ БНФ
Сума купівлі	Дискретний	«Сума валюти для купівлі»
Ціна купівлі	Дискретний	«Ціна для купівлі»
Ціна продажу	Дискретний	«Ціна для продажу»
Рівень комісії	Дискретний	«Рівень комісії з кожної операції »
Загальна сума	Дискретний	«Загальна сума на яку виконується купівля»
Комісійний рівень бірж	Дискретний	«Рівень комісій визначений біржою»
Профіль користувача	Дискретний	«Доступ до системи управління»
Рівень прибутку	Дискретний	«Рівень прибутку від торгів»

Таблиця 2.4 Опис потоків даних функції Калькулятор прибутку

РОЗДІЛ 3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ ЕЛЕКТРОННОЇ БІРЖИ

База даних (скорочено — БД) — впорядкований набір логічно взаємопов'язаних даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів.

Головне завдання БД — гарантоване збереження значних обсягів інформації (так звані записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином, БД складається з двох частин: збереженої інформації та системи керування нею.

Створення бази даних слід починати з її проектування (розробки). У результаті проектування має бути визначена структура бази, тобто склад таблиць, їхня структура та логічні зв'язки. Структура реляційної таблиці визначається складом стовпців, їхньою послідовністю, типом даних кожного стовпця та їхнім розміром, а також ключем таблиці.

3.1 Визначення сутностей БД системи

Таблиця 3.1 Сутності та їх атрибути

Назва сутності	Атрибути
Користувачі	ІД користувача
	Електронна пошта
	Номер телефону
	Нікнейм
	ПІБ
Рахунки	ІД рахунку
	Валюта
	Баланс
	Дата створення
	Дата оновлення
	Комісія

Ордери	ІД ордеру
	Тип
	Валюта
	Ціна
	Обсяг
	Початковий обсяг
	Кількість торгів
	Дата створення
Торги	ІД торгу
	Валюта
	Обсяг
	Дата виконання
	Ціна
	Кошти
	Тренд
Внески	ІД внеску
	Валюта
	Кількість
	Комісія
	Статус
	Дата внеску
	Дата підтвердження
Виводи	ІД виводу
	Валюта
	Кількість
	Комісія
	Кошти
	Дата створення
	Дата підтвердження

Платіжні адреси	ІД адреси
	Адреса
	Дата створення
	Дата оновлення
	Валюта
Транзакції	ІД транзакції
	Кількість
	Статус
	Кількість підтверджень
	Дата створення
	Валюта
	Тип
Ключі доступу	ІД ключа
	Назва
	Секретний ключ
	Ключ доступу
	Дата використання

3.2 Концептуальна модель бази даних

Розробка концептуальної моделі базується на опису предметної області, отриманому у результаті її обстеження. Визначення складу і структури даних предметної області, які повинні знаходитись у БД будують виходячи з необхідності забезпечення потрібних запитів, задач і додатків користувача. Ці дані представлені у вигляді атрибутів необхідних документів, а також у мовному опису предметної області. Концептуальна модель бази даних зображена на Рисунку 3.1.

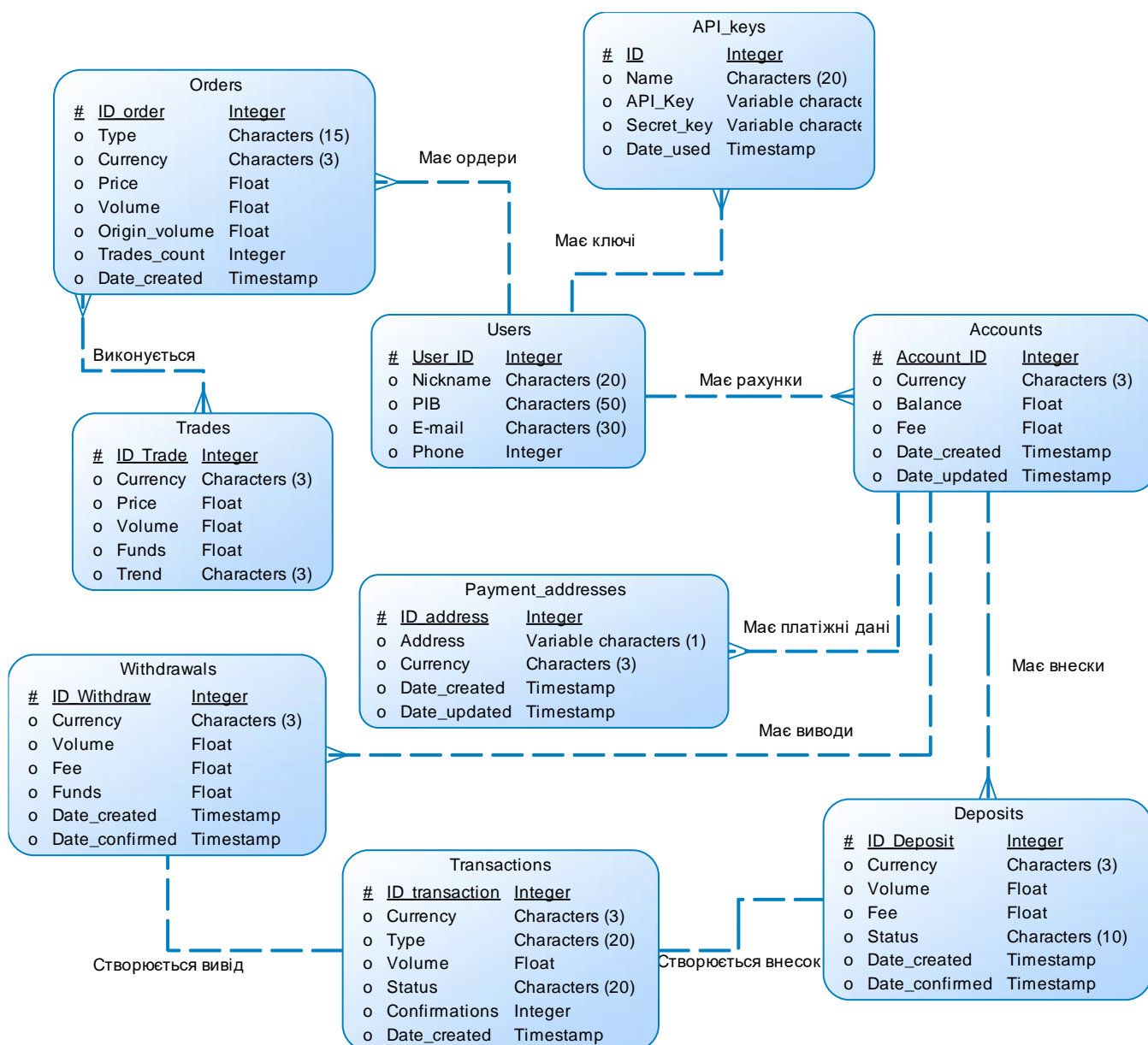


Рисунок 3.1 Концептуальна модель бази даних

3.3 Логічна модель бази даних

Логічна структура реляційних таблиць визначається відповідно до складу атрибутів інформаційних об'єктів. Структурні зв'язки ІЛМ відображаються на логічному рівні у вигляді ключових зв'язків, які реалізуються, коли у логічно зв'язаних таблицях є однакові атрибути (ключі зв'язку). При цьому ключ зв'язку обов'язково повинен бути первинним ключем для головного об'єкту (таблиці) і може бути, як первинним так і вторинним (найчастіше) ключем для другого об'єкту.

Відповідно до приведеної вище ІЛМ і правил побудуємо логічну структуру реляційної БД.

Виходячи з задач системи та проведеного інфологічного дослідження сформовані наступні таблиці для збереження інформації.

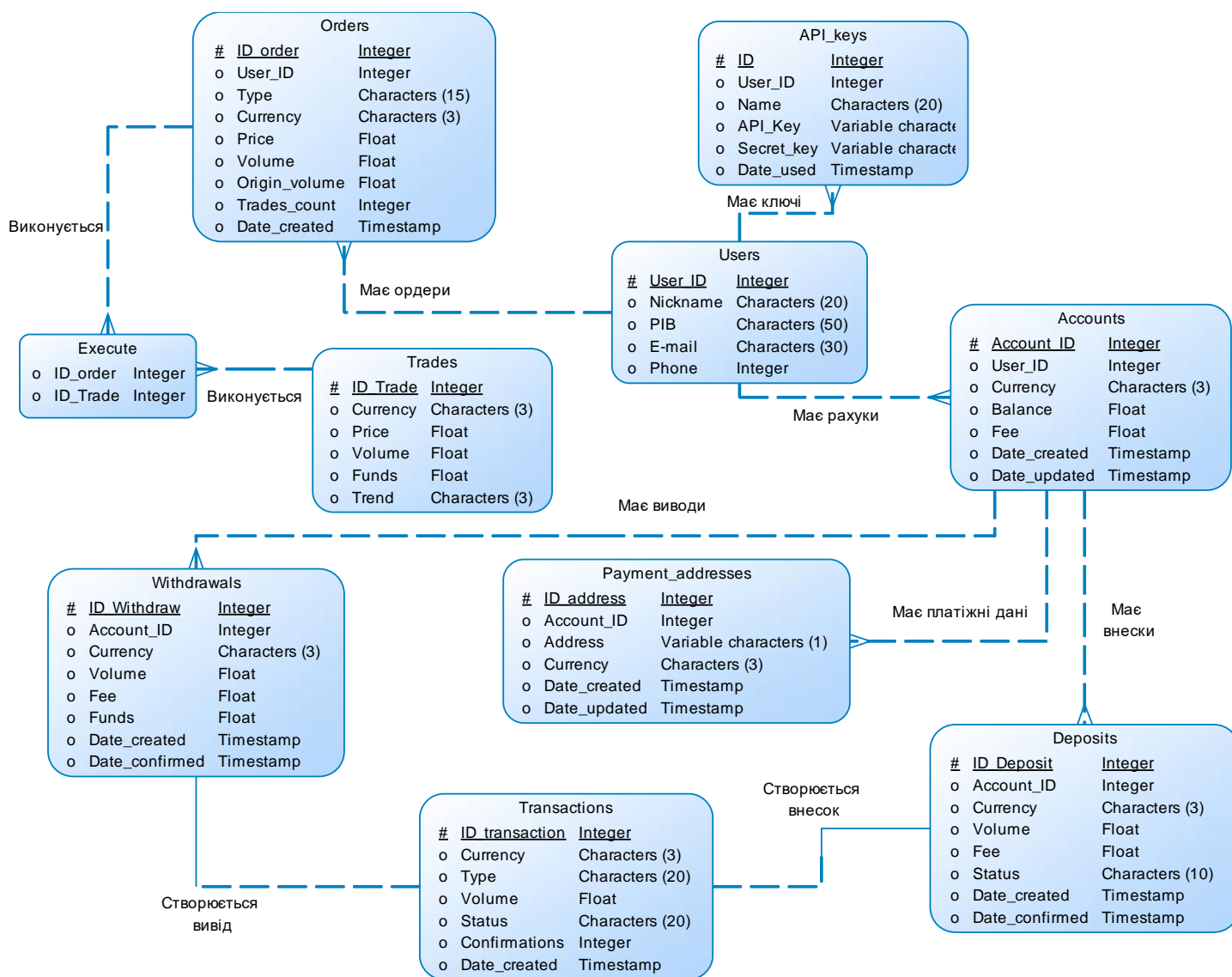


Рисунок 3.2 Логічна модель бази даних

Список відношень між сутностями бази даних наведено у таблиці 3.2

Таблиця 3.2 Відношення

Назва зв'язку	Сутність 1	Сутність 2	Відношення
Виконується	Execute	Orders	1,n
Виконується	Execute	Trades	1,n
Має ордери	Users	Orders	0,n
Має ключі	Users	API_keys	1,n
Має рахунки	Users	Accounts	1,n
Має виводи	Accounts	Withdrawals	0,n
Має платіжні дані	Accounts	Payment_addresses	1,n
Має внески	Accounts	Deposits	1,n
Створюється внесок	Transactions	Deposits	1,1
Створюється вивід	Transactions	Withdrawals	1,1

3.3 Фізична модель бази даних

Фізичне проектування – це процес створення опису реалізації БД на вторинних запам'ятовуючих пристроях із указівкою структур збереження і методів доступу, використовуваних для організації ефективної обробки даних. Фізичний етап проектування забезпечує вибір раціональної структури збереження даних і методів доступу до них, виходячи з арсеналу методів і засобів, що надаються розроблювачу конкретної СУБД.

Основною метою фізичного проектування БД є опис способу фізичної реалізації логічного проекту БД.

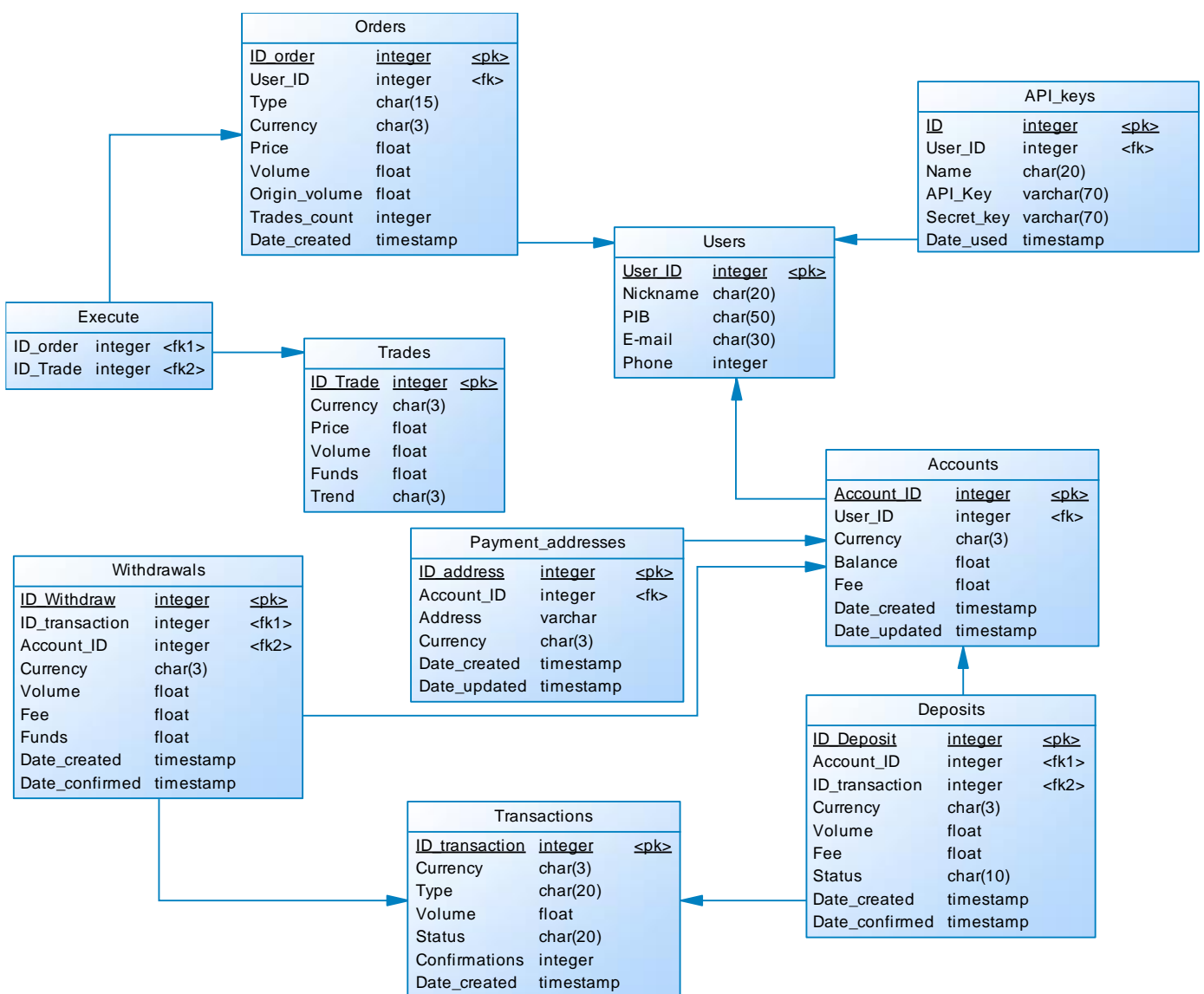


Рисунок 3.3 Фізична модель бази даних

В таблицях 3.3 – 3.13 приводиться фізична реалізація таблиць бази даних

Таблиця 3.3 Користувач

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер користувача	ID_ User	int	X	
Нікнейм	Nickname	char		
ПІБ	PIB	char		
Телефон	Phone	int		
Електронна пошта	E-mail	char		

Поле «*Номер користувача*» - ідентифікаційний номер користувача у таблиці «Користувач».

Поле «*Нікнейм*» - Нікнейм користувача для відображення.

Поле «*ПІБ*» - ПІБ користувача.

Поле «*Телефон*» - номер телефону користувача.

Поле «*Електронна пошта*» - адреса електронної пошти користувача.

Таблиця 3.4 Рахунок

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер рахунку	ID_ Account	int	X	
Номер користувача	ID_ User	int		X
Валюта	Currency	char		
Баланс	Balance	float		
Комісія	Fee	float		
Дата створення	Date_created	date		
Дата оновлення	Date_updated	date		

Поле «*Номер рахунку*» - ідентифікаційний номер рахунку у таблиці «Рахунок».

Поле «*Номер користувача*» - ідентифікаційний номер користувача у таблиці «Користувач». Це зовнішній ключ.

Поле «*Валюта*» - валюта рахунку.

Поле «*Баланс*» - поточний баланс рахунку.

Поле «*Комісія*» - комісія за операції з рахунком.

Поле «*Дата створення*» - дата створення рахунку.

Поле «*Дата оновлення*» - дата останньої зміни на рахунку.

Таблиця 3.5 Ордери

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер ордеру	ID_Order	int	X	
Номер користувача	ID_User	int		X
Тип	Type	char		
Валюта	Currency	char		
Ціна	Price	float		
Обсяг	Volume	float		
Початковий обсяг	Origin_volume	float		
Кількість торгів	Trades_count	int		
Дата створення	Date_created	date		

Поле «*Номер ордеру*» - ідентифікаційний номер рахунку у таблиці «Ордери».

Поле «*Номер користувача*» - ідентифікаційний номер користувача у таблиці «Користувач». Це зовнішній ключ.

Поле «*Тип*» - тип створеного ордеру.

Поле «*Валюта*» - валюта створеного ордеру.

Поле «*Ціна*» - ціна по якій виконується ордер.

Поле «*Обсяг*» - поточний обсяг валюти ордеру.

Поле «*Початковий обсяг*» - початковий обсяг валюти ордеру.

Поле «*Кількість торгів*» - кількість проведених торгів по ордеру.

Поле «*Дата створення*» - дата створення ордеру.

Таблиця 3.6 Торги

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер торгу	ID_Trade	int	X	
Валюта	Currency	char		
Ціна	Price	float		
Обсяг	Volume	float		
Кошти	Funds	float		
Тренд	Trend	char		

Поле «*Номер торгу*» - ідентифікаційний номер торгівлі у таблиці «Торги».

Поле «*Валюта*» - валюта проведеної торгівлі.

Поле «*Ціна*» - ціна по якій виконується торгівля.

Поле «*Обсяг*» - обсяг торгівлі.

Поле «*Кошти*» - сума коштів виконаної торгівлі.

Поле «*Тренд*» - поле, яке вказує в яку сторону змінилась ціна після торгівлі.

Таблиця 3.7 Виконати

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер ордеру	ID_Order	int	X	X
Номер торгу	ID_Trade	int	X	X

Таблиця «Виконати» створена, для реалізації відношення «багато до багатьох»

Поле «*Номер ордеру*» - ідентифікаційний номер рахунку у таблиці «Ордери».

Це зовнішній ключ.

Поле «*Номер торгу*» - ідентифікаційний номер торгівлі у таблиці «Торги».

Це зовнішній ключ.

Таблиця 3.8 Внески

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер внеску	ID_Deposit	int	X	
Номер рахунку	ID_Account	int		X
Номер транзакції	ID_Transaction	int		X
Валюта	Currency	char		
Обсяг	Volume	float		
Комісія	Fee	float		
Статус	Status	char		
Дата створення	Date_created	date		
Дата підтвердження	Date_confirmed	date		

Поле «*Номер внеску*» - ідентифікаційний номер внеску у таблиці «Внески».

Поле «*Номер рахунку*» - ідентифікаційний номер рахунку у таблиці «Рахунок». Це зовнішній ключ.

Поле «*Номер транзакції*» - ідентифікаційний номер запису у таблиці «Транзакції». Це зовнішній ключ.

Поле «*Валюта*» - валюта внеску.

Поле «*Обсяг*» - обсяг внеску.

Поле «*Комісія*» - комісія утримана за внесок.

Поле «*Статус*» - поточний статус внеску.

Поле «*Дата створення*» - дата створення внеску.

Поле «*Дата підтвердження*» - дата успішного підтвердження внеску.

Таблиця 3.9 Виводи

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер виводу	ID_Withdraw	int	X	
Номер рахунку	ID_Account	int		X
Номер транзакції	ID_Transaction	int		X
Валюта	Currency	char		
Обсяг	Volume	float		
Комісія	Fee	float		
Кошти	Funds	char		
Дата створення	Date_created	date		
Дата підтвердження	Date_confirmed	date		

Поле «*Номер виводу*» - ідентифікаційний номер виводу у таблиці «Внески».

Поле «*Номер рахунку*» - ідентифікаційний номер рахунку у таблиці «Рахунок». Це зовнішній ключ.

Поле «*Номер транзакції*» - ідентифікаційний номер запису у таблиці «Транзакції». Це зовнішній ключ.

Поле «*Валюта*» - валюта виводу.

Поле «*Обсяг*» - обсяг виводу.

Поле «*Комісія*» - комісія утримана за вивід.

Поле «*Дата створення*» - дата створення виводу.

Поле «*Дата підтвердження*» - дата успішного підтвердження виводу.

Таблиця 3.10 Ключі доступу

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер ключа доступу	ID_API	int	X	
Номер користувача	ID_User	int		X
Назва	Name	char		
Ключ API	API_key	char		
Секретний ключ	Secret_key	char		
Дата використання	Date_used	date		

Поле «*Номер ключа доступу*» - ідентифікаційний номер ключа у таблиці «Ключі доступу».

Поле «*Номер користувача*» - ідентифікаційний номер користувача у таблиці «Користувач». Це зовнішній ключ.

Поле «*Назва*» - назва ключа доступу.

Поле «*Ключ API*» - основний ключ для доступу.

Поле «*Секретний ключ*» - тут зберігається секретний ключ доступу.

Поле «*Дата використання*» - дата останнього використання ключа доступу.

Таблиця 3.11 Платіжні адреси

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер адреси	ID_Address	int	X	
Номер рахунку	ID_Account	int		X
Адреса	Address	char		
Валюта	Currency	char		
Дата створення	Date_created	date		
Дата оновлення	Date_updated	date		

Поле «*Номер адреси*» - ідентифікаційний номер адреси у таблиці «Платіжні адреси».

Поле «*Номер рахунку*» - ідентифікаційний номер рахунку у таблиці «Рахунок». Це зовнішній ключ.

Поле «*Адреса*» - адреса платіжного рахунку.

Поле «*Валюта*» - валюта платіжної адреси.

Поле «*Дата створення*» - дата створення платіжної адреси.

Поле «*Дата оновлення*» - дата оновлення даних платіжної адреси.

Таблиця 3.12 Транзакції

Назва поля	Ідентифікатор	Тип даних	PK	FK
Номер транзакції	ID_Transaction	int	X	
Валюта	Currency	char		
Тип	Type	char		
Обсяг	Volume	float		
Підтвердження	Confirmations	int		
Дата створення	Date_created	date		

Поле «*Номер транзакції*» - ідентифікаційний номер запису у таблиці «Транзакції».

Поле «*Валюта*» - валюта в якій проводиться транзакція.

Поле «*Обсяг*» - обсяг транзакції.

Поле «*Підтвердження*» - кількість підтверджень транзакцій в мережі.

Поле «*Дата створення*» - дата створення транзакції.

**РОЗДІЛ 4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПІДСИСТЕМИ ТОРГІВ
ЕЛЕКТРОННОЇ БІРЖІ**

4.1 Вибір сучасних інформаційних технологій

Для реалізації поставленої цілі, а саме створення торгового терміналу, були обрані, згідно рисунку 4.1, наступні технології:

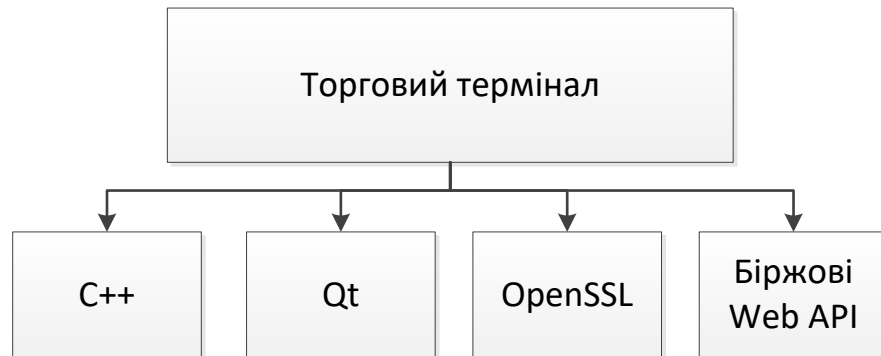


Рисунок 4.1 Технології реалізації проекту

C++

C++ - компільована статично типізована мова програмування загального призначення.

Підтримує такі парадигми програмування як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування, забезпечує модульність, роздільну компіляцію, обробку винятків, абстракцію даних, оголошення типів (класів) об'єктів, віртуальні функції. Стандартна бібліотека включає, зокрема, загальноживані контейнери і алгоритми. C++ поєднує властивості як високорівневих, так і низькорівневих мов. У порівнянні з його попередником - мовою C, - найбільшу увагу приділено підтримці об'єктно-орієнтованого і узагальненого програмування.

C++ широко використовується для розробки програмного забезпечення, будучи одним з найпопулярніших мов програмування. Область його застосування включає створення операційних систем, різноманітних прикладних програм, драйверів пристроїв, додатків для вбудованих систем, високопродуктивних серверів, а також розважальних додатків (ігор).

Qt

Qt — кросс-платформенний інструментарій розробки програмного забезпечення (ПЗ) мовою програмування C++. Дозволяє запускати написане за його допомогою ПЗ на більшості сучасних операційних систем (ОС), просто компілюючи текст програми для кожної операційної системи без зміни первісного коду. Містить всі основні класи, які можуть бути потрібні для розробки прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу й закінчуючи класами для роботи з мережею, базами даних, OpenGL, SVG і XML. Бібліотека дозволяє керувати нитями, працювати з мережею та забезпечує крос-платформенний доступ до файлів.

Відмітна особливість Qt від інших бібліотек - використання Meta Object Compiler (MOC) - попередньої системи обробки вихідного коду (в загальному, Qt - це бібліотека не для чистого C ++, а для його особливого нареччя, з якого і «переводить» MOC для подальшої компіляції будь-яким стандартним C ++ компілятором). MOC дозволяє у багато разів збільшити потужність бібліотек, вводячи такі поняття, як слоти і сигнали. Крім того, це дозволяє зробити код більш лаконічним.

Qt надає програмісту не тільки зручний набір бібліотек класів, а й певну модель розробки додатків, певний каркас їх структури. Дотримання принципів та правил «гарного стилю програмування на C ++ / Qt» істотно знижує частоту таких важко відловлюють помилок в додатках, як виток пам'яті (memory leaks), необроблені виключення, незакриті файли або незвільнені дескриптори ресурсних об'єктів, чим нерідко страждають програми, написані «на чистому C ++ » без використання бібліотеки Qt.

Важливою перевагою Qt є добре продуманий, логічний і стрункий набір класів, що надає програмісту дуже високий рівень абстракції. Завдяки цьому програмістам, які використовують Qt, доводиться писати значно менше коду, ніж це має місце при використанні, наприклад, бібліотеки класів

MFC. Сам же код виглядає стрункіше і простіше, логічніше і зрозуміліше, ніж аналогічний за функціональністю код MFC або код, написаний з використанням «рідного» для X11 тулкіта Xt. Його легше підтримувати і розвивати.

OpenSSL

OpenSSL — відкритий програмний продукт, розроблений як універсальна бібліотека для криптографії, що використовує протоколи Secure Sockets Layer і Transport Layer Security.

SSL (англ. Secure sockets layer - рівень захищених сокетів) - криптографічний протокол, який передбачає більш безпечний зв'язок. Він використовує асиметричну криптографію для аутентифікації ключів обміну, симетричне шифрування для збереження конфіденційності, коди аутентифікації повідомлень для цілісності повідомлень.

Протокол SSL забезпечує захищений обмін даних за рахунок двох наступних елементів:

- Аутентифікація
- Шифрування

Протокол SSL надає "безпечний канал", який має три основні властивості:

- 1) Канал є приватним. Шифрування використовується для всіх повідомлень після простого діалогу, який служить для визначення секретного ключа.
- 2) Канал аутентифікований. Серверна сторона діалогу завжди аутентифікує, а клієнтська робить це опціонально.
- 3) Канал надійний. Транспортування повідомлень включає в себе перевірку цілісності.

Перевагою SSL є те, що він незалежний від прикладного протоколу. Протоколи додатків (HTTP, FTP, TELNET і т.д.) можуть працювати поверх протоколу SSL абсолютно прозоро, т е SSL може погоджувати алгоритм шифрування і ключ сесії, а також аутентифіцировать сервер до того, як додаток прийме або передасть перший байт повідомлення.

API

API (інтерфейс програмування додатків, інтерфейс прикладного програмування) - набір готових класів, процедур, функцій, структур і констант, що надаються додатком (бібліотекою, сервісом) для використання в зовнішніх програмних продуктах. Використовується програмістами при написанні всіляких додатків.

API визначає функціональність, яку надає програма (модуль, бібліотека), при цьому API дозволяє абстрагуватися від того, як саме ця функціональність реалізована.

Якщо програму (модуль, бібліотеку) розглядати як чорний ящик, то API - це безліч «ручок», які доступні користувачеві даного ящика і які він може вертати і смикати.

Програмні компоненти взаємодіють один з одним за допомогою API. При цьому зазвичай компоненти утворюють ієрархію - високорівневі компоненти використовують API низькорівневих компонентів, а ті, в свою чергу, використовують API ще більш низькорівневих компонентів.

Web API в свою чергу представляє собою інтерфейс прикладного програмування (API), як для веб-сервера так і для веб-браузера.

- Серверна частина

Серверний Web API є програмний інтерфейс до певної системи повідомлень типу «запиту-відповіді», як правило, виражається в JSON або XML, який подається через веб - найбільш часто за допомогою веб-сервера на основі HTTP. Гібридні додатки є веб-додатки, які поєднують в собі використання декількох таких веб-API. Webhooks є серверні веб-інтерфейси, які беруть в якості вхідних такий URI, який призначений для використання як дистанційний “named pipe” або як тип “callback” виклику так, що сервер виступає в якості клієнта для розмінування наданої URI та ініціює подію на іншому сервері, який обробляє цю подію таким чином, щоб забезпечити однорангову мережа IPC.

У той час як "Web API" в цьому контексті іноді вважається синонімом до веб-служби, Web 2.0 веб-додатки відійшли від SOAP веб-сервісів до більш згуртованих колекцій RESTful веб-ресурсів. Ці RESTful API для веб доступні через стандартні методи HTTP та різними HTTP клієнтами, включаючи браузері і мобільні пристрої.

- Клієнтська частина

На стороні клієнта Web API є програмний інтерфейс для розширення функціональності в рамках веб-браузера або іншого клієнта HTTP. Спочатку найчастіше вони були у вигляді різних підключаючихся архітектур, проте більшість нових направлені на стандартизовані прив'язки JavaScript.

4.2 Архітектура програми та опис класів і модулів

Програмний продукт, що реалізує роботу підсистеми торгів електронної біржі, складається з трьох частин, а саме:

- GUI (головне вікно програми)
- API requests (система взаємозв'язку з біржами)
- Rules Script (класи та розширення, які дозволяють створювати правила та складні стратегії)

Частина GUI складається з наступних програмних модулів:

- PasswordDialog (відповідає за авторизацію користувача)
- Info (відображає інформацію з бірж)
- Orders (відповідає за відображення активних ордерів користувача)
- History (відповідає за відображення історії проведених операцій користувача)
- FeeCalculator (відповідає за відображення калькулятора прибутковості)
- Depth (модуль, який відображає «стакан» біржі)
- Trades (модуль, який відображає усі проведені операції біржі за останній період часу)
- Rules (модуль, який відповідає за відображення поточних правил на виконання)

Щодо API requests частини, то вона складається з наступних модулів:

- Exchange_btc-e (відповідає за взаємодію з біржею btc-e.com)
- Exchange_bitfinex (відповідає за взаємодію з біржею bitfinex.com)
- Exchange_bitstamp (відповідає за взаємодію з біржею bitstamp.net)

Третя частина являє собою спеціальну вбудовану мову Rules Script, яка складається з наступних модулів:

- Event (відповідає за відстеження подій)
- Function (відповідає за повернення значень чи розрахунків)
- Command (відповідає за виконання команд при досягненні умов)
- Log (відображає проміжну інформацію з активних правил)

Загальна архітектура проекту зображена на рисунку 4.3.

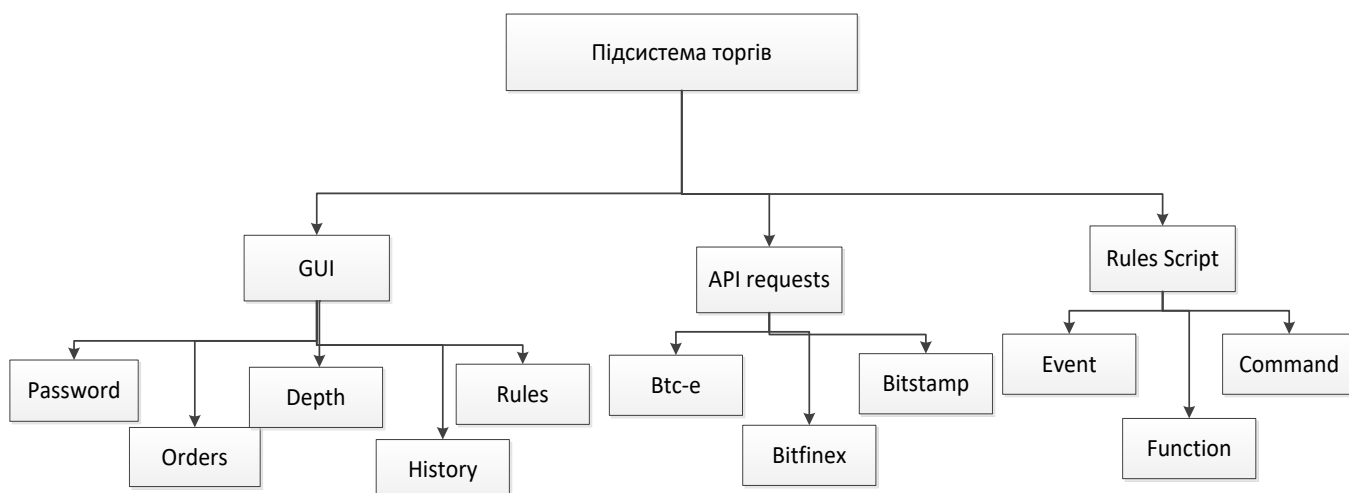


Рисунок 4.3 Загальна архітектура терміналу

Щодо програмних модулів, то кожен з них реалізований згідно документації розширення Qt, який був розглянутий у попередньому підрозділі. Бекенд модулі в свою чергу використовують API функції бірж згідно до їх опису.

Розглядати модулі GUI частини проекту немає сенсу, так як у всіх контролерах реалізована тільки одна функція на відображення тих, чи інших даних, тому ми відразу перейдемо до розгляду API requests частини, де і відбувається взаємодія з біржами, розглянемо основні запити.

Клас модуля **Exchange_btce** зображено на рисунку 4.4.

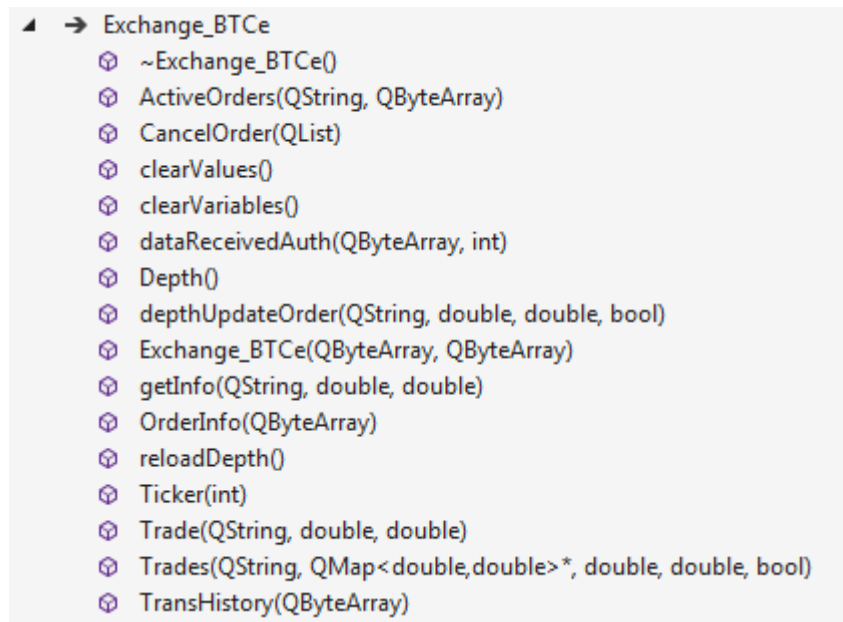


Рисунок 4.4 Клас модуля Exchange_btce

API методи модуля Exchange_btce:

- getInfo() – метод, який повертає інформацію про поточний баланс користувача, привілеї API-ключа, кількість відкритих ордерів і час сервера.
- Trade() – Основний метод використовуючи який можна створювати ордера і торгувати на біржі.
- ActiveOrders() – Повертає список активних ордерів.
- OrderInfo(order_id) – Повертає інформацію про конкретний ордер.
- CancelOrder(order_id) – Метод призначений для скасування ордера.
- TransHistory() – Метод повертає історію угод.
- Depth() – Метод повертає «стакан» біржі.
- Ticker() – Повертає загальну інформацію за останню добу.
- Trades() – Метод повертає поточні проведені торги на біржі.

Клас модуля **Exchange_bitfinex** зображено на рисунку 4.5.

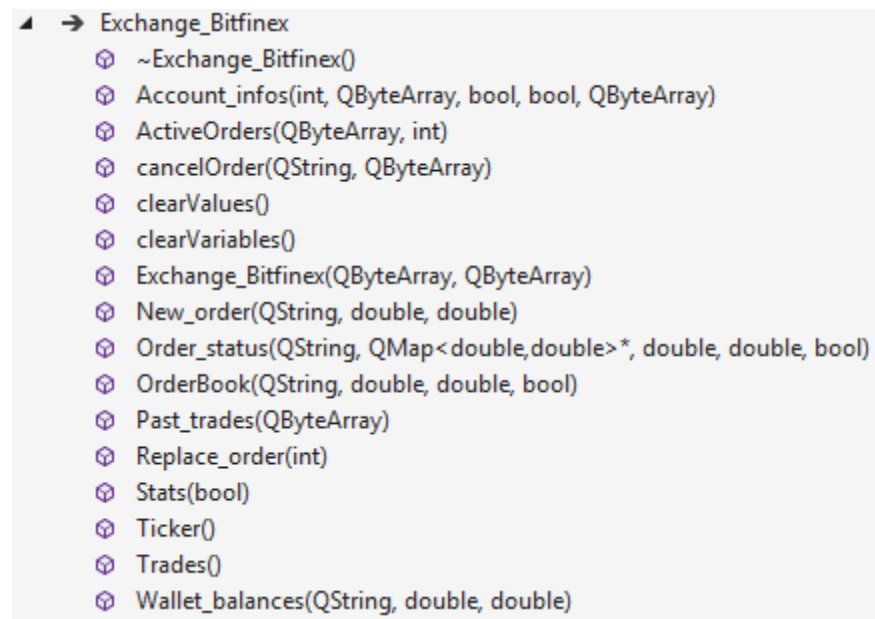


Рисунок 4.5 Клас модуля Exchange_bitfinex

API методи модуля Exchange_bitfinex:

- Wallet_balances() – метод, який повертає інформацію про поточний баланс користувача.
- Account_infos() – метод, який повертає інформацію про профіль користувача та рівень його комісій.
- New_order() – Основний метод використовуючи який можна створювати ордера і торгувати на біржі.
- Replace_order() – Метод заміни старого ордеру на новий.
- Order_status(order_id) – Повертає інформацію про ордер.
- ActiveOrders() – Повертає список активних ордерів.
- Cancel_order(order_id) – Метод призначений для скасування ордеру.
- Past_trades() – Метод повертає історію угод.
- Stats(symbol) – Різноманітна статистика по конкретній парі.
- Orderbook(symbol) – Метод повертає «стакан» біржі.
- Ticker() – Повертає загальну інформацію за останню добу.
- Trades() – Метод повертає поточні проведені торги на біржі.

Клас модуля **Exchange_bitstamp** зображено на рисунку 4.6.

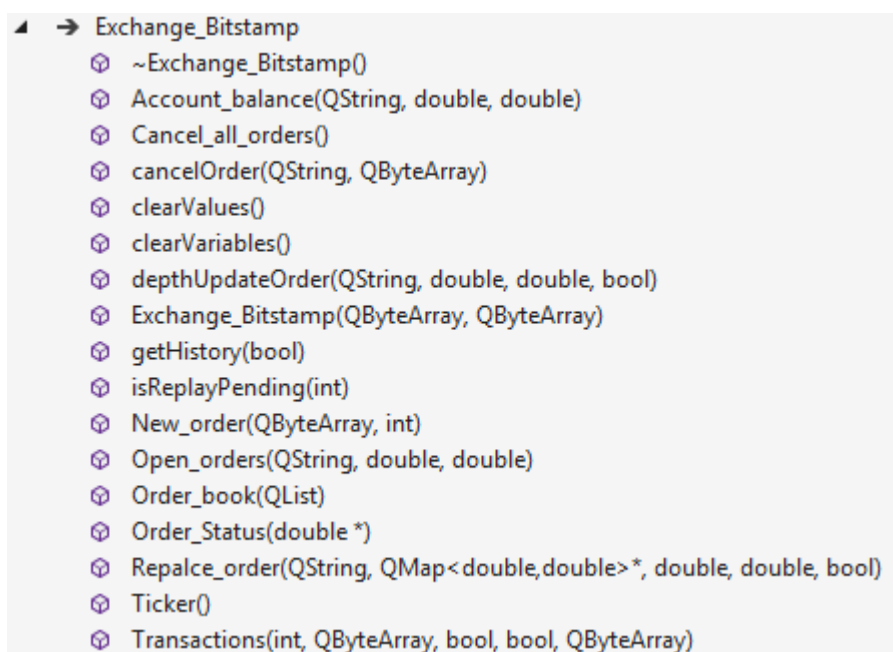


Рисунок 4.6 Клас модуля Exchange_bitstamp

API методи модуля Exchange_bitstamp:

- Account_balance() – метод, який повертає інформацію про поточний баланс користувача та рівень його комісій.
- New_order() – Основний метод використовуючи який можна створювати ордера і торгувати на біржі.
- Replace_order() – Метод заміни старого ордеру на новий.
- Order_status(order_id) – Повертає інформацію про ордер.
- Open_orders() – Повертає список активних ордерів.
- Cancel_order(order_id) – Метод призначений для скасування ордеру.
- Cancel_all_orders(order_id) – Метод призначений для скасування всіх ордерів.
- User_transactions() – Метод повертає історію угод.
- Order_book() – Метод повертає «стакан» біржі.
- Ticker() – Повертає загальну інформацію за останню добу.
- Transactions() – Метод повертає поточні проведені торги на біржі.

Rules Script це спрощена мова програмування, яка дозволяє створювати складні стратегії і повністю автоматизувати торгівлю. Є повна підтримка JavaScript функцій і методів.

Події

Подія ця функція яка пов'язана з індикатором і викликається в момент зміни значення індикатора.

У кожної події є три змінні локальної дії. Це: symbol, name, value.

- Symbol – це пара, символ прикладом якої є текст "BTCUSD".
- Name – код індикатора, текстове назву, приклад "LastPrice".
- Value – дробове значення яке містить останнє значення індикатора.

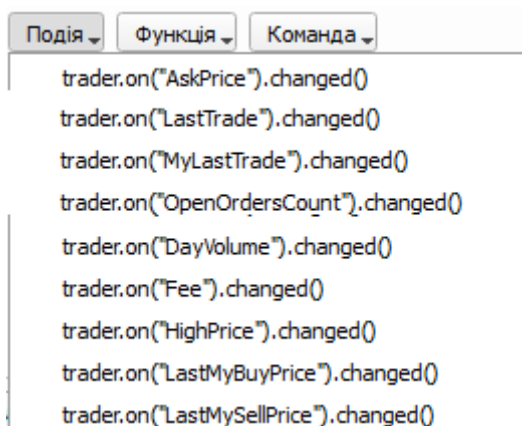


Рисунок 4.7 Події

Функції

Функція це метод, який повертає значення індикатора або яких-небудь обчислень.

Можливо дістати значення будь-якого індикатора використовуючи метод "get".

Функції які повертають значення індикаторів підтримують два аргументи, перший це пара (символ), а другий назва індикатора.

Список функцій які повертають значення індикаторів:

- trader.get("Balance","BTC") – Цей метод повертає баланс по заданій валюті.
- trader.get("10MinVolume") – Обсяг за останні 10 хвилин.
-

- trader.get("ApiLag") – це значення в секундах яке є проміжком часу між відправленням запиту й прийомом результату.
- trader.get("AskPrice") – Ціна пропозиції маркету.
- trader.get("BidPrice") – Ціна попиту маркету.
- trader.get("DayVolume") – Обсяг торгів за останні 24 години.
- trader.get("Fee") – Комісія аккаунта.
- trader.get("HighPrice") – Максимальна ціна за останні 24 години.
- trader.get("LastMyBuyPrice") – Ціна останньої покупки аккаунта.
- trader.get("LastMySellPrice") – Ціна останньої продажі аккаунта.
- trader.get("LastPrice") – Остання ціна маркету.
- trader.get("LowPrice") – Мінімальна ціна за останні 24 години.

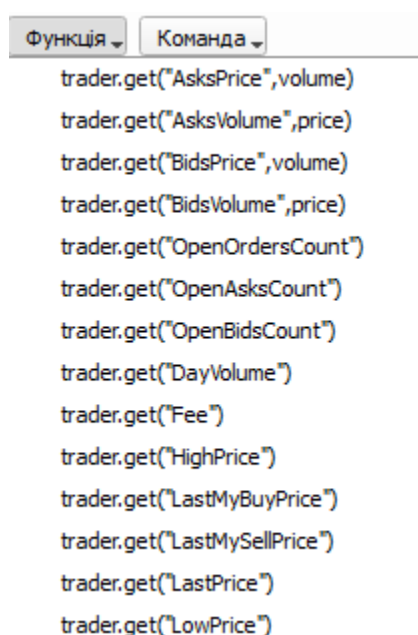


Рисунок 4.8 Функції

Команди

Торгові команди це купівля, продаж та скасування ордерів. Для будь-якого з методів може використовуватися додатковий перший параметр symbol, який вказує в який конкретно парі треба відкрити ордер.

- trader.sell(обсяг, ціна) – Продати зазначений обсяг за вказаною ціною.

- `trader.buy(обсяг,ціна)` – Купити зазначений обсяг за вказаною ціною.
- `trader.cancelOrders()` – Скасовує всі відкриті ордера.
- `trader.cancelAsks()` – Скасовує всі ордера на продаж.
- `trader.cancelBids()` – Скасовує всі ордера на купівлю.

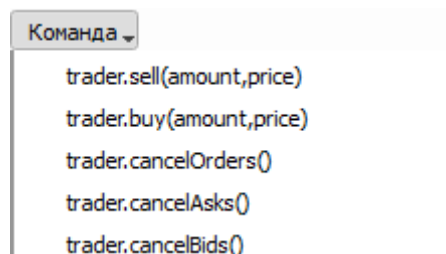


Рисунок 4.9 Команди

4.3 Алгоритм виконання правил

Алгоритм — послідовність, система, набір систематизованих правил виконання обчислювального процесу, що обов'язково приводить до розв'язання певного класу задач після скінченного числа операцій. При написанні комп'ютерних програм алгоритм описує логічну послідовність операцій. Для візуального зображення алгоритмів часто використовують блок-схеми.

У цьому підрозділі буде розглянутий основний алгоритм програми, а саме: алгоритм виконання правил.

«Коли Остання ціна (BTC/USD) < Точне значення (BTC/USD) 220 та Баланс USD > Точне значення (BTC/USD) 300 тоді Купити BTC \$1 - комісія за Остання ціна»

Алгоритм виконання прикладу правила зображено на рисунку 4.10

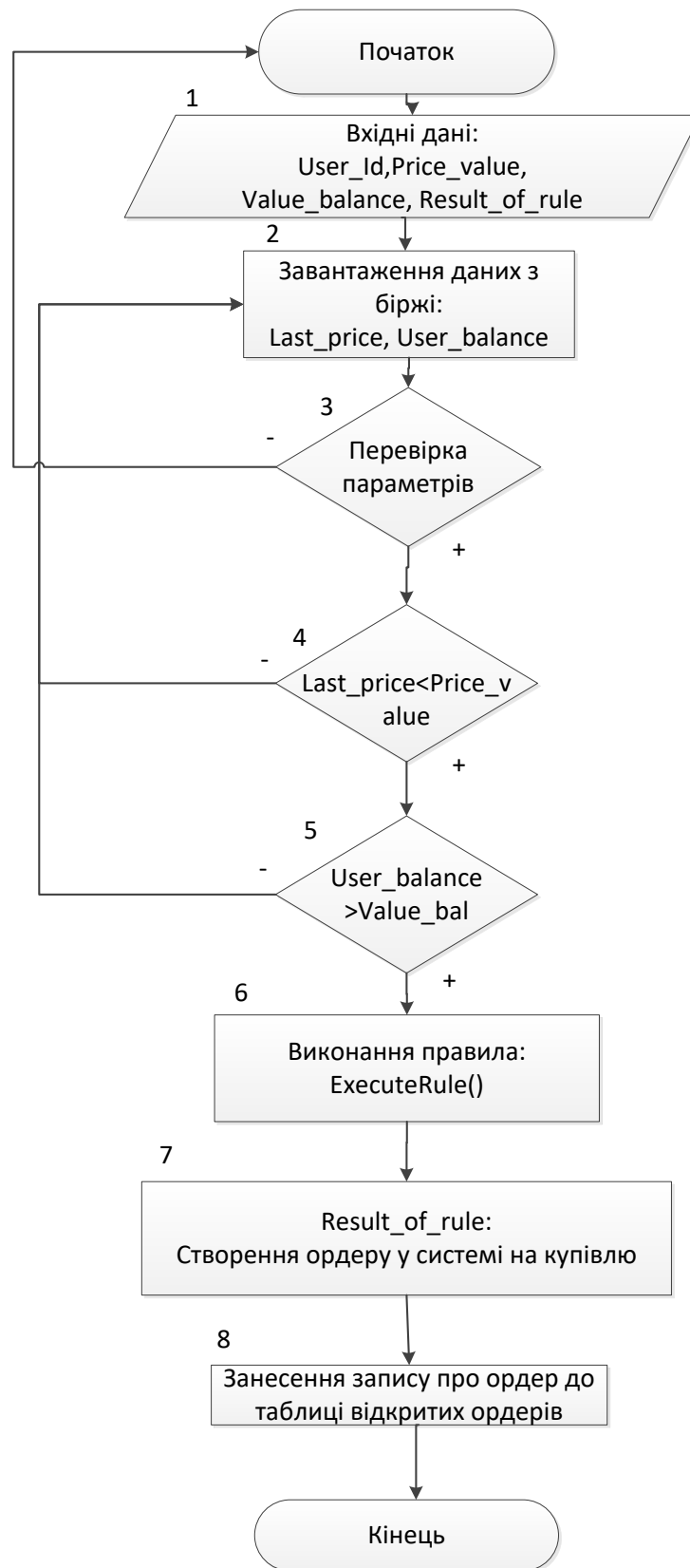


Рисунок 4.10 Схема алгоритму виконання правила

Опис алгоритму «Виконання правила»

1. Спочатку отримуємо вхідні дані:
User_Id(номер користувача), Price_value(потрібна ціна для виконання правила), Balance_value (потрібний баланс користувача для виконання правила), Result_or_rule(результат, який потрібно виконати після співпадання всіх параметрів).
2. Далі проводимо завантаження потрібних числових параметрів з біржі:
Last_price(Остання ціна маркету), User_balance(Поточний баланс користувача).
3. Далі проводимо перевірку вхідних даних, якщо якийсь параметр не пройде її, то завершуємо роботу, якщо все в порядку то продовжуємо.
4. Якщо остання ціна Last_price менша, ніж заданий параметр Price_value то виконуємо наступну перевірку, інакше повертаємось до завантаження нових даних з біржі(пункт 2).
5. Якщо баланс користувача User_balance більший, ніж заданий параметр Balance_value то переходимо до виконання правила, інакше повертаємось до завантаження нових даних з біржі(пункт 2)..
6. Виконуємо правило командою ExecuteRule().
7. Результатом виконання правила в цьому випадку являється створення ордеру у системі на купівлю.
8. Заносимо запис про створений ордер у таблицю відкритих ордерів.

**РОЗДІЛ 5 ПРИКЛАД ВИКОРИСТАННЯ ПІДСИСТЕМИ ТОРГІВ
ЕЛЕКТРОННОЇ БІРЖІ**

Як зазначалося раніше (див. розділ 4), підсистема торгів електронної біржі складається з 3 частин: GUI, API requests та Rules Script. API requests частина не буде розглядатися, так як лише відповідає за взаємодію з біржовими даними, які і відображає GUI частина. Тому будуть детально розглянуті частини 1 та 3.

5.1 Розгляд роботи GUI підсистеми

GUI - це головний механізм взаємодії користувача та торгової платформи, саме у цій частині створюються ордери та правила, відображаються дані користувача та завантажується поточна інформація по маркету(Історія, «стакан» та ін.).

Стартове вікно зображено на рисунку 5.1

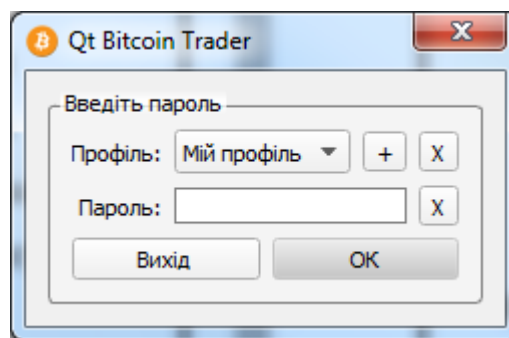


Рисунок 5.1 Стартове вікно

На стартовому вікні користувача запрошується обрати потрібний профіль та ввести пароль.

Кнопка «+» викликає функцію додавання нового профілю користувача, тут йому пропонується обрати одну з доступних в програмі бірж.

Вікно вибору бірж зображено на рисунку 5.2

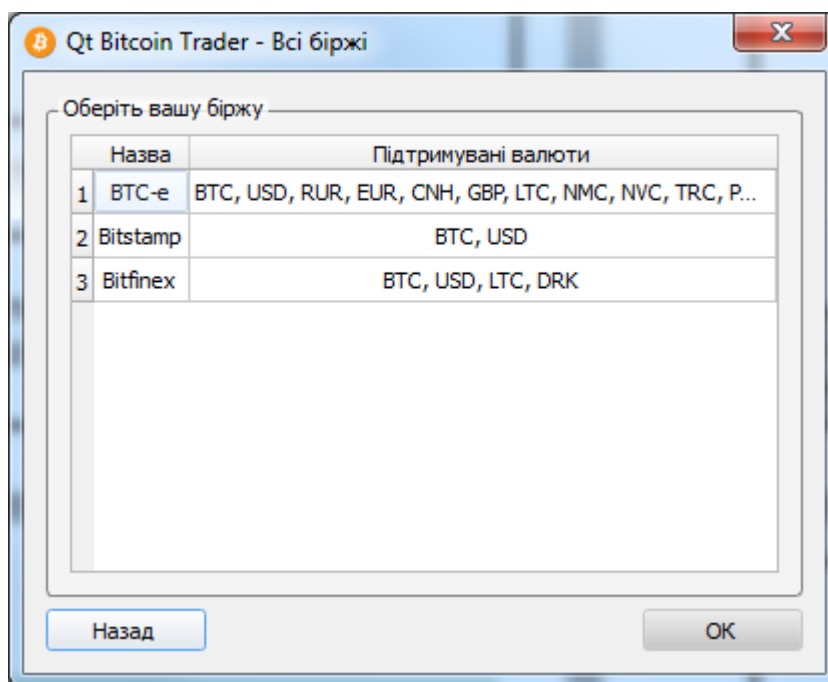


Рисунок 5.2 Вікно вибору біржі

Після обрання потрібної користувачеві біржі, з'являється вікно для додавання нового профілю. Потрібно ввести спеціальні ключі (API та Секрет) для авторизованої роботи з біржою. Для безпечної роботи, користувач має створити новий пароль, яким будуть зашифровані його ключі. В подальшому цей пароль буде використовуватися для авторизації в програмі. Також тут користувач задає бажане ім'я профілю. Форма створення профілю зображена на рисунку 5.3.

The screenshot shows a window titled "Qt Bitcoin Trader" with a close button (X) in the top right corner. The window is divided into two main sections:

- BTC-e API Ключ і Секрет:** This section contains a message: "Ваш API Ключ і Секрет зашифровано за допомогою AES256." Below this are two input fields: "API Ключ:" and "Секрет:", each with a small 'X' button to its right. At the bottom of this section is a large "BTC-e" logo and a button labeled "Отримати нові API ключі".
- Новий пароль:** This section contains a message: "Довжина паролю має бути не менше 8 символів, містити букви, цифри, та спец символи." Below this are three input fields: "Пароль:", "Підтвердження:", and "Ім'я профілю:", each with a small 'X' button to its right.

At the bottom of the window, there are two buttons: "Закрити" (Close) on the left and "ОК" on the right.

Рисунок 5.3 Форма створення профілю

Після створення нового профілю та його підтвердження, програма автоматично перенаправляє нас до стартового вікна де вже буде відображатися наш створений профіль(Рисунок 5.4). Тут потрібно ввести пароль, який ми створили на попередньому кроці. Після успішної авторизації відкриється головне вікно програми(Рисунок 5.5).

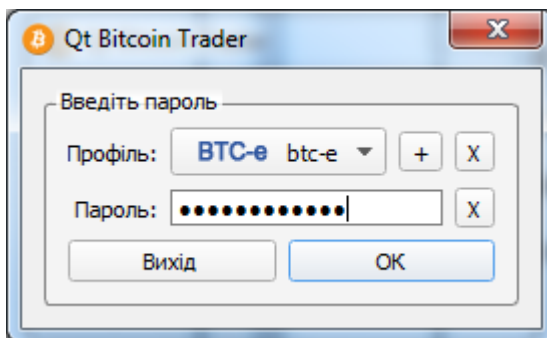


Рисунок 5.4 Стартове вікно зі створеним профілем

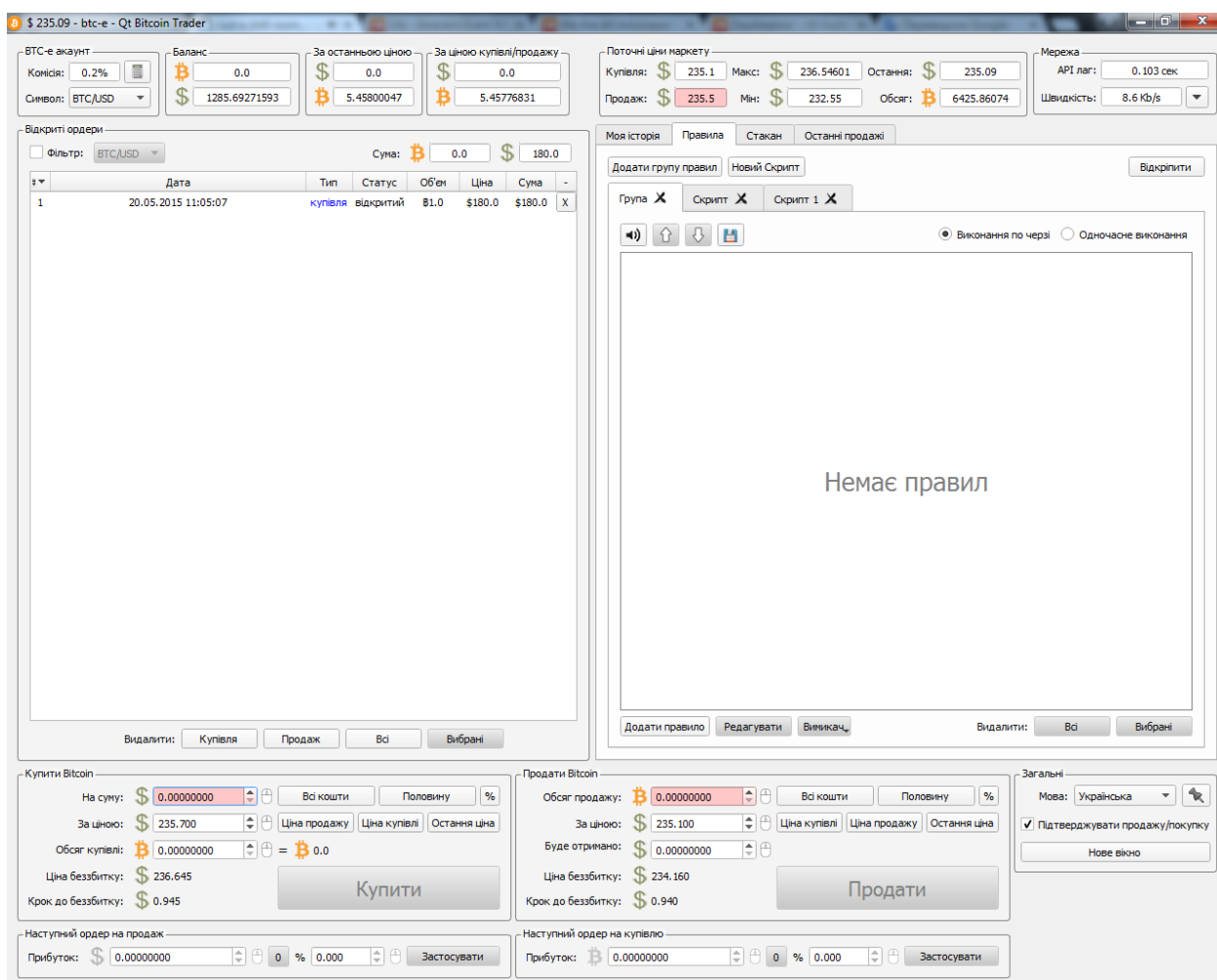


Рисунок 5.5 Головне вікно програми

Головне вікно програми складається з дев'яти модулів:

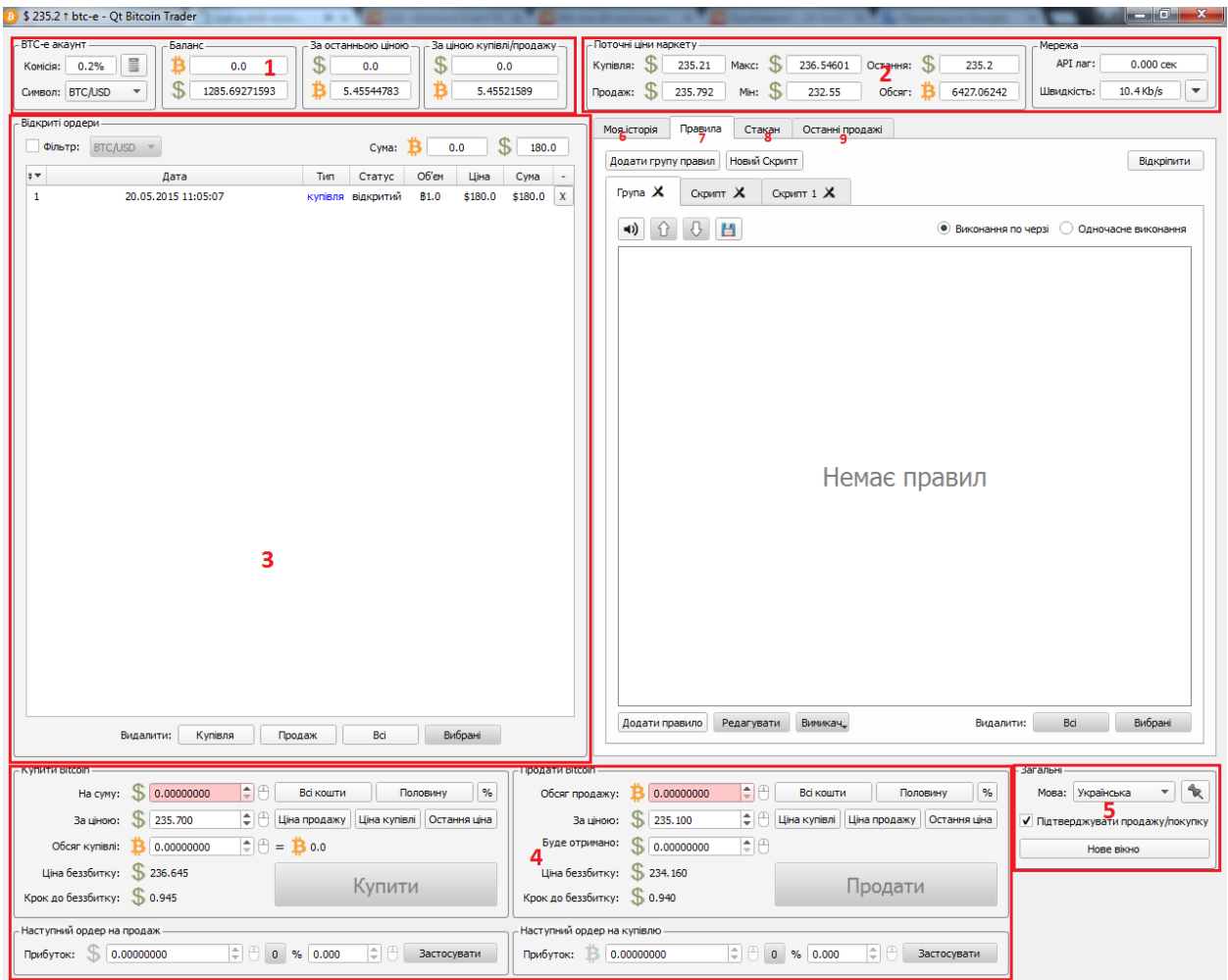


Рисунок 5.6 Модулі програми

- 1) Тут відображається основна інформація про акаунт користувача.
- 2) Блок з поточними даними маркету біржі.
- 3) Блок з усіма виставленими ордерами користувача.
- 4) Основний блок для здійснення операції купівлю чи продажу.
- 5) Загальний блок налаштувань та доступу до декількох вікон.
- 6) Блок з історією проведених торгів користувача.
- 7) Блок створення правил та стратегій для ефективної торгівлі.
- 8) Блок, що відображає «стакан» маркету.
- 9) Блок з останніми за певний період операціями купівлі/продажу на біржі.

Далі розглянемо кожен з блоків більш детально.

У першому блоці знаходиться інформація про баланс користувача, валютна пара для торгів, розрахунковий еквівалент у другій валюті пари та комісія.

BTC-е акаунт		Баланс		За останньою ціною		За ціною купівлі/продажу	
Комісія:	0.2%	Bitcoin	0.0	USD	0.0	USD	0.0
Символ:	BTC/USD	USD	1285.69271593	Bitcoin	5.44456796	Bitcoin	5.44456796

Рисунок 5.7 Блок про інформацію акаунта

Також тут же знаходиться кнопка для виклику калькулятора прибутку(Рис 5.8):

Купити		За ціною		На суму		Комісія		Буде отримано	
Bitcoin	5.47098767	USD	235.001	USD	1285.68757344	Bitcoin	0.01094198	Bitcoin	5.46004569
Продати	5.46004569	USD	235.942	USD	1288.25410019	USD	2.57650820	USD	1285.67759199
Комісія:	0.20 %	Ціна беззбитку		<input type="checkbox"/> Поверх всіх вікон		<input checked="" type="checkbox"/> Тільки одне вікно		Прибуток	
								Bitcoin	-0.00998145

Рисунок 5.8 Калькулятор прибутку

В калькуляторі прибутку розраховується рівень прибутку при виконанні пари операцій купівля/продаж. Користувач задає потрібні йому параметри у полях «Купити», «За ціною», «На суму», «Буде отримано» та у рядку продажі лише «За ціною», всі інші параметри будуть підраховані автоматично та користувач отримає розрахунковий рівень прибутку у полі «Прибуток».

Якщо користувач хоче дізнатись лише рівень ціни для беззбиткового продажу потрібно тільки натиснути на кнопку «Ціна беззбитку» та поле «За ціною» у продажі буде розраховано автоматично для нульового прибутку.

У другому блоці розташовані інформаційні елементи маркету біржі та інформація про мережу.

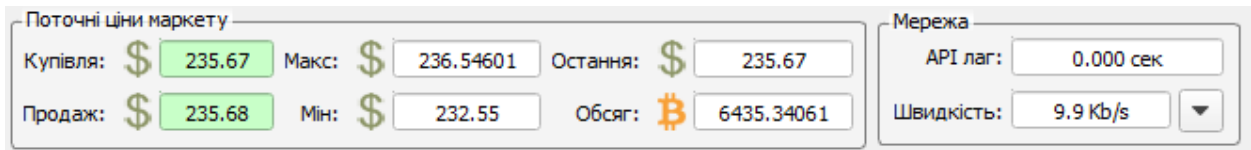


Рисунок 5.9 Блок з поточною інформацією по маркету та мережі

Розглянемо детальніше блок «Мережа». Тут знаходиться два поля: API лаг та швидкість. API лаг відображається для того, щоб користувач міг знати коли запити до біржі затримуються для оновлення. Поле «Швидкість» же відображає поточну кількість даних, що передаються між клієнтом та сервером, це поле має чисто інформативний характер.

У наступному, третьому блоці, відображаються всі поточні ордери користувача(Рисунок 5.10).

У верхній частині блоку користувач має можливість за бажанням застосувати фільтр для відображення тільки по окремій парі. Також тут відображається загальна сума виставлена на купівлю/продаж по усім ордерам.

У нижній частині користувач має можливість масово видаляти непотрібні ордери за тим чи іншим параметром, чи просто обрати ордери та видалити лише вибрані.

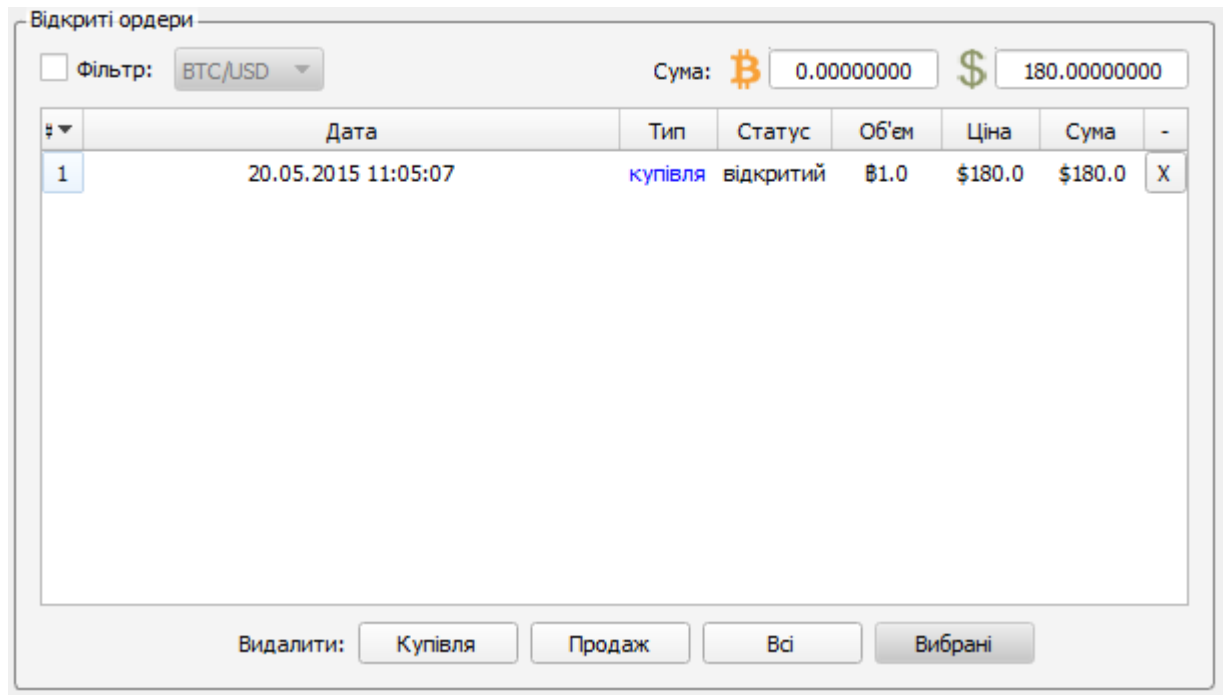


Рисунок 5.10 Блок з відкрити ордерами

Четвертий блок відповідає за створення ордерів на купівлю чи продаж. Детальніше зображено на рисунку 5.11.

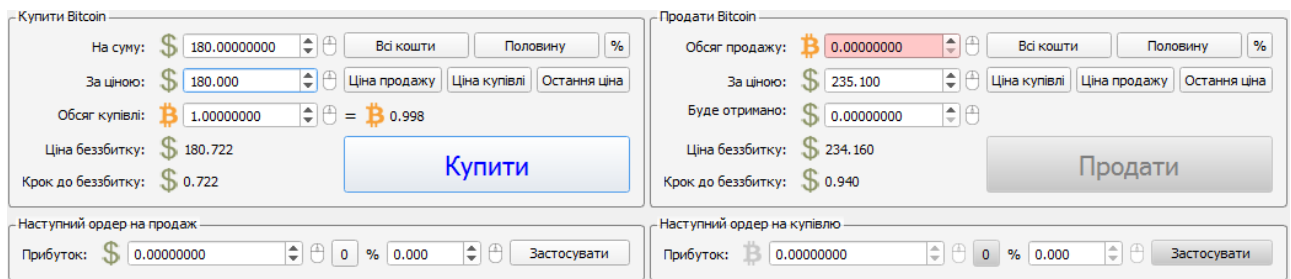


Рисунок 5.11 Блок створення ордерів

У даному блоці користувачу надається зручний метод створення ордерів за допомогою допоміжних кнопок, які в залежності від поточної інформації про маркет чи баланс користувача змінюють відповідні поля для купівлі/продажу.

Також тут знаходяться блоки «Наступний ордер», що дозволяють користувачу одразу створювати додатковий ордер на зворотну операцію з бажаним рівнем прибутку у відсотковому чи валютному еквіваленті.

П'ятий, загальний блок, відповідає за такі параметри та функції як:

- Локалізація
- Нове вікно
- Підтвердження операцій

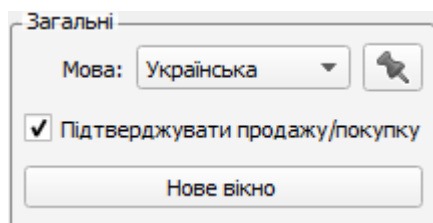


Рисунок 5.12 Блок загальних налаштувань

Локалізація доступна на трьох мовах (українська, російська та англійська). Кнопка «Нове вікно» запускає ще один екземпляр програми, проте використовувати вже активний профіль користувачу неможливо.

Шостий блок «Моя історія» відображає всі проведені операції користувача з моменту реєстрації на біржі (не в програмі). Детально блок зображено на рисунку 5.13.

У верхній частині блоку винесено два інформативні поля з цінами останніх здійснених купівель та продажів.

Дата	Об'єм	Тип	Ціна	Сума
20:14:17	฿0.642976	продано	\$233.7	150.2634912
07.05.2015 20:14:17	฿3.95702	продано	\$233.7	924.755574
2:53:59	฿0.69689	продано	\$213.0	148.43757
2:53:50	฿1.1925	продано	\$213.0	254.0025
2:52:59	฿1.2166	продано	\$213.0	259.1358
2:52:14	฿0.0110906	продано	\$213.021	2.3625307
2:52:14	฿0.882922	продано	\$213.2	188.2389704
2:26:22	฿0.00072235	продано	\$214.0	0.1545829
2:26:22	฿0.079708	продано	\$214.0	17.057512
2:26:22	฿0.0110173	продано	\$214.0	2.3577022
2:26:22	฿0.0115065	продано	\$214.0	2.462391
2:26:22	฿0.0108103	продано	\$214.0	2.3134042
2:26:22	฿0.011178	продано	\$214.0	2.392092
2:26:22	฿0.021758	продано	\$214.0	4.656212
2:26:22	฿0.0103821	продано	\$214.0	2.2217694
2:26:22	฿0.0276187	продано	\$214.0	5.9104018
2:26:22	฿0.0146478	продано	\$214.0	3.1346292
2:26:22	฿0.0213149	продано	\$214.0	4.5613886
2:26:22	฿0.020294	продано	\$214.0	4.342916

Рисунок 5.13 Блок з історією користувача

У наступному блоці «Правила» (Рисунок 5.14) користувач може створювати спеціальні правила при виконанні умов яких, здійснюється та чи інша операція.

Для створення нового правила потрібно спочатку натиснути кнопку «Додати групу правил». З'явиться вікно створення групи (Рис 5.15), де користувач може обрати як порожню групу, так і завантажити в неї правила з інших груп, файлів чи шаблонів.

Після створення групи тепер можливо додавати в неї користувацькі правила. Для цього потрібно натиснути кнопку «Додати правило» та з'явиться форма, в якій користувач обирає потрібні параметри майбутнього правила (Рисунок 5.16).

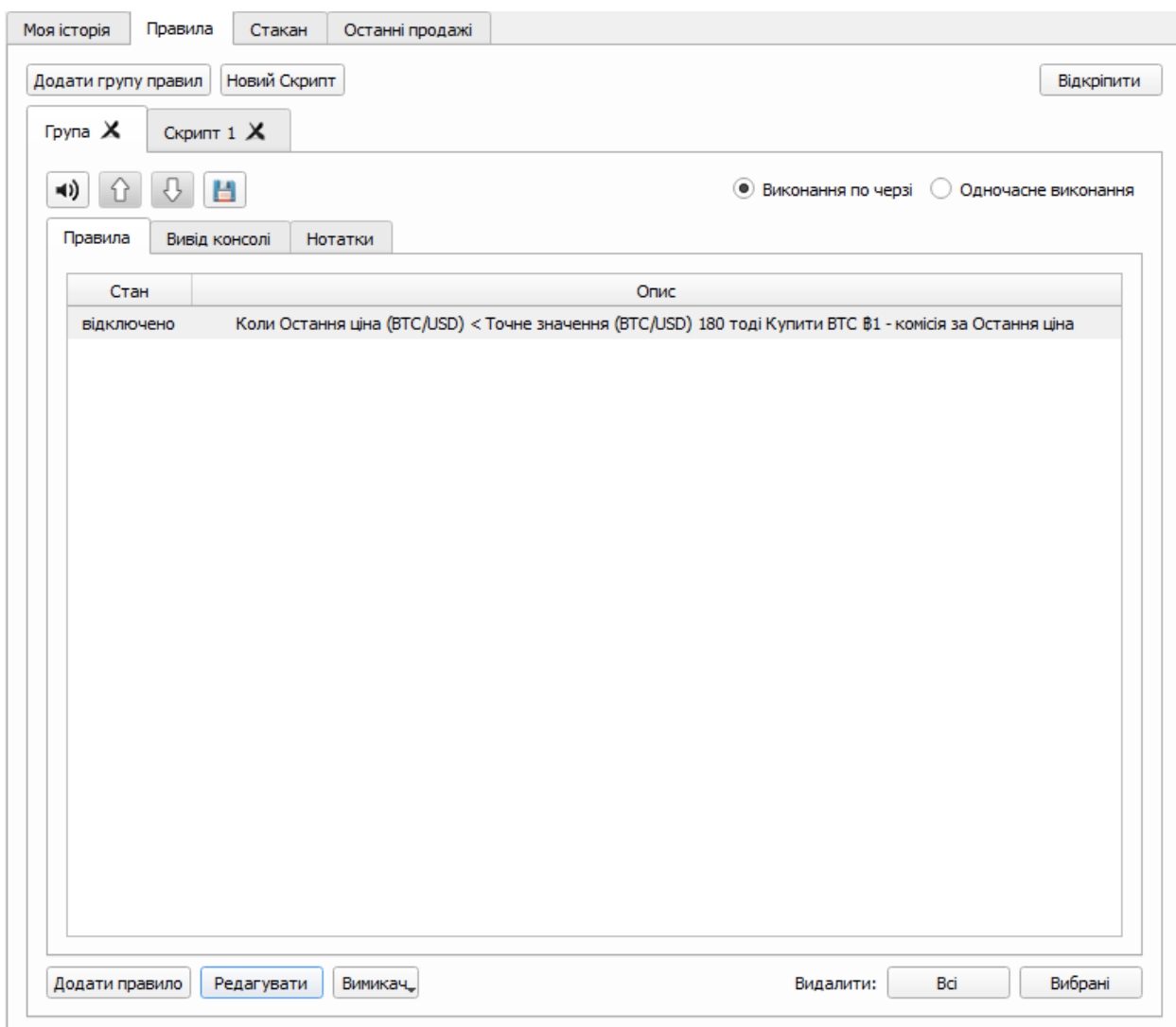


Рисунок 5.14 Блок правил

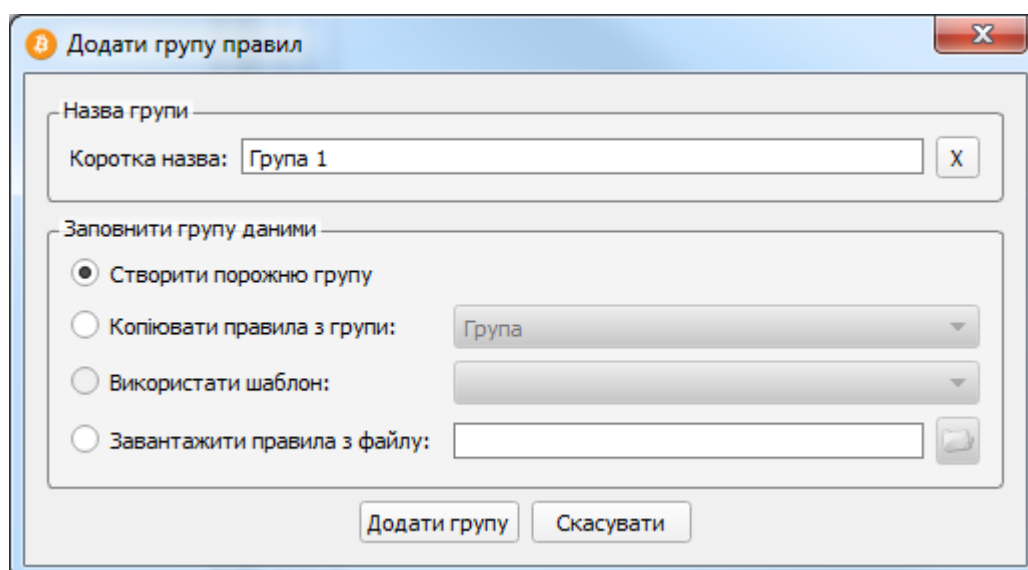


Рисунок 5.16 Додавання групи правил

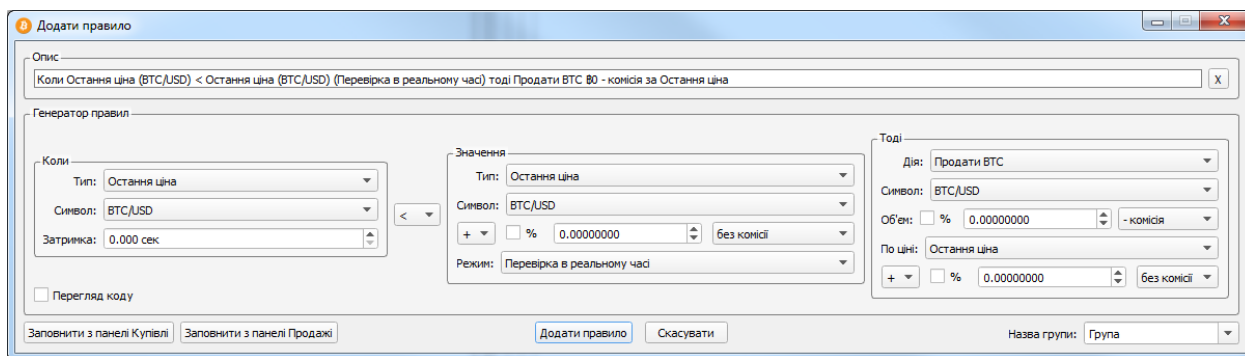


Рисунок 5.17 Форма додавання правила

В генераторі правил влаштовані такі типи умов та дій:

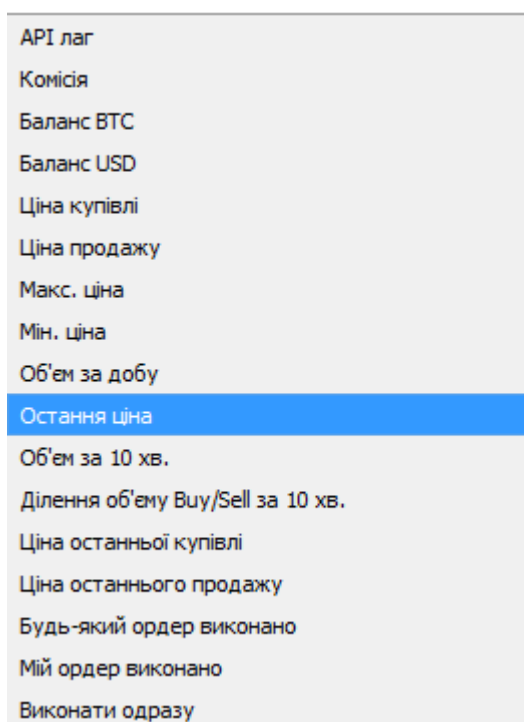


Рисунок 5.18 Умови правил

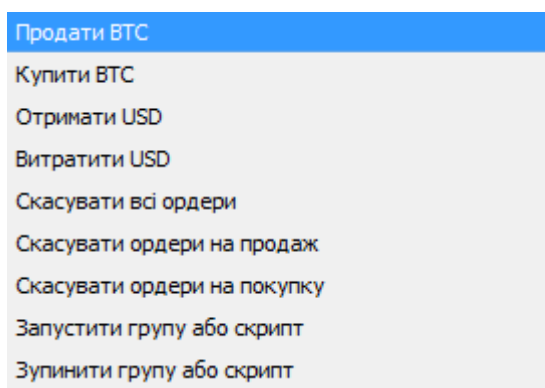


Рисунок 5.19 Дії правил

Додані правила з'являються у головну блоці (Рисунок 5.14), де користувач має можливість задавати спосіб їх виконання, редагування, включення/відключення, перегляд консолі з інформацією про стан виконання правил. Також у вкладці «Нотатки» користувач може залишати власні будь-які відмітки.

Блок «Стакан»

Моя історія | Правила | **Стакан** | Останні продажі

API лаг: 0.971 сек

Ордери на продаж ↔ Ордери на купівлю

Відкріпити

Сума ₴	↑↓	Об'єм ₴	Ціна \$	Ціна \$	Об'єм ₴	↑↓	Сума ₴
0.012		0.01200000	235.528	235.324	0.19163000		0.19163
0.026		0.01400000	235.634	235.317	0.01200000		0.20363
3.356		3.33000000	235.792	235.313	0.01400000		0.21763
3.37434609		0.01834609	235.97	235.3	1.00000000		1.21763
5.13134609		1.75700000	235.98	235.241	514.86076059		516.07839059
19.0606209	↓	13.92927481	236.0	235.2	9.68393031	↓	525.7623209
19.5606209		0.50000000	236.01	235.19	15.12000000		540.8823209
20.0606209		0.50000000	236.02	235.18	0.01100000		540.8933209
20.0861209		0.02550000	236.117	235.063	0.01100000		540.9043209
25.5591209		5.47300000	236.168	235.048	0.01100000	↓	540.9153209
26.17685513		0.61773423	236.2	235.029	0.01400000		540.9293209
26.18685613		0.01000100	236.211	234.998	0.01100000		540.9403209
26.21285613		0.02600000	236.212	234.992	0.01400000		540.9543209
26.22585613		0.01300000	236.24	234.978	0.01300000		540.9673209
26.98561909		0.75976296	236.25	234.957	0.01100000		540.9783209
26.99861909		0.01300000	236.256	234.955	0.09957498		541.07789588
28.03296452	↑	1.03434543	236.259	234.939	0.05106000	↑	541.12895588
28.04296452		0.01000000	236.3	234.937	0.01400000		541.14295588
29.83696452		1.79400000	236.357	234.913	0.01400000		541.15695588
29.84996452		0.01300000	236.367	234.908	0.01400000		541.17095588
29.86196452		0.01200000	236.379	234.895	0.01300000		541.18395588
30.08896452		0.22700000	236.389	234.886	0.01400000		541.19795588
36.65096452		6.56200000	236.39	234.866	0.01200000		541.20995588
37.65096452		1.00000000	236.399	234.864	0.01200000		541.22195588
37.66396452		0.01300000	236.407	234.858	0.01400000		541.23595588
37.67696452		0.01300000	236.415	234.856	0.01300000		541.24895588

Крок групування: \$ Нема Ширини стовбців авто. Ліміт рядків: 100

Рисунок 5.20 Блок зі «стаканом» біржі

У восьмому блоці «стакан» біржі відображаються всі поточні активні ордери маркету (не тільки користувача). Окрім стандартного перегляду, користувачеві надається можливість задати потрібний йому «крок групування» при якому ордери, що знаходяться на меншому відрізку ціни групуються в один. Також тут присутнє поле API лаг, яке навідміну від поля мережевого блоку відображає затримку інформації зі сторони біржі.

Дев'ятий блок «Останні продажі»

Моя історія		Правила		Стакан		Останні продажі	
<input checked="" type="checkbox"/>	Авто прокрутка	Об'єм за 10 хв:	Bitcoin	10.91	↑	15.58 %	Відкрити
Дата	Об'єм B	Тип	Ціна \$	Сума \$			
21.05.2015 15:03:54	0.10438	купівля	↑ 235.69	24.6013222			
21.05.2015 15:03:52	0.1	продаж	↓ 235.321	23.5321			
21.05.2015 15:03:44	0.014	продаж	↑ 235.444	3.296216			
21.05.2015 15:03:37	0.1	продаж	↓ 235.321	23.5321			
21.05.2015 15:03:29	0.1	продаж	235.509	23.5509			
21.05.2015 15:03:22	0.1	продаж	↑ 235.509	23.5509			
21.05.2015 15:03:14	0.1	продаж	235.2	23.52			
21.05.2015 15:03:07	0.1	продаж	↓ 235.2	23.52			
21.05.2015 15:03:01	0.11	продаж	235.25	25.8775			
15:02:59	0.1	продаж	↑ 235.25	23.525			
21.05.2015 15:02:52	0.1	продаж	235.2	23.52			
21.05.2015 15:02:45	0.1	продаж	235.2	23.52			
21.05.2015 15:02:43	0.131841	продаж	235.2	31.0090032			
21.05.2015 15:02:37	0.1	продаж	235.2	23.52			
21.05.2015 15:02:30	0.1	продаж	235.2	23.52			
21.05.2015 15:02:23	0.1	продаж	235.2	23.52			
21.05.2015 15:02:15	0.1	продаж	↓ 235.2	23.52			
21.05.2015 15:02:08	0.039795	продаж	235.664	9.37824888			
21.05.2015 15:02:01	0.1	продаж	235.664	23.5664			
21.05.2015 15:01:53	0.1	продаж	235.664	23.5664			
21.05.2015 15:01:49	0.012	купівля	↑ 235.664	2.827968			
15:01:49	0.013	купівля	↑ 235.509	3.061617			
15:01:47	0.1	продаж	235.2	23.52			
21.05.2015 15:01:40	0.1	продаж	↓ 235.2	23.52			
21.05.2015 15:01:32	0.0371096	продаж	235.506	8.73953346			
21.05.2015 15:01:25	0.1	продаж	235.506	23.5506			
21.05.2015 15:01:21	0.00176562	купівля	↑ 235.506	0.4158141			

Рисунок 5.21 Блок останніх продажів

В блоці «Останні продажі» відображаються всі проведені операції купівлі/продажу за останні 10 хвилин. Додатково тут показується розрахунковий обсяг за 10 хвилин та відношення операцій продажу до купівлі.

Слід зазначити, що блоки 6, 7, 8 та 9 мають можливість бути «відкріпленими» користувачем для розташування в довільному порядку на екрані. Приклад зображено на рисунку 5.22.

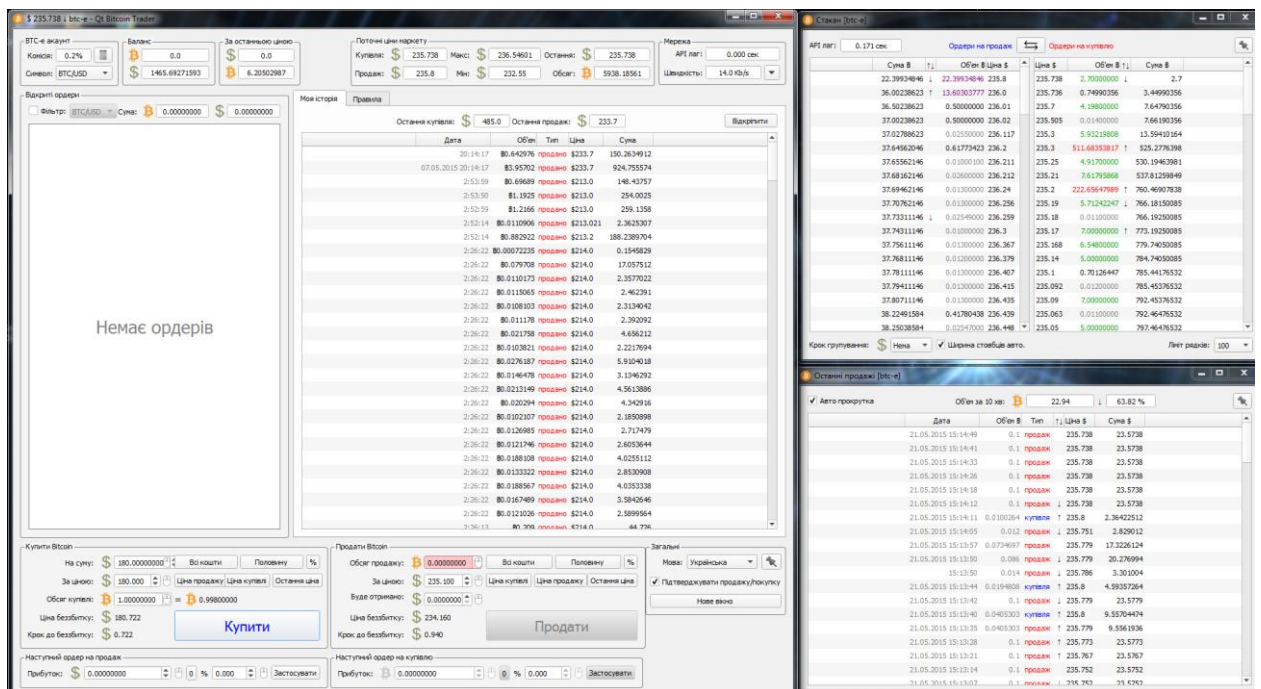


Рисунок 5.22 Довільне розташування блоків

5.2 Розгляд та приклад роботи Rules Script

Rules Script – це спеціальна мова влаштована в програму для створення складних стратегій, індикаторів та ін.

Вікно створення скриптів знаходиться у блоці «Правила» (Рисунок 5.23). Для створення нового скрипту потрібно натиснути кнопку «Новий скрипт» після чого з'явиться спеціальне вікно (Рисунок 5.24).

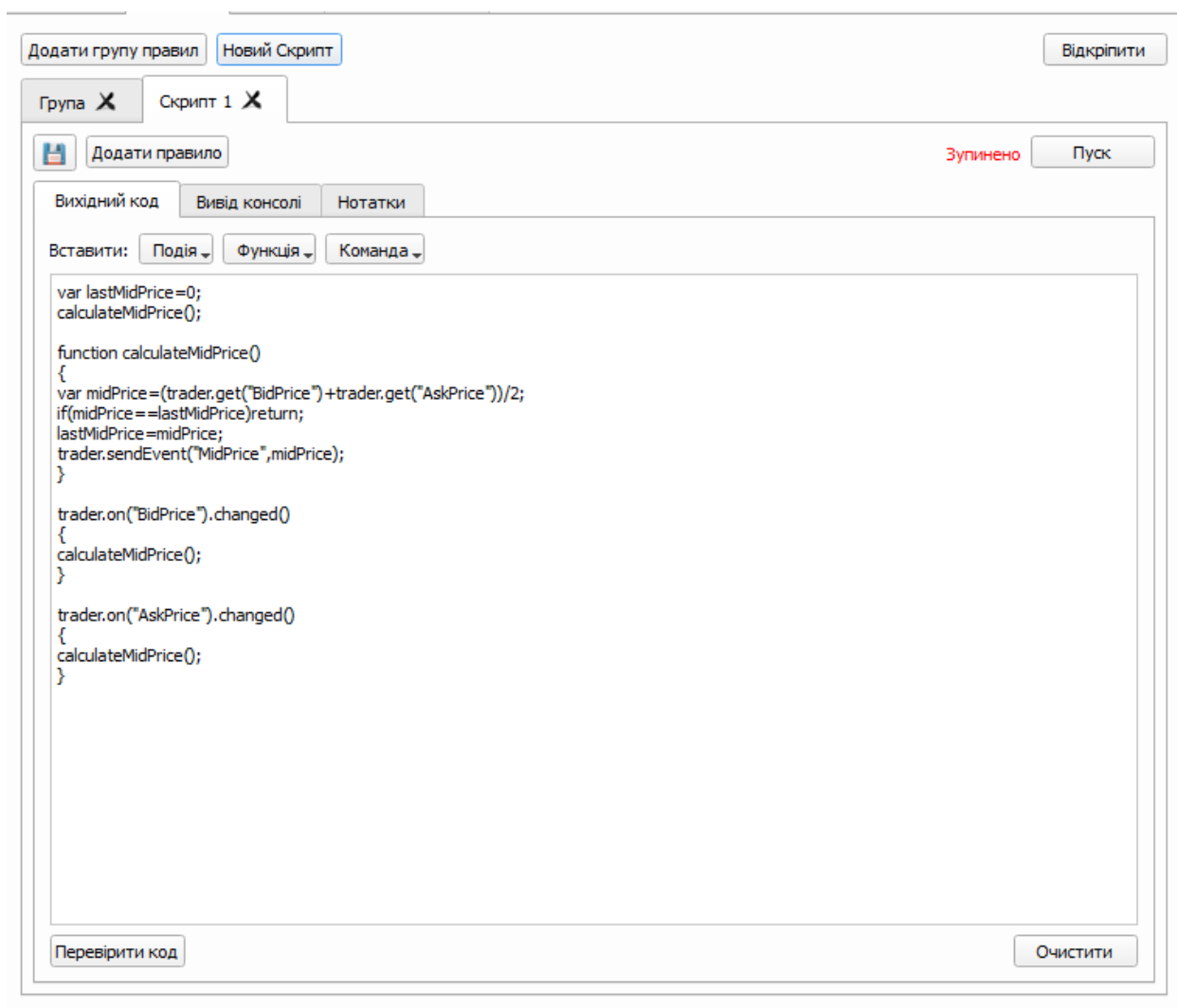


Рисунок 5.23 Блок скриптів

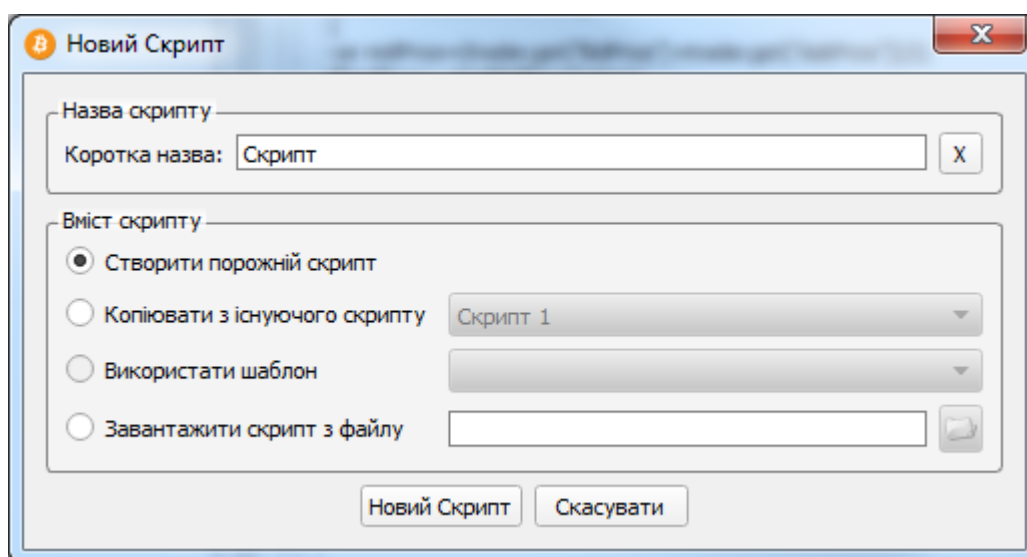


Рисунок 5.24 Вікно нового скрипту

Після створення, у вкладці «Вихідний код» користувач може починати розробку свого функціоналу скрипту. Приклад наведений на рисунку 5.25.

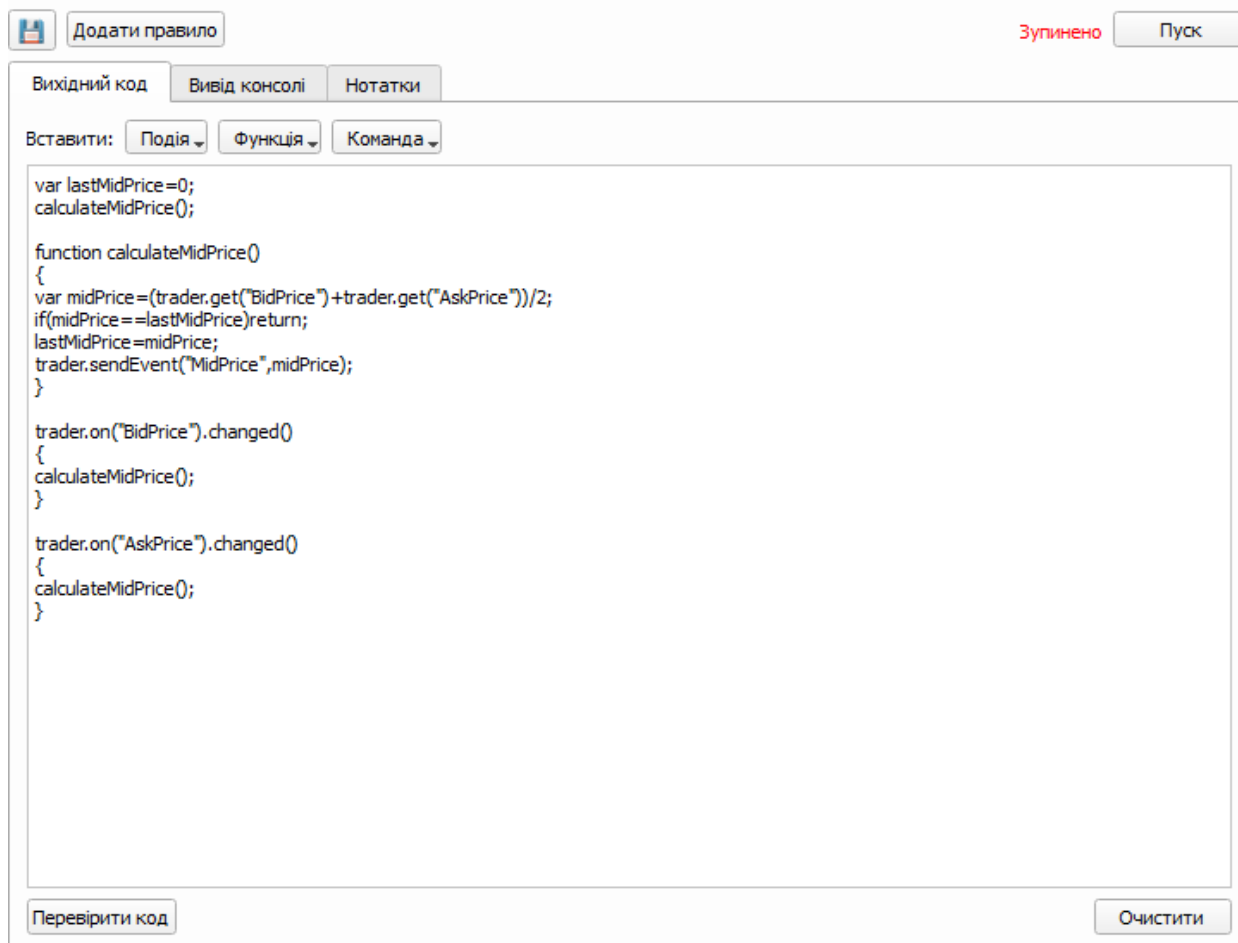


Рисунок 5.25 Приклад скрипту

Коли користувач закінчить написання свого скрипту, він має можливість перевірити працездатність свого коду кнопкою «Перевірити код», яка в разі помилок сповістить про це.

Вкладка «Вивід консолі» відображає будь-яку інформацію, яку користувач запросив командою `trader.log()`. Приклад виводу зображено на рисунку 5.26.

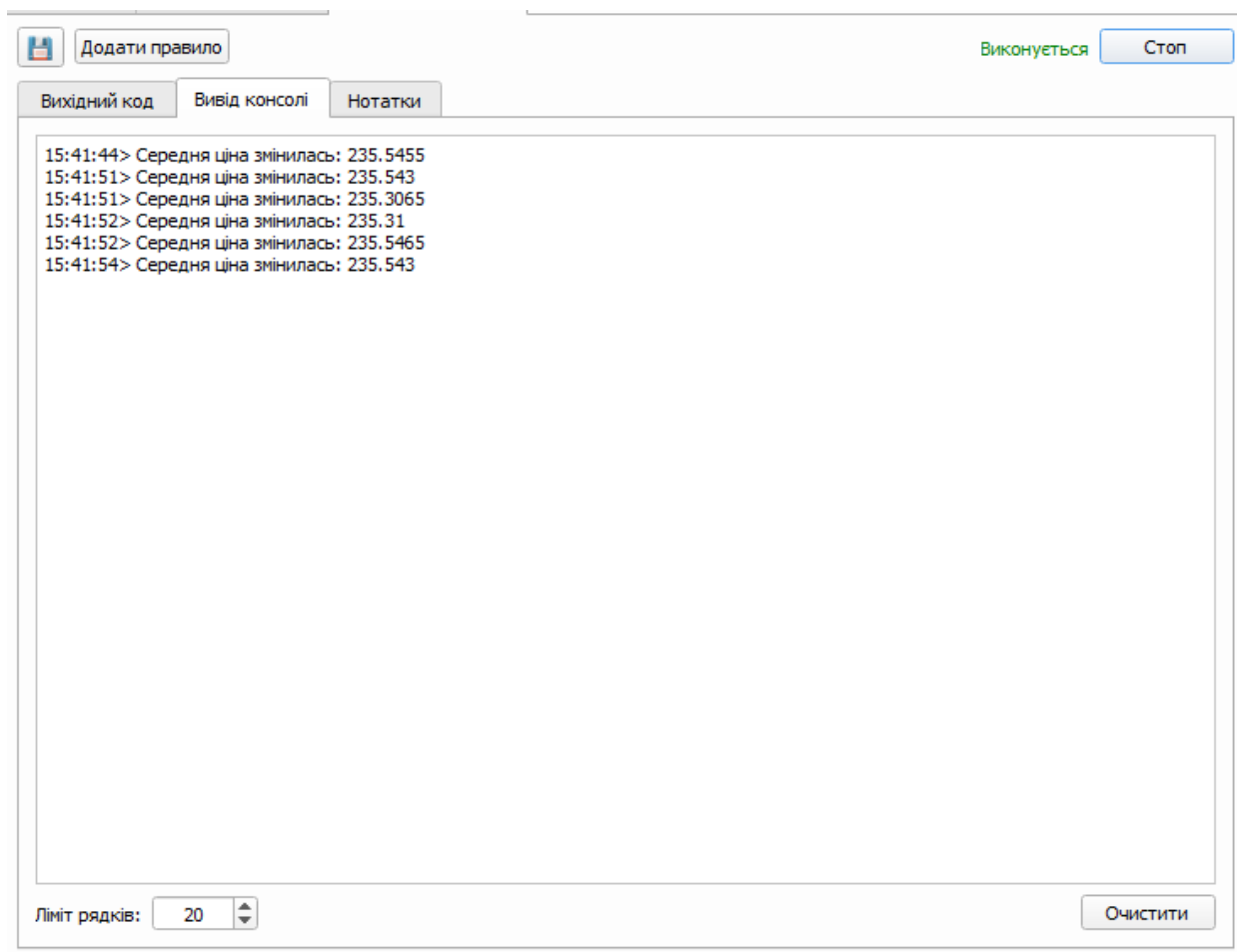


Рисунок 5.26 Приклад виводу консолі

СПИСОК ЛІТЕРАТУРИ

1. Бланшет, Саммерфилд - Qt4 Программирование GUI на C++. 2ed. - 2008— 362 с.
2. Андрей Боровский - Qt4.7+. Практическое программирование на C++ - 2012 - 227 с.
3. Прата С. - Язык программирования C++. Лекции и упражнения - 2011 - 628 с.
4. Макс Шлее - Qt4.5 Профессиональное программирование на C++ - 403 с.
5. Дэвид А. Марка, Клемент Л. МакГоуэн Методология структурного анализа и проектирования SADT. -М.:1993.
6. Гайна Г.А. Основи проектування баз даних. Навч. посібник.- К.:Кондор, 2007.- 208с.
7. Берко А. Ю., Верес О. М. Організація баз даних: практичний курс: Навч. посібн./ За наук. ред. В. В
8. Боггс У., Боггс М. UML и Rational Rose: Пер. с англ. – М.: ”Лори”, 2000 – 582 с.
9. Прохоренок Н.А. Программирование на C++ в Visual Studio® 2010 – 550 с
10. М В Финков, "Пиринговые сети" 2010.
11. The Book of Qt 4: The Art of Building Qt Applications Paperback – July 19, 2007 by Daniel Molkentin
12. Alan Ezust - An Introduction to Design Patterns in C++ with Qt 4 , 2013.- 255с.
13. Peer-to-Peer Harnessing the Power of Disruptive Technologies By Andy Oram 2001, - 448с
14. Жидецький В. Ц. Основи охорони праці : підруч. / В. Ц. Жидецький. — 3-тє вид., перероб. і доп. — Львів : Укр. акад. друкарства, 2006. — 336

15. Дейли Дж. Эффективное ценообразование — основа конкурентного преимущества. — М., 2004.
16. Mastering Bitcoin Unlocking Digital Cryptocurrencies By Andreas M. Antonopoulos, 2014 – 298с