

Міністерство освіти і науки України
Одеський національний технологічний університет
Кафедра комп'ютерної інженерії



**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему Проектування та розробка симулятора роботи
(назва кваліфікаційної роботи згідно наказу ОНТУ)

криптовіржі

Здобувача Курилка А.Д.
(прізвище, ініціали)

4 курсу 541б групи

Керівники: к.т.н., доц. Ненов О.Л.
(посада, прізвище та ініціали)

асист. Колумба І.В.
(посада, прізвище та ініціали)

Консультанти: _____
(посада, прізвище та ініціали)

д.е.н., проф. Басюркіна Н.Й.
(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від 10.06 2023 р., протокол № 8

Завідувач кафедри комп. інженерії _____ Сергій АРТЕМЕНКО
(назва кафедри) (підпис) (Ім'я ПРІЗВИЩЕ)

Одеса - 2023 рік

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерної інженерії, програмування та кіберзахисту
Кафедра комп'ютерної інженерії
Ступінь вищої освіти бакалавр
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма Розробка ігор та інтерактивних медіа у віртуальній реальності

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної інженерії

Сергій АРТЕМЕНКО

« 10 » серпня 2022 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Курілка Артемія Дмитровича

1. Тема роботи Проектування та розробка симулятора роботи криптобіржі

Затверджена наказом університету від « 10 » квітня 2023 р., наказ № 146-03

2 Термін задачі здобувачем закінченої роботи 5 червня 2023 р.

3. Вихідні дані роботи

1. IDE IntelliJ Idea; 2. СУБД MySQL, MongoDB;

3. Language Java8;

4. Spring Framework.

4. Перелік питань, які потрібно розробити

1. Збір та аналіз інформації. 2. Проектування симулятора.

3. Створення симулятора криптобіржі

4. Економічна частина.

5. Охорона праці.

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

Слайд 1. Мета, предмет, ціль. Слайд 2. Основні терміни. Слайд 3. Огляд

Існуючих криптобірж. Слайд 4. Переваги конкурентів. Слайд 5 Переваги проекту.

Слайд 6. Етапи розробки. Слайд 7. Вибір стеку. Слайд 7. Розробка головних функцій.

Слайд 8. Розробка двигуна. Слайд 9. Робота з оредрами. Слайд 10 Висновки.

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Економіка</i>	<i>Басюркіна Н.Й., д.е.н., проф.</i>		
<i>Охорона праці</i>	<i>Колумба І.В., асист.</i>		
<i>Нормоконтроль</i>	<i>Жуковецька С. Л., ст. викл.</i>		

7. Дата видачі завдання 30.09.2022

Керівники

Олексій НСНОВ

Ірина КОЛУМБА

Завдання прийняв до виконання

Артемій КУРІЛКО

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	<i>Дослідження об'єкту криптобірж</i>	<i>20.02.23</i>	
2.	<i>Дослідження технологій для розробки криптобіржі</i>	<i>10.03.23</i>	
3.	<i>Постановка завдання</i>	<i>10.03.23</i>	
4.	<i>Визначення вхідних даних</i>	<i>20.03.23</i>	
5.	<i>Проектування симуляції криптобіржі</i>	<i>25.03.23</i>	
6.	<i>Тестування криптобіржі</i>	<i>15.04.23</i>	
7.	<i>Економічні розрахунки</i>	<i>05.05.23</i>	
8.	<i>Оформлення пояснювальної записки</i>	<i>20.05.23</i>	
9.	<i>Підготовка графічного матеріалу</i>	<i>03.06.23</i>	
10.	<i>Корегування даних</i>	<i>05.06.23</i>	

Керівники роботи

Олексій НСНОВ

Ірина КОЛУМБА

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач - дипломник

Артемій КУРІЛКО

АНОТАЦІЯ

Дипломна робота виконувалась за темою "Проектування та розробка симулятора роботи криптобіржі". Ця робота присвячена створенню програмного інструменту, який дозволить симулювати роботу криптовалютної біржі.

Метою даного проекту є розробка симулятора, здатного моделювати основні аспекти роботи криптобіржі, такі як ордери на купівлю та продаж, склянку ордерів, історію торгів та зміну цін. Симулятор надасть можливість дослідження та тестування різних торгових стратегій, а також дозволить проводити аналіз ринку та оцінку ризиків.

В рамках роботи буде проведено аналіз функціональних та нефункціональних вимог, визначено основні компоненти системи та їх взаємодію, а також спроектовано інтерфейс користувача для зручного використання симулятора. Буде розроблено алгоритми, що забезпечують генерацію випадкових даних, моделювання торгових операцій та оновлення стану ринку. У процесі розробки будуть використані сучасні технології програмування, такі як мова *Java*, фреймворки для розробки веб-застосунків та бази даних для зберігання історії торгів.

Результати цього дослідження можуть бути корисними для трейдерів та інвесторів, дозволяючи їм перевірити та покращити свої стратегії торгівлі на основі реалістичних симуляцій. Також симулятор може бути використаний з освітньою метою, щоб допомогти студентам та новачкам розібратися в принципах роботи криптобірж та практичному застосуванні торгових стратегій.

ANNOTATION

The thesis of the robot was held for the topic "Design and development of the crypto exchange robot simulator". This robot is dedicated to the creation of a software tool that allows the robot to simulate a cryptocurrency exchange.

The method of this project is the development of a simulator, building a simulation of the main aspects of the operation of the crypto exchange, such as orders for the purchase and sale, a bottle of orders, the history of trades and the change of prices. The simulator allows you to carry out and test various trading strategies, as well as allow you to analyze the market and assess risks.

Within the framework of the work, an analysis of functional and non-functional variables will be carried out, the main components of the system and their interaction will be identified, and a core interface for the manual simulator will be designed. Algorithms will be developed to ensure the generation of trading data, the modeling of trading operations and the updating of the market. The distribution process will use state-of-the-art programming technologies, such as Java language, frameworks for the development of web downloads, and databases for collecting trading history.

The results of this study can be significant for traders and investors, allowing them to change and improve their trading strategies based on realistic simulations. Also, the simulator can be used with an illuminating method to help students and beginners learn about the principles of crypto-exchange robots and practical trading strategies.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ЗБІР ТА АНАЛІЗ ІНФОРМАЦІЇ.....	10
1.1 Обґрунтування актуальності дослідження.....	10
1.2 Огляд існуючих криптобірж та їх обмежень.....	16
1.3 Мета та завдання проекту.....	20
1.4 Вибір програмних засобів для реалізації програмного комплексу.....	22
Висновки до першого розділу.....	29
РОЗДІЛ 2 ПРОЕКТУВАННЯ СИМУЛЯТОРА.....	30
2.1 Основні принципи роботи криптобірж.....	30
2.2 Визначення вимог до симулятора.....	32
2.3 Вибір програмних інструментів та технологій.....	36
2.4 Архітектура симулятора та його основних компонентів.....	38
2.5 Опис процесу моделювання криптобіржі.....	39
Висновки до другого розділу.....	41
РОЗДІЛ 3 СТВОРЕННЯ СИМУЛЯТОРА КРИПТОБІРЖІ.....	42
3.1 Реалізація основних функціональностей симулятора.....	42
3.2 Інтеграція симулятора із даними реальних ринків.....	47
3.3 Тестування та налагодження симулятора.....	50
Висновки до третього розділу.....	52
РОЗДІЛ 4 ЕКОНОМІЧНА ЧАСТИНА.....	53
4.1 Організаційно-економічне та маркетингове обґрунтування проекту...	53

					<i>КРБ.КІ.1.146-03.1.2</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Проектування та розробка симулятора роботи криптобіржі	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Розробив</i>		<i>Артемій КУРІЛКО</i>					6	
<i>Перевірив</i>		<i>Ірина КОЛУМБА</i>						
<i>Рецензент</i>		<i>Олександр ЛАТЕНКО</i>						
<i>Нормоконтроль</i>		<i>Світлана ЖУКОВЕЦЬКА</i>						
<i>Затвердив</i>		<i>Сергій АРТЕМЕНКО</i>			<i>гр. КІ-5416, ОНТУ</i>			

4.2 Розрахунки економічної ефективності впровадження нового програмного продукту.....	59
4.3 Визначення капітальних витрат.....	65
Висновки до четвертого розділу.....	69
РОЗДІЛ 5 ОХОРОНА ПРАЦІ.....	70
5.1 Загальні вимоги до особистого робочого місця.....	70
5.2 Загальні вимоги безпеки при роботі з комп'ютером та іншою оргтехнікою.....	73
5.3 Вимоги безпеки перед початком роботи з комп'ютером (ноутбуком) та іншою оргтехнікою.....	74
5.4 Вимоги безпеки під час роботи з комп'ютером та іншою оргтехнікою.....	75
5.5 Вимоги безпеки після закінчення роботи з комп'ютером, принтером, ксероксом, сканером та іншою оргтехнікою.....	76
5.6 Вимоги техніки безпеки та безпеки життєдіяльності в аварійних ситуаціях при роботі з комп'ютером та іншою оргтехнікою.....	77
5.7 Вимоги безпеки в аварійних ситуаціях при роботі з комп'ютером.....	77
Висновки до п'ятого розділу.....	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	79
ДОДАТКИ.....	82

Змн.	Арк.	№ докум.	Підпис	Дата	

ВСТУП

Обмін криптовалют – це платформа, яка дозволяє користувачам купувати та продавати різні монети. Це як фондова біржа, але замість акцій ви купуєте та продаєте криптовалюту. Найпоширеніший тип біржі, основною характеристикою якої є наявність централізованого оператора, що відповідає за безпеку, оновлення та функціональність біржі. Головними перевагами централізованих платформ є швидкість криптовалютних транзакцій і відсутність проблем з ліквідністю. Основний недолік — безпека, оскільки централізовані біржі зберігають кошти користувачів у своїх гаманцях, і, як показує практика, ці гаманці часто піддаються злому. Основний недолік — безпека, оскільки централізовані біржі зберігають кошти користувачів у своїх гаманцях, і, як показує практика, ці гаманці часто піддаються злому.

Концепція криптовалюти народилася у 2008 році, коли анонімна людина чи група людей опублікували під псевдонімом Сатоші Накамото документ під назвою "Біткойн: однорангова електронна грошова система".

У документі описувалася ідея використання технології блокчейн для створення децентралізованої електронної валюти, яка не контролюється центральним органом або організацією. У 2009 році з'явився перший блок блокчейна Bitcoin, започаткувавши епоху криптовалют.

З моменту появи біткойна зявилося безліч криптовалютів, кожна з яких має свої унікальні особливості та цілі. Одні, такі як ethereum, є платформою для розробки та експлуатації децентралізованих додатків (dapps), інші, такі як ripple, призначені для використання як інструмент для міжнародних платежів. Хоча криптовалюти мають безліч переваг, таких як анонімність, безпека та децентралізація, вони також страждають від протидії через потенційні ризики для інвесторів та питання зловживання у незаконній діяльності.

Криптовалютна промисловість ще молода і постійно змінюється. Різні регулюючі органи по всьому світу створюють закони та правила для

контролю та забезпечення безпеки використання криптовалюти. Хоча багато інвесторів, як і раніше, з обережністю ставляться до криптовалютів, є й ті, хто зробив значні інвестиції в цей ринок.

За останні кілька років криптовалютний ринок пережив стрімке зростання і популярність. Криптові біржі стали основним пунктом обміну цифрових активів, даючи змогу трейдерам з усього світу купувати та продавати криптовалюти, відкривати та закривати позиції, аналізувати ринки та виконувати інші операції. Однак робота на криптовалютній біржі вимагає від трейдерів значних знань, досвіду, стратегій та розуміння ринкових умов і факторів, що впливають на ціни активів.

Метою даної дипломної роботи є проектування та розробка симулятора криптовалютної біржі. Завдання - вивчити основні процеси, що відбуваються на криптовалютних біржах, проаналізувати функціональність популярних платформ, визначити вимоги до симулятора і розробити програмне забезпечення, що забезпечує реалістичну симуляцію криптовалютної біржі.

Під час дослідження буде враховано особливості криптовалютного ринку, як-от висока волатильність, швидкість та обсяги торгів.

РОЗДІЛ 1

ЗБІР ТА АНАЛІЗ ІНФОРМАЦІЇ

1.1 Обґрунтування актуальності дослідження

Актуальність дослідження на тему "Проектування та розробка симулятора роботи криптобіржі" обґрунтовується декількома факторами:

— Швидкий розвиток криптовалютного ринку: криптовалютний ринок постійно зростає і змінюється. Останніми роками обсяг торгівлі криптовалютами і кількість людей, які беруть активну участь у цих ринках, значно збільшилися.

— У такому динамічному середовищі важливо мати інструменти, що дають змогу аналізувати та прогнозувати коливання цін, ефективно управляти ризиками та розробляти торгові стратегії. Симулятори криптобірж дають змогу трейдерам та інвесторам випробувати різні стратегії та попрактикуватися без ризику втратити реальні гроші.

— Необхідність практичного досвіду Для ефективної роботи на криптобіржах необхідно набратися досвіду.

Однак процес набуття досвіду на реальній біржі може бути дорогим і ризикованим. Симулятори криптобірж дають змогу трейдерам та інвесторам здобути практичний досвід у віртуальному середовищі, даючи змогу їм експериментувати з різними стратегіями, аналізувати результати та вдосконалювати свої навички без ризику втратити реальні гроші.

Ризикованість криптовалютних інвестицій. Криптовалюти належать до активів з високим рівнем ризику, оскільки їхні ціни можуть значно коливатися протягом коротких періодів часу.

Щоб ефективно управляти ризиками та ухвалювати обґрунтовані рішення, важливо вміти аналізувати ринок і експериментувати з різними торговими стратегіями. Симулятор криптобіржі дає змогу трейдерам та інвесторам перевірити свої ідеї та ризиковані стратегії в безпечному

віртуальному середовищі.

— Інновації та технологічний розвиток: розвиток технології блокчейн і криптовалютного ринку призводить до появи нових можливостей та інструментів для трейдерів та інвесторів.

Розробка симуляторів криптобірж сприяє інноваціям і впровадженню нових технологій у фінансовому секторі. Вони можуть досліджувати й експериментувати з новими торговими стратегіями, аналізувати вплив технологій на ринок і розробляти нові підходи до торгівлі криптовалютою.

Тому розробка симуляторів криптобірж є актуальним питанням, яке допоможе трейдерам та інвесторам здобути практичний досвід, управляти ризиками та розвиватися на криптовалютному ринку, що швидко змінюється.

Під час аналізу криптовалютних бірж можна виокремити два напрямки: торгівля та обмін. Наприклад, ви хочете отримати вигідний курс обміну для своєї криптовалюти, і біржі ідеально підходять для цього. Деякі користувачі навіть вносять гроші на такі ресурси і використовують їх як своєрідний гаманець. Однак досвідчені експерти не рекомендують цього робити, оскільки обмінники, хоч і рідко, але відправляються на смітник. Найголовніша функція біржі - продавати валютні пари і приносити величезний дохід трейдерам.

Щоб отримувати прибуток, не обов'язково бути експертом у сфері криптовалютних бірж. Заробити статки можна і за допомогою спекуляцій: є багато прикладів, коли люди, які спочатку вклали всього кілька сотень фунтів, змогли зібрати свій перший мільйон доларів. Щоб зрозуміти, наскільки ви близькі до великого прибутку, досить подивитися, як швидко змінюються курси на криптовалютних біржах у режимі онлайн.

Традиційний валютний ринок є найбільшим ринком у світі, що перевершує за обсягом щоденних торгів навіть фондовий ринок. Щодня на валютному ринку укладається угод на суму, еквівалентну 5,1 трильйона доларів США, а основними гравцями є комерційні банки і центральні банки, а також інвестиційні фонди, хедж-фонди, приватні інвестори і корпорації.

Валютні ринки - це не просто місце для обміну однієї валюти на іншу або отримання прибутку від арбітражу.

Вони виконують важливі функції, що підтримують стабільність світової економіки, як-от усунення валютного і кредитного ризику, диверсифікація валютних резервів держав і компаній, а також слугують форумом для коригування валютного курсу за допомогою інтервенцій.

Криптовалютні ринки виконують аналогічну функцію, а криптовалюти біржі є найважливішими інфраструктурними гравцями.

Простота входу та участі у торгах на криптовалюти біржах дала змогу роздрібним інвесторам стати найважливішими учасниками ринку. За словами голови криптовалюти біржі Binance, на роздрібних інвесторів припадає близько 60% обсягу торгів на Binance. Щоб розпочати торгівлю, користувачам майже завжди необхідно прискорити процес реєстрації, верифікації та депонування криптовалют та фіатних валют. Це може зайняти від години до кількох днів, залежно від методу поповнення счета - чи то швидко переказ криптовалюти, чи то трудомісткий банківський переказ.

Звідки беруться криптовалюти?

Золото видобувають у шахтах, криптовалюти добувають у майнінгу (буквально, майнінг - це видобуток). Робота учасників полягає у створенні нових блоків у блокчейні. Блоки створюються за допомогою ферми - спеціального потужного пристрою для майнінгу. Нагородою за створення блоку є криптовалюта.

Криптовалюта має ряд переваг:

—Доступність. Віртуальними грошима можна скористатися будь-якої миті;

—Анонімність. Якщо операції зі звичайними електронними грошима ретельно контролюються банком і легко відстежуються, то з такою криптою немає.

—Надійність. Віртуальні монети ретельно захищені, систему неможливо зламати чи обдурити;

—Незалежність. Емісію криптовалют ніхто не регулює. Те саме стосується пересування коштів;

—Обмеженість. Криптомонети, як правило, випускаються в обмеженому обсязі. Це за замовчуванням дефляційна валюта, яка не піддається інфляції.

Крім очевидних переваг, майнінг має і низку недоліків. По-перше, існує ризик державних заборон. Багато органів управління насторожено ставляться до криптовалют і тому обмежують їх використання.

Ще одним недоліком криптовалют є їхня волатильність. Майнінг - це непередбачуваний і ризикований процес. Вартість криптовалют залежить від попиту на них. А попит постійно змінюється через багато факторів, зокрема через зміни в законодавстві. Зрештою, люди ризикують втратити свої кошти, якщо забудуть ключ доступу, який являє собою спеціальний пароль для електронних грошей. Цей ключ неможливо відновити, тому в разі його втрати доступ до грошей на гаманці неможливий.

В даний час криптовалютні біржі отримують прибуток в основному від комісійних за торгівлю криптовалютами та від інституційних інвесторів (інвестиційних фондів та керуючих), які тримають криптовалюти. Чим частіше і більше обсяг торгів, тим вищий прибуток.

Саме тому криптовалютні біржі отримують прибуток від волатильності, тобто зростання та падіння цін на криптовалюти.

Якщо біткоїн за один день виросте на 30%, а потім раптово впаде на 20%, то власники криптовалютних бірж "затаять дихання". Однак вони все одно матимуть вигоду від зростання вартості криптовалют, що привабить нових інвесторів та трейдерів.

За останній рік, з квітня 2020 року до квітня 2021 року, біткоїн, ефіріум та інші криптовалюти досягли рекордних максимумів, що безпосередньо вплинуло на доходи і чистий прибуток криптовалютних бірж 13 травня криптовалютна біржа Coinbase оголосила фінансові результати за перший квартал 2021 року.

Фінансові показники за перший квартал 2021 року. Цифри вражають: чистий прибуток компанії за перший квартал у 24 рази перевищив аналогічний показник за перший квартал 2020 року.

Інвестори зазвичай оцінюють потенціал зареєстрованої на біржі компанії, беручи попередні доходи з поправкою на майбутнє зростання. У разі криптовалютних бірж цього зробити не можна, оскільки бізнес занадто молодий, але можна оцінити потенціал криптовалютного ринку в цілому та його учасників зокрема. Криптовалютний ринок вимагає чіткого розуміння не тільки криптовалют, а й смарт-контрактів, децентралізованих додатків (*dApps*), децентралізованих фінансів (*DeFi*) та небіржових токенів (*NFT*) на центральній вулиці. Криптовалюти та блокчейн – це величезні галузі, які тільки зароджуються та мають потенціал кардинально змінити наше життя у найближчі десятиліття.

У глобальному масштабі майбутнє криптовалютних бірж залежить від майбутнього блокчейну та криптовалют загалом - ринку, який, за оцінками аналітиків, зросте в кілька разів, причому аналітики *Goldman Sachs* особливо відзначили: "Майбутнє криптовалютних бірж залежить від майбутнього блокчейну та криптовалют в цілому".

Вони також прогнозують, що акції криптовалютної біржі *Coinbase* зростуть на 36% протягом наступних 12 місяців. Звичайно, за такий короткий період блокчейн не встиг захопити світ, а *Coinbase* не встигла отримати з цього вигоду. Однак очікується, що комісії за транзакції, які є основним джерелом доходу для криптовалютних бірж, зростатимуть через зростання кількості користувачів, що продовжується.

Насправді кількість активних адрес на блокчейні неухильно зростає. Найголовніше, що біткоїн та інші криптовалюти купують не тільки індивідуальні інвестори, а й так звані інституційні інвестори, такі як інвестиційні фонди та керуючі. Наприклад, у жовтні 2020 року інвестиційна компанія *Stone Ridge* з активами під управлінням у 10 мільярдів доларів оголосила, що купить 10 000 біткоїнів. Найбільш широко відомими

прикладми є покупки криптовалюти компаніями *Tesla*, *Square* та *MicroStrategy*: зокрема, *Tesla* придбала біткоїн на суму \$1,5 млрд.

Найголовніша перевага криптовалютної біржі – простота використання. Транзакції на платформі можуть здійснюватися 24 години на добу з будь-якого пристрою.

Проте є один недолік. Власники криптовалюти не мають повного контролю за коштами, що знаходяться на їхньому особистому балансі.

Суть цього недоліку в тому, що вони не мають контролю над своїми монетами (адже навіть після купівлі криптовалюти закритий ключ, як і раніше, належить її власнику, а не покупцю) і не можуть бути повністю незалежними при торгівлі на біржі (всі дії повинні бути санкціоновані керівництвом біржі). Це накладає деякі обмеження на роботу з цифровими активами, наприклад, наявність верхнього та нижнього порогів виведення монет.

Безпека користувачів на криптовалютних біржах є важливою частиною рівняння - хоча рівень захисту облікових записів може бути підвищений, наприклад, шляхом впровадження *2FA* (двофакторної аутентифікації), самі біржі залишаються вразливими для хакерів. Після неприємних інцидентів (наприклад, платформа *Mt. Gox* була серйозно зламана в 2014 році), біржі почали страхувати свої заощадження для безпеки своїх клієнтів. Проте з того часу це питання залишається невирішеним.

Загалом, щоб набути повної свободи у світі криптовалют, необхідно стати володарем закритого ключа. Другий варіант – почекати ще трохи. Оскільки криптовалюти не є законним платіжним засобом у більшості країн, для цифрових монет немає єдиного регулятора, а отже, немає й безпеки. Проте, враховуючи надшвидкі темпи розвитку віртуальних валют, криптовалютні біржі незабаром зможуть забезпечити безпеку власних коштів клієнтів.

Чи варто інвестувати в криптовалюту?

Ризики, пов'язані з волатильністю криптовалют, дружелюбні.

Ретельно відстежуючи та аналізуючи невеликі цінові рухи, досвідчені трейдери навчилися отримувати вигоду з руху цін, купуючи та продаючи монети у потрібний час доби. Це почуття здобувається з практикою, але також підкріплюється знаннями. Вивчення інструментів торгівлі не тільки допоможе вам залишатися прибутковими, але й збільшить прибуток.

Так, криптовалютний ринок мінливий і непередбачуваний. Деякі експерти не рекомендують використовувати його як довгострокові інвестиції. Проте біткойн існує вже понад десять років, набагато довше, ніж передбачали ці експерти. І, незважаючи на часті злети та падіння, у довгостроковій перспективі очікується стабільне зростання криптовалютного ринку: незважаючи на кризу, спричинену пандемією *COVID-19* у 2020 році, біткойну вдалося встановити новий ціновий рекорд у 2021 році.

1.2 Огляд існуючих кріпторіж та їх обмежень

Існують два суттєво різних типи бірж: централізовані та децентралізовані, розглянемо кожен тип окремо.

— Централізована біржа *CEX*:

Основною характеристикою централізованої біржі і те, що це кошти користувачів зберігаються на самій біржі. Це розкриває переваги та недоліки даного механізму. У широкому сенсі перевагами є надійність, швидкість торгівлі, гарантії біржі та технічна підтримка. Недоліками є той факт, що ваші гроші йдуть у чужу кишеню, повна прозорість операцій для державних та фіскальних органів, підтримка міжнародних санкцій за географічною ознакою (деякі біржі нині не працюють із користувачами з Росії).

Ще одна важлива проблема полягає в тому, що обмін фіатних та криптовалют відбувається майже виключно на централізованих біржах. Децентралізовані біржі, зазвичай, торгують лише криптовалютами.

Централізовані біржі схожі на звичайні банки: якщо ви хочете купити долари у банку, ви спочатку кладете рублі на свій банківський рахунок, а потім конвертуєте їх у долари. Криптовалюти працюють так само: ви вносите

гроші на біржу через банківську картку, *PayPal* або готівкою, а потім купуєте, продаєте або обмінюєте їх на те, що хочете. Гроші залишаються на біржі, доки ви їх не знімете. Ось чому ви не можете торгувати анонімно на централізованій біржі: якщо ви відправите фотографію документа або селфі з документом, вам знадобиться *KYC* (перевірка паспорта). Якщо ви хочете продати тонну кокаїну за один біткоїн, а потім використовувати його для купівлі партії мінометів, викрадених зі складу, то централізована біржа – не найкращий варіант. Однак, якщо ви втратите пароль, ви все одно можете використати свій паспорт, щоб відновити доступ до гаманця.

Для більшості користувачів останнє явно важливіше за перше.

Ще один момент полягає в тому, що централізовані біржі зручні та прості у використанні навіть для новачків.

—Децентралізована біржа *DEX*:

Децентралізована криптовалютна біржа обмінює криптовалюти між користувачами без зберігання коштів у гаманцях. Вона працює за принципом смарт-контрактів і допускає лише угоди типу криптовалюта-криптовалюта, а не фіатні валюти, оскільки спеціальні процедури у блокчейні гарантують угоди та їхню справжність. Крім того, смарт-контракти можуть обмінюватися валютами лише в межах однієї мережі блокчейн. Це означає, що, наприклад, біткоїн і етеріум не можна обміняти безпосередньо. Вони знаходяться на різних блокчейнах.

Транзакції не потребують депозиту. Транзакції здійснюються із підключеного криптовалютного гаманця користувача, тому гроші завжди у вашій кишені. Обмінник не може витратити гроші, але компенсує це в середньому вищою комісією.

Binance

Binance - одна з найбільших та найпопулярніших криптовалютних бірж у світі. Заснована в 2017 році і розташована в Шанхаї, Китай, *Binance* пропонує ряд криптовалют, включаючи *Bitcoin*, *Ether* та *Litecoin*. Біржа також

пропонує низку торгових інструментів, включаючи маржинальну торгівлю та ф'ючерси.

Давайте розглянемо деякі критерії, які використані для оцінки *Binance* як криптовалютної торгової платформи.

Комісійні: *Binance* має одні з найнижчих комісійних зборів на ринку криптовалютних бірж. Комісія за транзакцію становить 0,1%, але може сягати 0,075% при використанні tokenів *Binance (BNB)*.

Ліквідність: біржа має високий рівень ліквідності більшість торгових пар. Це означає, що криптовалют можна швидко і легко купити і продати на *Binance*, що робить її привабливою біржею для торгівлі.
Безпека: *Binance* інвестує значні засоби у свою безпеку та вживає різноманітних таких як двофакторна автентифікація, система попередження про підозрілі активності та захист від *DDoS*-атак. *Binance* також зберігає більшу частину коштів у "холодних" гаманцях, що значно знижує ризик злочину.

Плюси:

—Багатий вибір криптовалют: *Binance* підтримує більше 100 криптовалют, що робить її однією з найбільших та найрізноманітніших криптовалютних торгових платформ.

—Низькі комісії: деякі комісії *Binance* є одними із найнижчих на ринку, особливо при використанні власної криптовалюти *Binance Coin (BNB)*.

—Висока безпека: *Binance* має багатофакторну автентифікацію, зашифроване зберігання засобів та захист від хакерів. Безпека є одним із пріоритетів *Binance*.

—Підтримка *API*: *Binance* підтримує *API*, що дозволяє розробникам створювати додатки та торгові боти, здатні автоматизувати операції на платформі.

—*Binance Launchpad*: *Binance Launchpad* дозволяє користувачам брати участь у публічному продажу tokenів на ранніх стадіях розвитку проектів. Це дає користувачам доступ до перспективних криптовалютних

Змн.	Арк.	№ докум.	Підпис	Дата	

проектів.

В цілому, *Binance* є однією з найпопулярніших і широко використовуваних криптовалютних торгових платформ. Однак, як і будь-яка платформа, вона має свої переваги і недоліки, які необхідно враховувати при виборі торгової платформи.

COINBASE

Coinbase - одна з найпопулярніших криптовалютних бірж у США. *Coinbase* була заснована в 2012 році і базується в Сан-Франциско, пропонуючи широкий спектр криптовалют, включаючи біткоїн та ефір. Біржа також пропонує послуги із зберігання криптовалюти, такі як гаманці.

Плюси:

—Простота використання: *Coinbase* має зручний інтерфейс, який легко використовувати для купівлі, продажу та зберігання криптовалюти. Він дуже зручний для користувачів-початківців.

—Безпека: *Coinbase* має ряд функцій безпеки, включаючи двофакторну автентифікацію, зберігання засобів у холодному гаманці та страхування для захисту активів користувачів від крадіжки.

—Репутація: *Coinbase* - одна з найбільших та найвідоміших платформ для купівлі та продажу криптовалют. Вона має відмінну репутацію та захищає своїх користувачів.

—Доступність: *Coinbase* доступна у багатьох країнах світу та підтримує безліч криптовалют.

В цілому, *Coinbase* – це надійна та зручна платформа для купівлі та продажу криптовалют, але перед її використанням слід уважно вивчити усі плюси та мінуси.

1.3 Мета та завдання проекту

Основною метою проекту є розробка інтерактивного симулятора криптобіржі, який точно відтворює реальні умови та функціонування сучасної криптовалютної біржі. Симулятор має слугувати навчальним посібником для

тих, хто хоче навчитися торгувати криптовалютами та розвинути практичні навички управління ризиками та портфелем.

Цілі проєкту:

—Аналіз і вивчення функціонування реальної криптовалютної біржі: перед розробкою симулятора необхідно провести детальний аналіз, щоб відтворити в симуляторі функціонування реальної криптовалютної біржі.

—Розробка користувацького інтерфейсу: основна мета симулятора - наблизити користувача до реального досвіду торгівлі на криптовалютній біржі. Тому важливо створити інтерфейс, що максимально відображає реальний інтерфейс криптобіржі, з необхідними елементами і функціями.

—Розробка систем опрацювання ордерів і управління активами: симулятор має включати системи опрацювання ордерів (торгові пари) і управління активами (криптовалютні кошти), щоб користувачі могли взаємодіяти з ринком і здійснювати торгові операції.

—Впровадження алгоритмів аналізу даних і прогнозування ринкових тенденцій: щоб підвищити точність симуляції і наблизити її до реального ринкового середовища, симулятор може використовувати алгоритми аналізу даних і прогнозування ринкових тенденцій.

—Розробка системи управління ризиками: управління ризиками - важливий елемент торгівлі на криптобіржах. Симулятори мають бути оснащені системами управління ризиками, які встановлюють обмеження на кількість угод і розмір ордерів і вчать користувачів ефективно управляти своїми портфелями.

—Інтеграція з реальними ринковими даними: для точного моделювання ринкового середовища симулятори мають бути інтегровані з реальними ринковими даними, що дасть змогу користувачам працювати з найсвіжішою інформацією і тенденціями.

—Реалізація функції відстеження та аналізу статистики: симулятор має надавати можливість відстежувати й аналізувати статистику торгових

операцій, таку як прибутковість, ризик та інші показники.

—Тестування та виправлення помилок: після розроблення симулятора необхідно провести тестування та виправлення помилок для забезпечення функціональності та надійності.

—Розробка навчальних матеріалів і посібників.

—Одночасно з тренажером необхідно розробити навчальні матеріали та посібники, які допоможуть людям зрозуміти основні концепції та стратегії криптовалютної торгівлі.

—Документування проєкту та підготовка до публікації або розповсюдження: після завершення розробки проєкт необхідно задокументувати та підготувати до публікації або розповсюдження, щоб велика кількість користувачів могла використовувати цей освітній інструмент.

—Проєктування і розробка симулятора криптобіржі - важливий крок у наданні новачкам можливості навчитися торгувати криптовалютами і розвинути навички управління ризиками. Цей освітній інструмент допомагає користувачам отримати практичний досвід торгівлі криптовалютами, не ризикуючи реальними грошима.

1.4 Вибір програмних засобів для реалізації програмного комплексу

У рамках дипломного проєкту було вирішено розробити автоматизовану систему, яка включає базу даних для зберігання облікових записів користувачів та замовлень. Враховуючи функціональність аналогічних існуючих програм, необхідно було створити сучасний програмний продукт, який уникнув би недоліків існуючих комерційних програм, таких як висока вартість впровадження та обслуговування, слабка підтримка користувачів з різним професійним досвідом. Також необхідно

ідеально підходять для створення бізнес-, мобільних та веб-додатків. *Java* – це об'єктно-орієнтована мова програмування.

Java належить до сімейства мов із *C*-подібним синтаксисом, серед яких найближчими є *C++* та *C#*. У ньому є статична типізація, підтримка поліморфізму, переваження операторів (включаючи явні та неявні оператори перетворення типів), делегати, властивості, події, атрибути, загальні типи та методи, ітератори, потоки, багатопоточність, збирачі сміття та винятки.

Всі мови програмування мають свої переваги та недоліки. Враховуючи їх, фахівці обирають потрібну мову для конкретного завдання; *Java* більше не є єдиним вибором для *Android*, як і стандартний вибір для веб-програмування, але він не втратив свого місця.

Іншим основним застосуванням мови є розробка серверних програм для підприємств, таких як фінансове програмне забезпечення. Великі компанії вважають за краще використовувати програмне забезпечення *Java* через його високу надійність і безпеку. Не менш поширене створення настільних та веб-додатків (наприклад, платформ електронної комерції, освітніх платформ, порталів новин) та веб-серверів на *Java*. Віртуальні машини існують для того, щоб код *Java* міг виконуватися не тільки на комп'ютерах, а й на багатьох інших типах пристроїв. Програмне забезпечення цією мовою може використовуватися в багатьох вбудованих системах, таких як банкомати, пристрої для платіжних терміналів, побутова електроніка та різні промислові та наукові пристрої. Інші поширені області застосування мови включають програмне забезпечення для роботи з великими даними та *Java*-ігри, такі як *Minecraft*.

Програми *Java* зазвичай перетворюються на спеціальний байткод, щоб вони могли працювати на будь-якій архітектурі комп'ютера, що реалізує віртуальну машину *Java*. Основні переваги це:

— Програмування *Java* відповідає стандартним вимогам корпоративних обчислювальних систем. Завдяки різним бібліотекам

розробники можуть реалізувати різні функції, необхідні організації.

— Об'єктно-орієнтована платформа дозволяє визначати типи даних, їх структурні характеристики та функції, що до них застосовуються. Управління структурами даних дозволяє встановлювати відносини між об'єктами; ООП дозволяє повторно використовувати об'єкти в інших додатках, не розкривати дані, до яких не слід звертатися, щоб запобігти помилкам, та полегшує оновлення старого коду за рахунок інтелектуально структурованих рішень.

— Для *Java* характерно дозвіл потенційних уразливостей. Адміністратори безпеки за допомогою політики безпеки можуть встановити правила доступу до бази даних, які дозволяють уникнути вразливості.

— Незалежність від платформи дозволяє написати код один раз і застосовувати його будь-де; *Java* має крос-платформні можливості, дозволяючи компілювати байткод для запуску на інших платформах, де працює *JVM*.

— Підтримка розподіленого програмування дозволяє співробітникам працювати віддалено.

— Функції автоматичного керування пам'яттю допомагають ефективно використовувати центральний процесор без зниження продуктивності, спричиненого програмами очищення пам'яті.

— Також підтримується багатопоточність. Коли потоки виконуються одночасно, вони використовують одні й самі області пам'яті. Це особливо важливо для анімаційних програм, оскільки дозволяє швидко перемикатися між потоками, які не залежать один від одного.

Переваги та недоліки сайтів на *Java* обговорюються вже давно, але статистика використання серверних фреймворків та мови показує, що ця платформа входить до п'ятірки найкращих.

На цій платформі часто будуються великі корпоративні портали та сайти великого обсягу.

Якщо ви користуєтесь більшістю з них, ви ніколи не зрозумієте, що

вони побудовані на *Java*.

Java була обрана через її високу надійність, стабільність і суворі стандарти і специфікації. Нижче наведено деякі з найбільш популярних веб-сайтів на базі *Java*:

Java є однією з найчастіше використовуваних платформ протягом багатьох років і навряд чи втратить свій статус у найближчому майбутньому. Ця проста у використанні платформа підходить і для новачків: понад 9 мільйонів людей використовують її на 3 мільярдах пристроїв. Незважаючи на його незначні недоліки, пройде зовсім небагато часу, перш ніж буде розроблена мова програмування, аналогічна *Java*, і займе його місце завдяки вищезгаданим перевагам - високій надійності, гнучкості та автоматичному управлінню пам'яттю.

Java - це мова високого рівня, тобто. він схожий на людську мову. Він відрізняється від мов нижчого рівня, які схожі на машинний код. Мови високого рівня використовують компілятори та інтерпретатори для виконання перетворень.

Це спрощує розробку, роблячи мову легшою для написання, читання та підтримки.

Java схожий на *C* тим, що його синтаксис заснований на *C++*. Однак синтаксис *Java* простіше, що дозволяє новачкам швидше освоїти мову та ефективніше використовувати код для досягнення конкретних результатів.

Java не така зручна для новачків, як *Python*, але для розробників, які розуміють основи фреймворків, пакетів, класів і об'єктів, вона досить проста. Вона проста, типізована та передбачувана і може допомогти вам навчитися мислити у правильному напрямку. Крім того, новачки завжди можуть звернутися до численних безкоштовних онлайн-підручників та курсів.

Корпоративні програми були основною перевагою *Java* з 1990-х років, коли організації почали шукати потужні інструменти програмування, крім *C*. *Java* підтримує безліч бібліотек, які є будівельними блоками корпоративних систем. Бібліотеки допомагають розробникам створювати будь-які

Змн.	Арк.	№ докум.	Підпис	Дата

функціональні можливості, потрібні підприємству. *Java* - широко популярна мова, яка викладається в рамках введення в програмування у більшості шкіл та університетів. Більшість хостинг-провайдерів підтримують *Java*. Крім того, *Java* - це мова, яка не вимагає особливого догляду. Ви можете використовувати *Java* на будь-якому комп'ютері, незалежно від апаратної інфраструктури.

Існує думка, що *Java* – безпечна мова, але це не зовсім так. Хоча сама мова не запобігає уразливості, є кілька особливостей, які дозволяють усунути поширені вразливості. По-перше, на відміну від мови *C*, *Java* немає покажчиків. Вказівники - це об'єкти, які зберігають адреси осередків пам'яті з різними значеннями, що може призвести до несанкціонованого доступу до пам'яті.

По-друге, *Java* має менеджера безпеки, який є політикою безпеки, створюваною для кожної програми, в якій можуть бути вказані правила доступу. Це дозволяє *Java*-програмам працювати в "пісочниці", що усуває вразливості.

Write once, use everywhere" (WORA) - термін, який зазвичай використовується компанією *Sun Microsystems* в ІТ-індустрії для опису кросплатформенності *Java*: ви створюєте *Java*-додаток на *Windows*, компілюєте його в байткод і запускаєте на інших платформах, що підтримують віртуальну машину *Java (JVM)*) Таким чином, *JVM* виступає як рівень абстракції між кодом і апаратним забезпеченням.

Усі основні операційні системи, такі як *Windows*, *Mac OS* та *Linux*, підтримують *JVM*. Будь-яка програма, яка не залежить від функціональності або інтерфейсів користувача, специфічних для конкретної платформи, може бути легко перенесена, принаймні, в більшості випадків.

Java розроблено як мову для розподіленого програмування. У ньому є вбудовані механізми для спільного використання даних та програм на кількох комп'ютерах, що підвищує продуктивність та ефективність.

У той час як інші мови вимагають використання зовнішніх *API* для розподіленого програмування, *Java* ця технологія вбудована. Унікальний підхід *Java* до розподілених обчислень відомий як *Remote Method Invocation (RMI)*. *RMI* забезпечує безпеку, незалежність від платформи та об'єктно-орієнтоване програмування для розподілених обчислень. Крім того, *Java* підтримує програмування сокетів та метод розподілу *CORBA* для обміну об'єктами між програмами, написаними різними мовами.

Завдяки автоматичному управлінню пам'яттю (*АММ*) розробникам *Java* не потрібно писати код для керування пам'яттю вручну; *АММ* також використовується в мові програмування *Swift* та в програмах для очищення пам'яті, які автоматично обробляють виділення та деалокацію пам'яті. Що це означає?

Ефективність програми безпосередньо з пам'яттю. Однак обсяг пам'яті обмежений.

При написанні програми на мові, яка управляє пам'яттю вручну, існує ризик, що розробник може забути виділити пам'ять, що призведе до великого обсягу пам'яті для програми та проблем з продуктивністю. Очищувачі пам'яті шукають та видаляють об'єкти, які більше не використовуються програмою. Це впливає на продуктивність процесора, але розумна оптимізація та налаштування можуть зменшити цей вплив.

Потоки - це найменша одиниця обробки програмування: щоб найефективніше використовувати час процесора, *Java* дозволяє потокам працювати одночасно, що як багатопоточність.

Потоки використовують ту саму область пам'яті, і між ними можна швидко перемикається. Потоки є незалежними один від одного, і один потік не впливає на інші. Особливо це корисно для ігор або програм, в яких використовується багато анімації. Протягом багатьох років розвиток *Java* підживлювався спільнотою, підтримкою *Oracle* та багатством додатків та мов на *JVM*. Крім того, постійно випускаються нові версії *Java* із новими захоплюючими можливостями.

Спільнота розробників *Java* не має собі рівних: близько 45% респондентів в опитуванні *StackOverflow* 2018 використовують *Java*; *Java* має дуже велику екосистему добре перевірених бібліотек та фреймворків для всіх завдань *Java* має дуже велику екосистему добре перевірених бібліотек та фреймворків для всіх завдань. задач. Розробники-початківці віддають перевагу *Java*: на *Udemy* більше 1 000 курсів з програмування на *Java*, а на *Coursera* - понад 300.

Більшість організацій використовують *Java* у тій чи іншій формі: через широкий спектр застосування *Java* майже непомітна. В результаті люди часто запитують: "Де використовується *Java*? Давайте розглянемо, де використовується *Java*:"

Android-додатки: незважаючи на активний розвиток *Kotlin*, *Java*, як і раніше, залишається де-факто основною мовою для *Android*-додатків. Тому кожному *Java*-розробнику дуже легко стати програмістом *Android*: хоча *Android* використовує *Android SDK*, а не *JDK*, код, як і раніше, пишеться на *Java*.

Програмні продукти Окрім вже згаданих *Hadoop* та *Apache Storm*, *Java* використовується для створення *Eclipse*, *OpenOffice*, *Gmail*, *Atlassian* і т.д.

Фінансове програмне забезпечення. *Java* - одна з найбільш затребуваних мов у фінансовій індустрії. Він використовується для створення потужних, швидких та простих веб-сайтів як на стороні сервера, так і на стороні клієнта. *Java* також використовується для моделювання даних.

Термінали точок продажу (*POS*); багато компаній використовують *Java*, тому що створення *POS*-систем часто потребує крос-платформенності та великої кількості людей.

Транзакційні системи. *Murex*, популярне програмне забезпечення для управління зовнішніми та внутрішніми банківськими операціями, написано на *Java*.

Рішення до роботи з великими даними; *Hadoop* написаний *Java*; *Scala*, *Kafka* та *Spark* використовують *JVM*. Крім того, *Java* має доступ до багатьох

складних бібліотек, інструментів налагодження та моніторингу.

За допомогою віртуальних машин код *Java* може виконуватися не тільки на комп'ютерах, але і на різних типах пристроїв. Програмне забезпечення цією мовою може використовуватися в багатьох вбудованих системах, таких як банкомати, пристрої для продажу товарів, побутова електроніка та різні промислові/наукові пристрої. Інші поширені області застосування мови включають великі дані та *Java*-ігри, такі як *Minecraft*.

Висновок до першого розділу

У цьому розділі була проведена вступна частина дипломної роботи на тему "Проектування та розробка симулятора роботи криптобіржі". Були обгрунтовані актуальність теми, її значення та цілі дослідження. З'ясовано, що криптовалютні біржі є важливими інструментами для торгівлі криптовалютами, а симулятори криптовалютних бірж можуть допомогти трейдерам отримати досвід без ризику втрати реальних коштів.

Змн.	Арк.	№ докум.	Підпис	Дата

РОЗДІЛ 2

ПРОЕКТУВАННЯ СИМУЛЯТОРА

Біржа криптовалют – це платформа для торгівлі та обміну цифровими монетами. Вони дозволяють обмінювати одну криптовалюту на іншу або фіатну валюту, купувати та продавати цифрові активи, спекулювати ними для отримання максимального прибутку. По суті, криптовалютна біржа схожа на традиційну фондову біржу, тільки тут замість акцій і ордерів угоди відбуваються менш цінними цифровими валютами.

2.1 Основні принципи роботи крипто бірж

Існує багато типів криптовалютних бірж, які пропонують різні варіанти монет, з більшими та меншими обсягами торгів, різними функціями платформи та різними заходами безпеки. Щоб максимально розподілити свої транзакції та знизити ризик крадіжки облікового запису, варто тримати монети на двох-трьох різних біржах одночасно. Існує також багато різних моделей обміну криптовалютою. Одна відмінність полягає в тому, чи можна використовувати фіатну валюту для покупки криптовалют, якими ви торгуєте на біржі, і чи можете ви конвертувати свої криптовалюти назад у фіатну валюту після закриття своєї позиції.

Зазвичай найбільш відомі та великі біржі централізовані. Серед них *Coinbase*, найбільша з усіх криптовалютних бірж за кількістю користувачів; *Binance*, яка торгує найбільше, і *Gemini*, ще один великий конкурент у цьому просторі. Централізовані біржі особливо популярні серед нових інвесторів у криптовалюту, оскільки вони, як правило, більш інтуїтивно зрозумілі та часто пропонують спеціальну підтримку 24/7 через різні соціальні канали та чат.

Іншим фактором, який робить централізовані біржі придатними для початківців, є те, що вони несуть відповідальність за зберігання ваших особистих ключів. Як правило, це набагато безпечніше, ніж зберігати дані на вашому особистому пристрої, оскільки біржі витрачають мільйони доларів на

кібербезпеку, щоб забезпечити цілісність і безпеку вашої особистої інформації. Забуття початкової фрази або неправильне збереження адреси може означати безповоротну втрату всіх заощаджень, тому багато новачків знаходять душевний спокій, перекладаючи відповідальність за зберігання даних на великі централізовані гаманці з авторитетом.

Для звичайного трейдера на біржі криптовалют все просто: він заходить на офіційний сайт біржі і відразу бачить таблицю курсів і інтерфейс для створення замовлення. У цьому контексті мало хто внутрішньо замислюється про те, як влаштована біржа.

На думку експертів проекту *Cryptonisation*, користувальницький інтерфейс – лише верхівка айсберга. Для безперебійної роботи біржі необхідний ряд компонентів:

—Торгове ядро забезпечує формування балансу, веде базу даних замовлень, пропонує можливість купівлі та продажу. Це найважливіший фактор у визначенні швидкості ресурсу.

—Основний і резервний гаманці для зберігання активів, забезпечених обміном. Як правило, біржі використовують гарячі гаманці для отримання внесків від користувачів біржі та надійно захищені холодні платформи для зберігання резервів.

—Додаткові модулі для аналітики, аналізу біржових показників та ін.

—Веб-інтерфейс, який відображає торговий термінал і дозволяє користувачеві створити обліковий запис.

—Система управління транзакціями та комісіями, включаючи

платіжний шлюз.

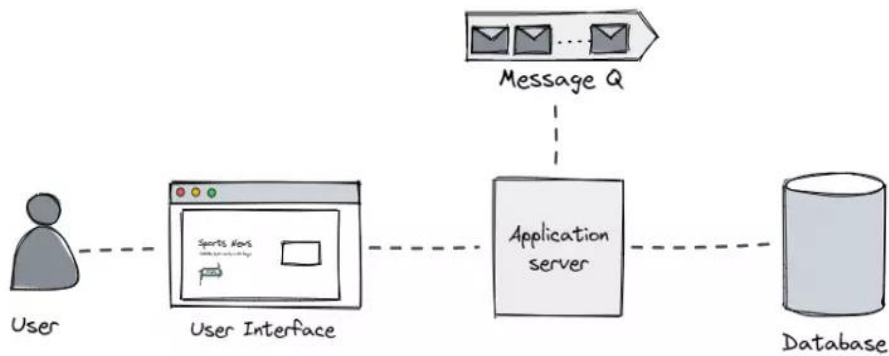


Рисунок 2.1 - Реалізація проекту

На рисунку зображено сторону клієнта до якої надходять запити, запити потім отримуються сервером в якості повідомлень, це може бути повідомлення на створення ордеру чи отримання балансу. Ці повідомлення взаємодіють з внутрішньою базою даних де ми через *sql* синтаксис можемо отримати необхідні данні.

2.2 Визначення вимог до симулятора

Вимоги до симулятора криптобіржі можуть варіюватися в залежності від конкретних потреб та функціональності, яку ви хочете отримати. Ось деякі загальні вимоги, які можуть виникнути:

1. Графічний інтерфейс: Симулятор криптобіржі повинен мати зручний та інтуїтивно зрозумілий графічний інтерфейс, що дозволить користувачам легко взаємодіяти з програмою.

2. Реалістичні дані ринку: Симулятор повинен мати можливість отримувати актуальні дані про криптовалютні курси, обсяги торгів та інші ринкові параметри для відтворення реальних умов.

3. Можливість торгувати різними криптовалютами: Симулятор повинен підтримувати широкий спектр криптовалют, щоб користувачі могли торгувати різними активами.

4. Різні типи ордерів: Симулятор повинен дозволяти встановлювати різні типи ордерів, такі як лімітні, ринкові, стоп-лосс, стоп-ліміт тощо.

5. Історія торгів: Симулятор повинен зберігати історію торгів користувача, щоб він міг аналізувати свої результати та вдосконалювати свої стратегії.

6. Аналітика та графіки: Симулятор може надавати графіки та аналітичні інструменти для вивчення ринку та здійснення технічного аналізу.

7. Торгові інструменти: Симулятор може надавати різні інструменти для управління ризиками, включаючи стоп-лосс, тейк-профіт, підтримку маржевого торгівлі тощо.

8. Різні режими: Симулятор може мати різні режими, такі як режим симуляції реальних ринкових умов, навчальний режим для новачків або режим тестування стратегій.

Ці вимоги можуть бути розширені або змінені в залежності від конкретних потреб користувача або бізнес-вимог. Рекомендується провести детальний аналіз вимог та розробити специфікацію перед початком розробки симулятора криптобіржі.

Звичайно, список критеріїв для створення криптовалютної біржі може змінюватись від людини до людини, але безпека – це перший фундамент.

На біржі кожен крок має бути безпечним, починаючи з моменту внесення фіатної валюти в гаманець і до моменту початку торгів.

На перший погляд, важко оцінити рівень безпеки криптовалютної біржі. Однак необхідно перевірити, які протоколи та стандарти використовує криптовалютна платформа, як і які персональні дані вона зберігає та наскільки прозоро вона працює.

Наприклад, *Coinbase* - це американська біржа, заснована у 2012 році і має ліцензію на роботу у кількох штатах США. Вона також стала першою криптовалютною біржею, яка вийшла на фондовий ринок.

Binance має міжнародно визнаний сертифікат безпеки *ISO/IEC 27001*. Його було отримано після двох зовнішніх аудитів систем інформаційної безпеки.

Ідеальна біржа прозора для своїх користувачів у плані умов, структури

зборів та прозорості торгівлі. Вона також регулярно оприлюднює звіти про свою діяльність.

Чим більше варіантів введення та виведення коштів має гаманець, тим простіше з ним працювати і, по-друге, це показник надійності схеми.

Також важливо, щоб кожен платіжний канал був безпечним. Зверніть увагу на те, як кожен із них забезпечує безпеку транзакцій.

Запитайте про комісії, що стягуються біржею. По суті це основне джерело доходу для криптовалютних бірж. Наприклад, *Coinbase*, друга за величиною криптовалютна біржа у світі, вважається однією з найкращих платформ завдяки простому процесу інвестування криптовалюти, але вона стягує величезну комісію у розмірі 1,49% за транзакцію.

У світі криптовалют є не лише Біткойн. Початківцям рекомендується вибирати біржу, яка має широкий вибір альтернативних монет та торгових пар. Це пов'язано з тим, що більше пар, тим більше можливостей для торгівлі та інвестицій.

Найбільш популярні пари - *BTC/ETH*, *BTC/LTC*, *USDT/BTC* та *ETH/LTC*, але популярність конкретних пар варіюється від біржі до біржі.

Однак деякі біржі включають до списку проблемні монети. У таких випадках трейдери, особливо початківці, ризикують інвестувати в токени, випущені шахраями. Існування таких криптовалютних активів – питання репутації та ставлення до клієнтів.

Криптовалютні біржі також повинні продемонструвати, що вони постійно розглядають лістинг криптовалют, що з'являються, і прислухаються до думки своїх користувачів. Також має значення пропонований сервіс.

Хороші криптовалютні біржі зазвичай пропонують низку продуктів, які можна використовувати залежно від схильності до ризику та обраної інвестиційної стратегії. До таких продуктів відносяться ставки, кредитування, ф'ючерси та маржинальна торгівля.

Більш розвинуті глобальні біржі також запровадили нову форму фінансових послуг під назвою *DeFi* (скорочення від "децентралізованих фінансів").

DeFi - це екосистема, де учасники взаємодіють один з одним безпосередньо, без посередників, таких як банки чи кредитори, що робить транзакції швидше та дешевше, і заснована на протоколі *DeFi*, який значною мірою базується на блокчейні *ethereum*.

Оглянемо вимоги для обраної симуляції криптобіржі, вона повинна мати змогу:

- створювати / відмінити ордер
- отримати інформацію про ордери, активні чи ті які вже виконані
- відображати баланс користувача, криптовалюту та її баланс відкритий та зарезервованій (якщо частина сумми використовується в ордері)
- виконання оредру. Сервіс повинен отримувати реальні данні з криптобіржі та порівнювати з нашими уявними оредрами. Якщо ціна підходить ордер повинен виконуватись частково або повністю в залежності від кількості.

Початківцям слід спочатку визначитися зі стратегією – інвестувати чи торгувати. Виділіть суму, яку ви можете дозволити собі втратити. Потім купіть криптовалюту та переведіть їх на біржу або холодний гаманець. Ви також можете тренуватися на передтренувальному рахунку, читати літературу з певних тем або відвідувати курси.

2.3 Вибір програмних інструментів та технологій

Стандарти кодування – це набір інструкцій і найкращих практик, які використовуються для створення узгодженого високоякісного коду. Стандарти кодування розглядаються, як правила, методи та найкращі практики для розробки чистішого, читабельнішого та ефективнішого коду з мінімальною кількістю помилок. Вони пропонують єдиний формат для

розробників програмного забезпечення для створення складного та високофункціонального коду.

KISS - keep it simple stupid

Відповідно до принципу «будь простим, дурним» ключем до створення успішного продукту є простота. Метод *KISS* застосовується до проектування та розробки цифрових продуктів, але він також широко використовується в інших сферах, таких як менеджмент чи інженерія.

KISS також використовується як аббревіатура для слова *keep it short and simple, keep it short and sweet i keep it simple and straightforward*. Однак усі ці варіації стосуються одного підходу.

Переваги *KISS*:

Швидше у розробці та використанні.

Простіша структура програмного забезпечення робить тестування, включно з автоматизованим, простішим і ефективнішим.

Зменшена складність кодової бази спрощує та пришвидшує обслуговування та адаптацію нових членів команди в середині проекту.

DRY- don't repeat yourself

DRY розшифровується як «не повторюйся» та рекомендує скоротити повторення шаблонів програмного забезпечення. Усуваючи надмірності в процесі та логіці, інженери зменшують технічну заборгованість і покращують ремонтпридатність коду – обидва ці фактори є важливими факторами витрат, особливо в довгостроковій перспективі.

Важливо, що *DRY* слід застосовувати в усій системі, включаючи не лише кодову базу, але й тестування та документацію.

Принцип *DRY* стверджує, що кожна частина знань повинна мати єдине, однозначне, авторитетне представлення в системі (цитата взята з книги «Прагматичний програміст», написаної Ендрю Хантом і Девідом Томасом). Це вимагає від програмістів використання абстракції для видалення характеристик, щоб мати можливість повторно використовувати елемент у різних частинах програми.

Принцип *DRY* гарантує, що будь-яка модифікація одного елемента системи не вимагає зміни інших, логічно не пов'язаних елементів. Тож це ефективний спосіб оптимізації процесу розробки. Крім того, елементи, які логічно пов'язані, змінюються передбачувано та рівномірно, що забезпечує їх синхронізацію.

YAGNI

Основна мета принципу *YAGNI* — уникнути витрачання часу та грошей на переробку речей, які, на вашу думку, знадобляться вам пізніше. Бо врешті-решт зазвичай виявляється, що вам це не потрібно, або те, що вам потрібно, відрізняється від того, на що ви очікували.

Основною перевагою використання *YAGNI* є більш економічно оптимізований процес розробки продукту завдяки:

Краща продуктивність розробників: команда зосереджена на ефективному виконанні поточних вимог. Вони не витрачають час і сили на здогади.

Гнучкіша кодова база: вам не потрібно шукати способи використання неоптимальних рішень, які ви вже розробили, перш ніж матимете повну картину.

Усі ці підходи будуть активно використовуватись в проекті для зменшення часу розробки, збільшення якості програмного продукту та мати більшу масштабованість.

2.4 Архітектура симулятора та його основних компонентів

Це завдання дозволить вам зосередитись на архітектурі, а не на написанні великої кількості бізнес-функцій. Результатом ваших початкових зусиль має стати логічний, добре продуманий код, який дозволить вам відразу ж розпочати реалізацію бізнес-логіки.

Визначте терміни використання інструментів та бібліотек. Я дотримуюсь принципів розробки "під ключ" та *KISS*. Це означає, що якщо ви будете використовувати лише бібліотеки з відкритим вихідним кодом і

впроваджувати їх самостійно, вам знадобиться багато часу, у тому числі час на навчання майбутніх колег та розробників.

—*backend: Java 8;*

—система управління версіями: *Git + Github;*

—установник пакетів: *Yarn;*

—контроль якості кода: *ESLint + Prettier + Stylelint;*

—view: *Postman;*

—store: *MobX;*

—автотести: *Cypress.io* (комплексне рішення на *javascript* замість модульної сборки наприклад

—*Mocha/Karma+Chai+Sinon+Selenium+Webdriver/Protractor*);

—стилі: *SCSS* через *PostCSS* (гнучкість налаштування, працює з *Stylelint*);

—регістрація помилок: *Sentry;*

—утіліти: *Lodash* (економія часу);

—роутінг: під ключ;

—локалізація: під ключ;

—робота с запитом: під ключ;

—метрики швидкодії: під ключ;

2.5 Опис процесу моделювання криптобіржі

Binance використовується для надання ринкових даних клієнтам, і біржа використовує підхід та модель, аналогічні до багатьох інших бірж. Для вимог до логіки торгівлі з низькою затримкою перше, що необхідно підтримувати локально, - це часткова книга ордерів (ціни та обсяги торгів, упорядковані близько до ринкової ціни).

Насправді лише перші кілька позицій впливають на торгові рішення. Інші хвости навряд чи торгуватимуться, і немає особливої необхідності витратити ресурси на аналіз. Біржові оновлення надаються через мережеве сокетне

з'єднання, зазвичай, по кількох каналах з різними характеристиками затримки.

Класичний підхід до аналізу *XML* або *JSON* полягає в делегуванні бібліотеки, яка повертає структуровану виставу, таку як *JSONObject*, *XMLObject*. Це уявлення зазвичай є деревом об'єктів, які видаляються після завершення розбору. Постійне створення тимчасових об'єктів може створювати додаткове навантаження на збирач сміття *JVM*, викликаючи непередбачувані стрибки у затримці.

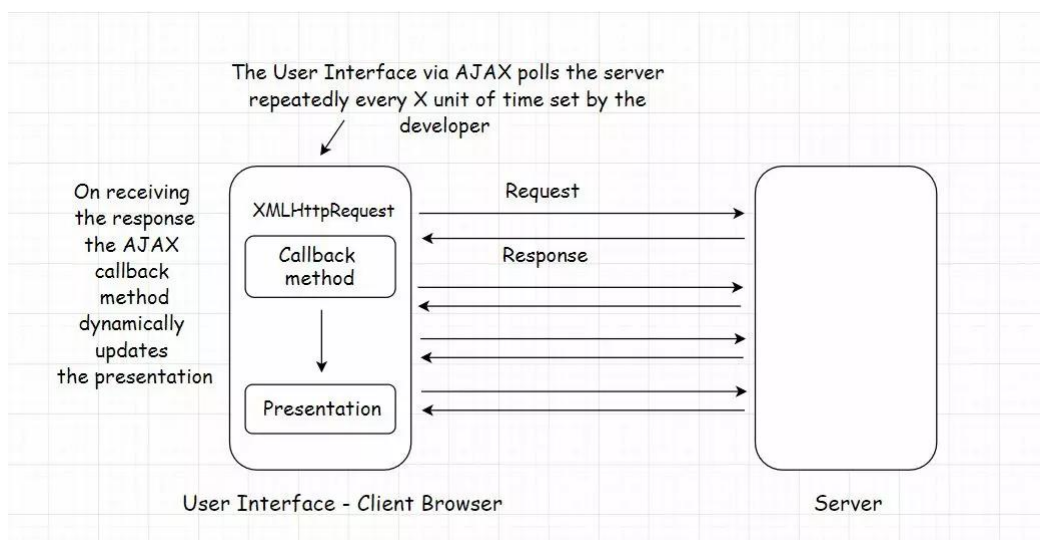


Рисунок 2.2 - Робота з XML на біржі

Ми отримуємо запит в форматі xml який має пари тега та його значення наприклад: "`<name>Denis</name>`". Отримуючи запит від клієнта в якому є вся необхідна інформація, сервіс знає яку операцію потрібно виконати та як.

Для обробки з'єднань через веб-сокети рекомендується використовувати бібліотеку *HttpClient* у *Java 11*. Ця бібліотека використовує неблокуючий введення-виведення та *TCP_NODELAY* за замовчуванням. Крім того, з'єднання через веб-сокети можуть оброблятися через спеціальний *ExecutorService*. Це добре поєднується з моделлю циклу подій, оскільки для виконання операцій, що не блокують, над усіма обробниками каналів ринкових даних потрібен лише один потік.

Наведена нижче діаграма лише підкреслює ідею відокремлення служб

конектора від торгових служб. Ви можете розпочати проект таким чином і при необхідності приєднувати до потоку даних більше ефемерних мікросервісів.

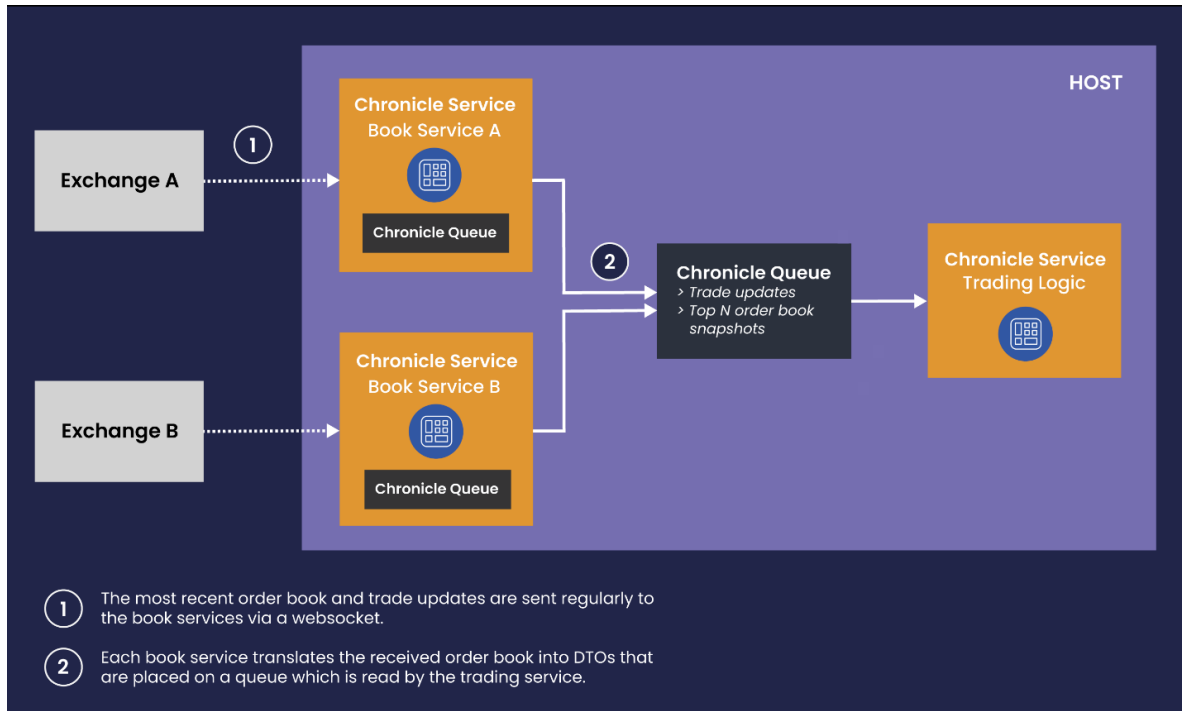


Рисунок 2.3 - Ідея відокремлення служб конектора від торгових служб

Висновок до другого розділу

У другому розділі проведений аналіз сучасного стану криптовалютних бірж. Були розглянуті основні принципи роботи криптовалютних бірж, а також виявлені переваги та недоліки існуючих платформ. Встановлено, що існуючі біржі мають обмежені можливості для навчання новачків та тестування стратегій трейдерів.

РОЗДІЛ 3

СТВОРЕННЯ СИМУЛЯТОРА КРИПТОБІРЖІ

3.1 Реалізація основних функціональностей симулятора

У даному розділі буде описано реалізацію основних функціональностей симулятора, які були визначені у попередньому розділі. Наведено деталі щодо процесу розробки та реалізації кожної функції, включаючи алгоритми, структури даних, використані технології та інструменти.

Функціональність 1: Моделювання фізичних процесів.

Для реалізації функціональності моделювання фізичних процесів було використано принципи фізики та математики для опису руху об'єктів та взаємодії зовнішніх сил. Залежно від потреби, були використані чисельні методи розв'язування диференціальних рівнянь, такі як методи Ейлера або Рунге-Кутта. Для забезпечення точності та швидкості обчислень використовувалися ефективні алгоритми та оптимізації.

Функціональність 2: Візуалізація симуляцій.

Для візуалізації симуляційного процесу було використано графічні бібліотеки або фреймворки, що дозволяють створювати графічні об'єкти, анімацію та інтерфейс користувача. Зазвичай використовувалися такі технології, як *OpenGL*, *WebGL* або інші графічні *API*. Симуляційні дані передавалися до цих бібліотек, які відповідали за відображення їх на екрані з використанням графічних об'єктів та анімації.

Функціональність 3: Управління симуляцією.

Управління симуляцією передбачає можливість контролювати процес симуляції, змінювати параметри об'єктів, встановлювати початкові умови та виконувати інші дії. Для реалізації цієї функціональності були створені інтерфейси користувача, такі як кнопки, ползунки або текстові поля, які дозволяють користувачу взаємодіяти з симулятором та змінювати його стан.

Функціональність 4: Збереження та завантаження симуляцій.

Для зручності використання та обміну симуляційними даними було реалізовано можливість збереження та завантаження симуляцій. Ця функціональність дозволяє користувачам зберігати результати своїх симуляцій на диск та пізніше завантажувати їх для подальшого аналізу або використання. Зазвичай для цього використовуються формати файлів, які зберігають дані у структурованому вигляді, наприклад, *JSON* або *XML*.

Функціональність 5: Взаємодія з користувачем.

Симулятор також має можливість взаємодіяти з користувачем через інтерфейс користувача. Це включає в себе показ повідомлень, сповіщень або відображення додаткової інформації про симуляцію. Користувач може вводити дані, налаштовувати параметри симуляції та отримувати результати через інтерфейс. Для реалізації цієї функціональності використовуються різноманітні елементи інтерфейсу користувача, такі як кнопки, поля вводу або вікна повідомлень.

У даному розділі було розглянуто реалізацію основних функціональностей симулятора. Кожна функціональність була детально описана з урахуванням використаних алгоритмів, структур даних та технологій. Реалізація цих функціональностей забезпечує потужний інструмент для моделювання фізичних процесів та дозволяє користувачам насолоджуватись взаємодією з симулятором.

Клас сервіс для роботи з даними користувача, його баланс, ордери та інше. Остаточний вигляд цих функцій матиме наступний вигляд:

```
public interface UserService {  
    /**  
     * Creates user with specified data.  
     * @param name user's name  
     * @param btcBalance user's btc balance  
     * @param usdtBalance user's usdt balance  
     */  
    void createUser(String name, float btcBalance, float usdtBalance);  
}
```

```

/**
 * Returns all user information.
 * @param userId user id
 * @return User instance
 * @throws UserNotFoundException if {@code userId} not found
 */
User findById(Long userId) throws UserNotFoundException;

/**
 * Returns user's free btc balance.
 * @param userId user's id
 * @return float value
 * @throws UserNotFoundException if {@code userId} not found
 */
float getFreeBTCBalance(Long userId) throws UserNotFoundException;

/**
 * Returns user's reserved btc balance.
 * @param userId user's id
 * @return float value
 * @throws UserNotFoundException if {@code userId} not found
 */
float getReservedBTCBalance(Long userId) throws
UserNotFoundException;

/**
 * Returns user's free usdt balance.
 * @return float value
 * @throws UserNotFoundException if {@code userId} not found
 */
float getFreeUSDTBalance(Long userId) throws UserNotFoundException;

/**
 * Returns user's reserved usdt balance.

```

```

    * @return float value
    * @throws UserNotFoundException if {@code userId} not found
    */
    float    getReservedUSDTBalance(Long    userId)    throws
UserNotFoundException;

    void    transferCurrency(Long    userId,    String    symbol,    boolean
transferToFree,    float    quantity)    throws    SymbolNotFoundException,
UserNotFoundException;

```

А для роботи з ордерами буде використовуватись *OrderService* для створення, редагування та відміни ордерів користувача:

```

public interface OrderService {

    /**
     * Is used to place buy/sell order.
     * @param isBuy boolean
     * @throws UserNotFoundException if user not found by given {@code
userId}
     */
    void placeOrder(Long userId, boolean isBuy, String side, float price, float
quantity) throws UserNotFoundException;

    /**
     * Is used to cancel user's active order by given id.
     * @param userId user id
     * @param orderId order id
     * @param isExecuted if order is executed
     * @throws OrderNotFoundException if order with {@code orderId} not
found
     * @throws UserNotFoundException if user not found by given {@code
userId}
     */
    void cancelOrder(Long userId, Long orderId, boolean isExecuted) throws

```

```

OrderNotFoundException, UserNotFoundException
/**
 * Executes active order and adds it to orders history.
 *
 * @param userId user id
 * @param orderId order id
 * @throws UserNotFoundException if user not found by given {@code
userId}
 * @throws OrderNotFoundException if order with {@code orderId} not
found
 */
void executeActiveOrder(Long userId, Long orderId) throws
UserNotFoundException, OrderNotFoundException;
/**
 * Executes active order partially.
 *
 * @param userId user id
 * @param orderId order id
 * @throws OrderNotFoundException if order with {@code orderId} not
found
 */
void executeActiveOrderPartially(Long userId, Long orderId, float quantity)
throws OrderNotFoundException, UserNotFoundException;
/**
 * Returns user's active orders.
 * @return json array
 */
JSONArray getActiveOrders(Long userId);

```

3.2 Інтеграція симулятора із даними реальних ринків

Класичний спосіб розбору *JSON* — це делегування *JSON* бібліотеці та повернення структурованого представлення, наприклад *JSONObject*. Це представлення зазвичай є деревом об'єктів, яке видається після завершення аналізу. Безперервне створення тимчасових об'єктів створює додаткове навантаження на збирач сміття *JVM*, що може призвести до непередбачуваних стрибків затримки.

Використання бібліотеки з відкритим кодом *Chronicle-Wire* для аналізу є ефективною альтернативою. Він може отримувати всю необхідну інформацію з повідомлень *Binance JSON* без необхідності створювати додаткові об'єкти *Java*.

Примітивні дані можна проаналізувати без створення екземпляра в коробці, а текст можна завантажити в повторно використаний екземпляр *StringBuilder*. Масиви можна обробляти за допомогою викликів послідовності. Замість того, щоб накопичувати елементи в масиві, можна передати об'єкт функції, який викликається для кожного елемента.

Нижче наведено приклад коду, який отримує вищезгаданий *JSON* з Байненс: Метод для отримання активних ордерів з байненс, по обраній криптовалюти.

```
@Scheduled(fixedRate = 5000) // 5sec delay of each method invocation
public void matchOrders() throws IOException {
    BinanceApiWebSocketClient client = new
    BinanceApiWebSocketClientImpl(getSharedClient());
    client.onBookTickerEvent(symbol.toLowerCase(), new
    BinanceApiCallback < BookTickerEvent > () {
        @Override
        public void onResponse(BookTickerEvent response) throws
        UserNotFoundException {
            JSONArray activeOrders = orderService.getActiveOrders(1 L);
```

```

// Matches all active order starting from the oldest one with binance order
book,
// using brute force algorithm
for (int i = 0; i < activeOrders.length(); i++) {
JSONObject activeOrder = activeOrders.getJSONObject(i);
float orderPrice = parseFloat(activeOrder.getString("price"));
    float orderQuantity = parseFloat(activeOrder.getString("quantity"));
if (activeOrder.getString("side").equals("buy")) {
if (orderPrice >= parseFloat(response.getAskPrice())) {
if (parseFloat(response.getAskQuantity()) >= orderQuantity)
    orderService.executeActiveOrder(activeOrder.getLong("clientId"),
activeOrder.getLong("_id"));
        else
orderService.executeActiveOrderPartially(activeOrder.getLong("clientId"),
activeOrder.getLong("_id"),
parseFloat(response.getAskQuantity()));
    }
}
else if (activeOrder.getString("side").equals("sell")) {
if (orderPrice <= parseFloat(response.getBidPrice())) {
if (parseFloat(response.getBidQuantity()) >= orderQuantity)
    orderService.executeActiveOrder(activeOrder.getLong("clientId"),
activeOrder.getLong("_id"))
        else
orderService.executeActiveOrderPartially(activeOrder.getLong("clientOrde
rId"),
activeOrder.getLong("_id"),
parseFloat(response.getAskQuantity()));
    }
}
}
}

```

```

}
}
@Override
public void onFailure(final Throwable cause) {
cause.printStackTrace();
}
});
Closeable ws = client.onBookTickerEvent(symbol, new BinanceApiCallback
< BookTickerEvent > () {
@Override
public void onResponse(BookTickerEvent response) {}
@Override
public void onFailure(final Throwable cause) {
cause.printStackTrace();
}
});

```

Матчинг ордерів проходить згідно данній діаграмі:

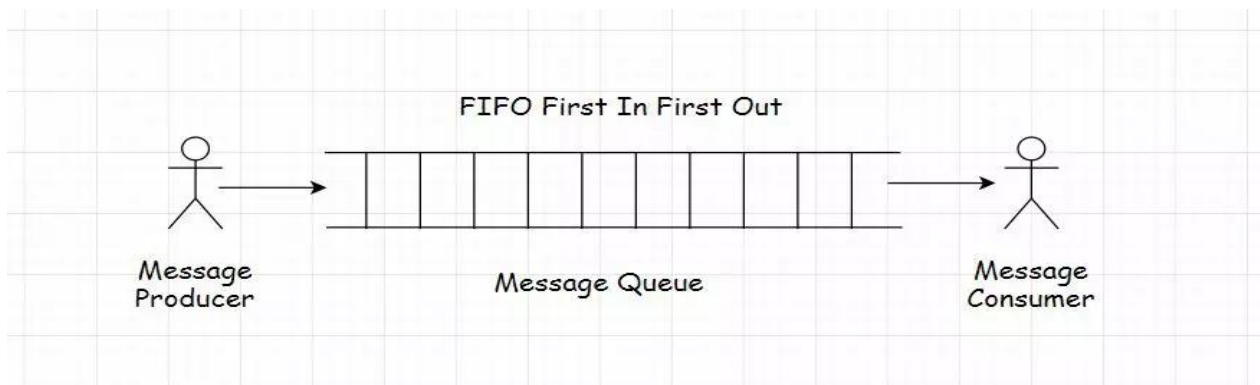


Рисунок 3.1- Діаграма Матчинг ордерів

Де найбільш недавні ордери перші порівнюються з ордерами користувача.

Такий підхід дозволяє швидко отримувати актуальні данні та реагувати на зміну руху цін.

Фреймворк *Chronicle-Services* дозволяє з'єднувачам ринкових даних працювати в одному циклі подій і розроблений для мінімізації затримки з

активним обертанням і блокуванням афінності, з потоками, що переміщуються між процесорами, контекстними перемикачами, запобігаючи втраті часу.

Бібліотека *HttpClient* у *Java 11* рекомендована для обробки підключень через веб-сокет, яка за замовчуванням використовує неблокуючий ввід-вивід і *TCP_NODELAY*. Ви також можете передати спеціальний *ExecutorService* для обробки підключень через веб-сокет.

Це добре працює з моделлю циклу подій, оскільки для виконання неблокуючої обробки в усіх обробниках ринкових даних потрібен лише один потік. Однопоточковий характер агрегаційного коду звільняє його від примітивів синхронізації, таких як блокування, опитування та операції *CAS*. Зменшення кількості блокувань і операцій *CAS* покращує загальну продуктивність системи.

3.3 Тестування та налагодження симулятора

Тестування та налагодження симулятора криптобіржі є важливим етапом в розробці програмного забезпечення для криптовалютної торгівлі. Цей розділ присвячений опису процесу тестування та налагодження, його цілям, методам і важливості забезпечення надійності та ефективності симулятора криптобіржі.

1. Цілі тестування та налагодження

Основною метою тестування та налагодження симулятора криптобіржі є забезпечення якості програмного забезпечення, його надійності та відповідності вимогам користувачів. Крім того, інші цілі включають:

—Виявлення та усунення помилок: Тестування допомагає ідентифікувати помилки, які можуть призвести до некоректної роботи симулятора криптобіржі. Після виявлення помилок, вони повинні бути виправлені для забезпечення стабільної та безперебійної роботи системи.

—Оцінка продуктивності: Тестування дозволяє виміряти продуктивність симулятора криптобіржі та ідентифікувати можливі проблеми

з швидкодією. Це важливо, оскільки швидка реакція системи на запити користувачів є одним із ключових факторів успіху для платформи криптовалютної торгівлі.

— Перевірка безпеки: Симулятор криптобіржі повинен бути захищеним від можливих атак і зловживань. Тестування допомагає виявити потенційні уразливості системи та розробити стратегії для їх запобігання.

2. Методи тестування

Для досягнення поставлених цілей використовуються різні методи тестування. Деякі з них включають:

— Юніт-тести: Цей метод спрямований на тестування окремих компонентів симулятора криптобіржі для перевірки їхньої правильності та функціональності.

— Інтеграційні тести: Вони перевіряють взаємодію між різними компонентами симулятора криптобіржі для виявлення можливих проблем при їхньому спільному використанні.

— Функціональні тести: Ці тести перевіряють, чи виконуються функції симулятора криптобіржі згідно з вимогами та специфікаціями.

— Навантажувальні тести: Цей вид тестування оцінює продуктивність симулятора криптобіржі при роботі з великим обсягом даних та запитів.

— Безпечність та зловмисники: Тестування безпеки використовується для виявлення потенційних проблем безпеки та уразливостей системи перед можливими атаками зловмисників.

3. Важливість надійності та ефективності.

— Надійність та ефективність симулятора криптобіржі мають велике значення для успіху платформи та задоволення користувачів. Неправильна робота або помилки можуть призвести до фінансових втрат для користувачів або псування репутації платформи.

— Забезпечення надійності означає, що симулятор криптобіржі повинен працювати стабільно та безперебійно навіть в умовах високого

торгового навантаження. Це вимагає ефективного моніторингу, обробки помилок та резервного копіювання даних.

—Ефективність важлива для забезпечення швидкої реакції на запити користувачів та обробки великого обсягу даних. Платформа повинна бути оптимізована для максимальної продуктивності та швидкодії, щоб забезпечити задоволення користувачів та уникнути втрати можливих угод через затримки.

В рамках проекту були використані всі типи тестів для затвердження якісної та надійної розробки та підтримки програми.

Висновок до третього розділу

У третьому розділі було розроблено проект симулятора криптовалютної біржі. Були визначені основні функціональні вимоги до симулятора, такі як реалістичне моделювання ринку, можливість торгувати різними криптовалютами та відстеження статистики трейдерської діяльності. Були визначені архітектура системи та використовувані технології.

РОЗДІЛ 4

ЕКОНОМІЧНА ЧАСТИНА

4.1 Організаційно-економічне та маркетингове обґрунтування проекту

Професійні трейдери та новачки (з США, Європи, Азії, України) шукають шляхи для отримання переваги та збільшення своїх прибутків. У такому конкурентному середовищі трейдери все більше віддають перевагу використанню симуляторів роботи криптобіржі.

Виконання науково дослідницької роботи по розробці симулятора роботи криптобіржі зробленої на *Java, SpringFramework*, в среде разработки *Intellij Idea*, база даних *Mysql* вимагає витрат. Витрати на реалізацію будь-якого проекту повинні постійно зменшуватись, адже в цьому і полягає її прогрес. Якщо цього немає, то можна зробити висновок, що робота не мала успіху, а її результат не можна вважати вигідним.

В даній роботі досліджено існуючі криптобіржі . Досліди були проведені на теоретичних даних. Основними результатом економічного дослідження поекта будуть - Строк окупності та рентабельність проекту.

Таблиця 4.1 - Класифікаційна оцінка проекту

Клас	Монопроект
Тип	Технічний
Вид	Дослідницький
Тривалість	Короткостроковий
за ступенем складності	Проект високої складності
Рівень	Корпоративний

Мета – Проектування та розробка симулятора роботи криптобіржі для тестування алгоритмів трейдингу, а також аналіз ринку та ризиків.

Результат – Програмний продукт для симуляції бізнес процесів криптобіржі.

Етапи виконання розділів кваліфікаційної роботи бакалавру:

1. Постановка технічного завдання. Термін виконання – чотири тижня.

2. Розробка робочого проекту: розробка логічної моделі програми, шляхом складання алгоритму. На стадії робочого проекту створюється вся необхідна документація об'єкта (специфікації, пояснювальна записка, інструкції та ін.). Термін виконання – один місяць.

3. Техніко-економічна частина: розрахунок витрат та обґрунтування економічної доцільності впровадження даного програмного продукту. Термін виконання – три тижні.

У кваліфікаційній роботі бакалавру представлений наступний склад робіт:

1. Технічне завдання. Термін виконання – від 10 до 14 днів.

2. Розробка алгоритму програмного продукту. Термін виконання – від 20 до 30 днів.

3. Розробка робочого проекту. Термін виконання – від 1 місяця до 45 днів.

4. Впровадження проекту. Термін виконання – за 1 тиждень.

Розрахунок ефективності оцінки системи управління полягає у знанні характеристик системи на момент розрахунку. Для кожної з оцінок створюється таблиця переведення діапазону можливих значень у шкалу балів в 1 до 5.

Таблиця 4.2 - Склад робіт по життєвому циклу проекту

№, код роботи	Назва роботи	T, дні
1-2	Збір та аналіз даних	21
2-3	Встановлення потреби в результатах	7
2-4	Затвердження концепцій	3
3-6	Визначення цілей, мотивів та вимог замовника та власника	21
4-5	Планування предметної області та інших елементів проекту	14
5-6	Розробка та затвердження зведеного плану	7
5-7	Організація виконання робіт	14
6-8	Інформаційний контроль за виконанням робіт	7
7-9	Детальне моделювання	30
7-8	Управління та координація робіт, коригування основних показників проекту	28
8-9	Підтвердження закінчення робіт	3
9-10	Експлуатаційні випробування остаточного продукту проекту	7
10-11	Підготовка звітів	7
11-12	Підготовка кадрів для експлуатації програмного продукту	5
12-13	Оцінка результатів проекту	7
13-14	Підготовка керівництва з користування програмного продукту	7

Надалі проведемо розрахунок параметрів мережевого графіку. Для цього визначимо: ранній строк здійснення роботи (T_{i0}); тривалість роботи (T_{j0}); ранній строк здійснення події (T_{pi}); раннє закінчення робіт (T_{po}), пізній строк здійснення події (T_{pj}); пізнє закінчення робіт (T_{po}), повний резерв часу роботи (R_j); вільний резерв часу роботи (R_c).

Таблиця 4.3 - Розрахунок параметрів мережного графіку

Попередня робота	Фактична робота	T _{ij}	T _{рн}	T _{ро}	T _{пн}	T _{по}	R _c	R _n	R _j
1	2	21	0	21	0	21	0	0	0
2	3	7	21	28	21	28	0	0	0
2	4	3	21	24	21	24	23	23	23
Попередня робота	Фактична робота	T _{ij}	T _{рн}	T _{ро}	T _{пн}	T _{по}	R _c	R _n	R _j
3	6	21	28	49	28	49	0	0	0
4	5	14	33	58	33	58	8	8	8
5	6	7	47	65	47	65	15	15	15
5	7	14	47	82	47	82	0	0	0
6	8	7	55	73	55	73	37	37	37
7	9	30	69	112	69	112	0	0	0
7	8	28	69	110	69	110	0	0	0
8	9	3	98	113	98	113	0	0	0
9	10	7	102	128	102	128	7	7	7
10	11	7	116	135	116	135	7	7	7
11	12	4	123	139	123	139	0	0	0
12	13	7	127	144	127	144	0	0	0
13	14	7	134	151	134	151	0	0	0

Тривалість критичного шляху 151 днів, не перевищує часу на проектування (153 днів), тому оптимізувати мережеву модель немає необхідності. В умовах даного проекту склад учасників буде наступним:

- автор и инициатор идеи (Курілко Артемій Дмитрович);
- замовник-інвестор (ТОВ «ГЕОНІК НЕТ»);
- керівник проекту (Ірина Вікторівна Колумба).

Для створення програмного продукту був виконаний аналіз існуючих методів збору та аналізу цінової політики торгових мереж.

Очікувані конкурентні переваги.

Даний продукт, являє собою простий і зручний у застосуванні інтерфейс, проста реалізація функцій, ефективність, що має забезпечити його затребуваність.

Автором проекту був проведений аналіз маркетингового ефекту з огляду на ринок розробників та споживачів інформаційних систем, ціни на розробки та переваги вашої розробки порівняно з аналогами:

Ринок розробників:

—*США:* США вважається одним з найбільших ринків розробників програмного забезпечення. Тут працюють багато досвідчених фахівців з розробки програмного забезпечення, які можуть бути зацікавлені у вашому проекті.

—*Європа:* Європейський ринок також має велику кількість талановитих розробників. Країни, такі як Велика Британія, Німеччина, Франція, Польща та Україна, мають розвинуту ІТ-галузь, що може стати потенційним джерелом спеціалістів для вашого проекту.

—*Азія:* Азійські країни, зокрема Індія, Китай та Японія, відомі своїми великими пулами розробників програмного забезпечення. Тут також можна знайти багато кваліфікованих фахівців.

Ринок споживачів:

—*Криптові біржі:* Споживачами вашого симулятора роботи криптобіржі можуть бути самі криптовалютні біржі, які зацікавлені в тестуванні та оптимізації своїх систем. Наприклад, *Binance, Coinbase, Bitstamp* тощо.

—*Фінансові установи:* Банки, інвестиційні компанії та хедж-фонди також можуть мати інтерес до вашого симулятора роботи криптобіржі.

—Вони могли б використовувати симулятор для тренування своїх фахівців та тестування стратегій торгівлі на криптобіржах.

Ціни на розробки:

Ціни на розробки програмного забезпечення можуть варіюватися залежно від регіону та рівня досвіду розробників. Наприклад, ціни на розробку програмного забезпечення в США та Європі можуть бути вищими порівняно з Азією та Україною. Проте, український ринок програмістів привабливий з точки зору ціни та якості.

Переваги вашої розробки порівняно з аналогами:

—*Технологічний стек:* Використання *Java, Spring Framework, IntelliJ IDEA* та *MySQL* бази даних є надійним та широко використовуваним технологічним стеком. Це дозволяє забезпечити високу продуктивність, масштабованість та стабільність вашої розробки.

—*Функціональні можливості:* симулятор роботи криптобіржі може включати широкий спектр функціональних можливостей, таких як створення та виконання торговельних стратегій, аналіз ринкових даних, управління портфелем та ризиками. Це дозволяє користувачам отримати реалістичне та повне відтворення роботи криптобіржі.

—*Зручний інтерфейс:* підкреслюється зручним та інтуїтивно зрозумілим інтерфейсом симулятора. Це дає користувачам зручну робочу зону та допомагає ефективно використовувати усі функції системи.

—*Гнучкість та налаштування:* запроектований симулятор може пропонувати гнучкі налаштування та можливість адаптації під конкретні потреби клієнта. Це означає, що користувачі зможуть налаштувати систему відповідно до своїх вимог та стратегій торгівлі.

—*Підтримка та оновлення:* виконання проекту на *Java, SpringFramework*, серед розробки *IntelliJ Idea*, база даних *Mysql* дає гарантовану підтримку та регулярні оновлення вашого симулятора. Це важливо для клієнтів, оскільки криптобіржі постійно змінюються, і їхні системи потребують актуальних оновлень та підтримки.

—*Безпека:* Забезпечення безпеки важливо для інформаційних систем, особливо коли мова йде про фінансові операції. Цей симулятор має надійні механізми безпеки, включаючи захист даних та аутентифікацію користувачів.

Окрім цього, автором проекту був проведений додатковий аналіз конкретних прикладів споживачів, щоб продемонструвати, як ваш симулятор роботи криптобіржі може використовуватися в реальних ситуаціях. Деякі приклади споживачів:

розробки, ступеня новизни та ступеня використання в розробці стандартних модулів на підставі формул 4.1 – 4.4:

$$T_{ТЗ} = T_p \cdot L_1 \cdot K_H \quad (4.1)$$

$$T_{ТП} = T_p \cdot L_2 \cdot K_H \quad (4.2)$$

$$T_{рп} = T_p \cdot L_3 \cdot K_H \cdot K_T \quad (4.3)$$

$$T_{вн} = T_p \cdot L_4 \cdot K_H \quad (4.4)$$

де T_p – укрупнення норма часу на розробку аналога ПЗ, чол/год, що коректується поправочним коефіцієнтом, що враховує умови розробки ПЗ, тобто в умовах комп'ютера, $K_H = 0,8$. У даному проекті $T_p = 312 \cdot 0,8 = 250$ чол/год;

L_j – питома вага i -го етапу розробки. У даному проекті залежно від ступеня новизни проекту (B): $L_1 = 0,12$; $L_2 = 0,15$; $L_3 = 0,5$; $L_4 = 0,12$;

K_H – поправочний коефіцієнт, що враховує ступінь новизни, у даній роботі $K_H = 1,2$;

K_T – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм, $K_T = 0,6$.

При розрахунках прийняті наступні об'єми розробленої документації по етапах проекту:

$N_{ТЗ} = 8$ – кількість сторінок технічного завдання;

$N_{ТП} = 55$ – кількість сторінок технічного проекту; $N_{рп} = 75$ – кількість сторінок робочого проекту;

$N_{інстр} = 10$ – кількість сторінок інструкції по налагодженню та в провадженню;

$N_{пр} = 93$ – кількість сторінок пояснювальної записки. Розрахунок трудомісткості розробки ІС:

1. Технічне завдання.

$$T_{ТЗ} = T_p \cdot L_1 \cdot K_H = 250 \cdot 0,12 \cdot 0,8 = 24;$$

$$T_{КК} = 0,7 \cdot N_{ТЗ} = 0,7 \cdot 8 = 5,6;$$

$$T_{НК} = 0,15 \cdot N_{ТЗ} = 0,15 \cdot 8 = 1,2.$$

2. Розробка технічного проекту (алгоритму й блок-схеми).

$$T_{ТЗ} = T_p \cdot L_2 \cdot K_H = 250 \cdot 0,15 \cdot 0,8 = 30;$$

$$T_{КК} = 0,7 \cdot N_{ТП} = 0,7 \cdot 55 = 38,5;$$

$$T_{НК} = 0,15 \cdot N_{ТП} = 0,15 \cdot 55 = 8,25.$$

3. Розробка робочого процесу.

$$T_{ТЗ} = T_p \cdot L_3 \cdot K_H \cdot K_T = 250 \cdot 0,5 \cdot 0,8 \cdot 0,6 = 60;$$

$$T_{КК} = 0,7 \cdot N_{ТП} = 0,7 \cdot 75 = 52,5;$$

$$T_{НК} = 0,15 \cdot N_{ТП} = 0,15 \cdot 75 = 11,25.$$

4. Налагодження і впровадження.

$$T_{ТЗ} = T_p \cdot L_4 \cdot K_H = 250 \cdot 0,12 \cdot 0,8 = 24;$$

$$T_{КК} = 0,7 \cdot N_{інст} = 0,7 \cdot 10 = 7;$$

$$T_{НК} = 0,15 \cdot N_{інст} = 0,15 \cdot 10 = 1,5.$$

5. Пояснювальна записка.

$$T_{ТЗ} = 1,5 \cdot N_{ПЗ} = 1,5 \cdot 93 = 139,5;$$

$$T_{КК} = 0,7 \cdot N_{ПЗ} = 0,7 \cdot 93 = 65,1; T_{НК} = 0,15 \cdot N_{ПЗ} = 0,15 \cdot 93 = 13,9$$

Всього:

$$T_p = 24 + 30 + 60 + 24 + 139,5 = 277,5;$$

$$T_{КК} = 5,6 + 38,5 + 52,5 + 7 + 65,1 = 168,7;$$

$$T_{НК} = 1,5 + 8,25 + 11,25 + 1,5 + 1,5 + 1,5 = 25,5;$$

$$T_{общ} = 277,5 + 168,7 + 25,5 = 471,7.$$

Тривалість розробки ІС у днях визначається за формулою 4.5:

$$T_{ПП} = \frac{\sum T_{ij}}{8 \cdot 0,73} \text{ дн} \quad , (4.5)$$

де $\sum T_{ij}$ – сумарна тривалість розробки, год;

8 – тривалість робочого дня (коефіцієнт переказу в робочі дні), год; 0,73

– коефіцієнт переведення в календарні дні;

T_{ij} – трудомісткість j-го виду робіт по

і-му етапу.

$$T_{\text{III}} = 471,7 / (8 \cdot 0,73) = 80,77 \text{ дн.}$$

Порівнюючи отримані результати з розрахунками тривалості розробки ІС за допомогою мережевої моделі, вибираємо для подальших розрахунків останнє значення, так як воно, на наш погляд, ближче до фактичної тривалості розробки ІС.

Визначення вартості програмного продукту.

Ціна визначається за формулою:

$$Ц = К \cdot \text{Вндр} + \text{Пр} , \quad (4.6)$$

Де Вндр – витрати на розробку програмної продукції (кошторисна собівартість);

$$К = 1,1.$$

Пр – нормативний прибуток, розраховується за формулою: $\text{Пр} = С \cdot R_n / 100$, (4.7) де R_n – норматив рентабельності, % $R_n = 25$ %; C_m – матеріальні витрати.

Вндр визначаються на підставі складання кошторису витрат на проведення НДР у таблиці 4.7.

При визначенні витрат на матеріали враховують: вартість сировини та матеріалів для проведення досліджень з урахуванням додаткових накладних витрат (витрат на транспорт, комісійних зборів тощо), вартість канцелярських матеріалів (паперів тощо), вартість ліцензій інтернет сервісів (табл. 4.3).

Таблиця 4.3 - Витрати на матеріал

Найменування матеріалів	Ціна, грн/од	Кількість	Сума, грн
Універсальна підписка Oracle Java SE (локальна)	5980	1	5980
Ліцензію від сервісу JetBrains*	3560	1	3560
Підписка бази даних Mysql	480	1	480
Папір	330	2	660
Ручки	22,4	2	44,8
Інформаційні носії	540	1	540
Дорога	20	12	240
Разом			11504

Отримуємо ліцензію від сервісу *JetBrains* - підписка *dotUltimate* терміном на 1 рік, яка поєднує ліцензування користування універсальною рамкою *SpringFramework* та середовищем розробки *Intellij Idea*.

Витрати по заробітній платі визначаються як сума заробітної плати усіх учасників НДР. Орієнтовний склад учасників, ступінь їх участі у НДР та заробітна плата наведені у таблиці 4.4.

Таблиця 4.4 - Орієнтовний склад учасників НДР, їх заробітна плата та ступінь участі

Учасник НДР	Місячна заробітна плата, грн	Тривалість роботи, днів	Ступінь участі, %
Студент – дослідник Науковий	7000	150	100
Керівник роботи	12000	44	25

Орієнтовний склад учасників НДР, їх заробітна плата та ступінь участі.

Основна заробітна плата виконавця з урахуванням окладу і часу.

$$V_{зп} = \frac{\sum Z_i \cdot K_0 \cdot \tau_i}{D_p} = \frac{(7000 \cdot 150) + (12000 \cdot 0,25 \cdot 44)}{22} = 52344,64 \text{ грн}$$

Де Z_i – середньомісячний оклад;

D_p – середня кількість робочих днів ($D_p = 22$); τ_i - трудомісткість робіт ($\tau_i=150$);

K_0 – коефіцієнт обліку окладу керівників і консультантів проекту

($K_0=0,1$).

$$V_{зд} = 52344,64 \cdot 0,1 = 5234,5 \text{ грн.}$$

Єдиний соціальний внесок беруть у розмірі 22 % від величини заробітної

$$C_{сс} = 0, \text{ Додаткова заробітна плата } C_{зд} = C_{зо} \cdot K_d$$

де K_d - коефіцієнт відрахувань на заробітну плату,

$$K_d = 0,1$$

$$0,22 \cdot (52344,64 + 5234,5) = 12667,4 \text{ грн.}$$

Амортизаційні відрахування S_a беруть від вартості основних виробничих фондів за встановленими нормативами до кожної групи фондів, які використовують при проведенні НДР (основного та додаткового обладнання, комп'ютерної техніки, інших фондів, крім приміщення). Амортизаційні відрахування необхідно розраховувати, виходячи з терміну.

Таблиця 4.5 - Розрахунок сум амортизаційних відрахувань за відповідний рік

Група ОЗ	Елементи основних засобів	Первісна Вартість, грн	На, %	Вартість,грн.
1	Комп'ютерна техніка	20000	25	5000
Всього				5000

Річна вартість споживаної електроенергії $S_{эл}$, визначається по формулі: $S_{эл} = M_y * T_{ко} * C_e * K_i$, де:

M_y - установлена сумарна потужність комп'ютерного устаткування, кВт;

$T_{ко}$ - річний фонд роботи ЕОМ з урахуванням часу на профілактичні огляди;

C_e - вартість 1 кВт-години ел. енергії;

K_i - коефіцієнт інтенсивного використання потужності (К и реком. =0,9)

$$S_{эл} = 0.6 * 1224 * 3,45 * 0,9 = 2280.31$$

Отже, витрати на електроенергію складуть 2280.31 грн.

Таблиця 4.6 - Кошторис витрат на проведення прикладних НДР

Найменування статей витрат	Сума витрат, грн
1. Матеріали	11504
2. Зароботна плата (основна і додаткова)	52344,64
3. Єдиний соціальний внесок	12667,4
4. Амортизаційні відрахування	5000
5. Витрати на електроенергію	2280,31
а. Виробнича собівартість	83796,3
б. Адміністративні витрати 3,5% від виробничої собівартості	2 932,8
с. Витрати на збут – 1% від виробничої собівартості	837,9
6. Всього повна собівартість	87567

$$P_p = (87567 - 11504) \cdot 0,25 = 19015,75 \text{ грн.}$$

Ціна програмного продукту складе:

$$C = 1,1 \cdot 87567 + 19015,75 = 115\,339,45 \text{ грн.}$$

4.3 Визначення капітальних витрат

Розрахунок капітальних витрат, пов'язаних з впровадженням (вдосконаленням) ІС здійснюється за формулою:

$$K = K_{\Pi} + K_{\text{ко}} + K_{\text{во}} + K_c, \quad (4.8)$$

K_{Π} – довиробничі витрати;

$K_{\text{ко}}$ – вартість комп'ютерного устаткування: $K_{\text{ко}} = 5000$ грн.; $K_{\text{во}}$ – вартість допоміжного устаткування: $K_{\text{во}} = 500$ грн.;

K_c – вартість будівництва у зв'язку з впровадженням: $K_c = 0$.

$K_{\Pi} = C = 117016,72$ грн. – перед виробничі витрати:

$$K_{\Pi} = 115\,339,45 + 5000 + 500 + 0 = 120\,839,45 \text{ грн.}$$

Розрахунок поточних (експлуатаційних) витрат:

$$C = C_{\text{опл}} + C_a + C_{\text{ел}} + C_{\Pi} + C_p + C_{\text{всп}}, \quad (4.9)$$

$C_{\text{опл}}$ – річний фонд основної і додаткової оплати праці персоналу;

C_a – сума річних амортизаційних відрахувань від вартості основного і допоміжного устаткування;

$$C_{\text{опл}} = Z_{\text{осн}} + Z_{\text{доп}} + Z_{\text{есв}}, \quad (4.11)$$

Разом:

$$C_{\text{опл}}^{\text{до}} = 288000 + 28800 + 69696 = 386496 \text{ грн.}$$

$$C_{\text{опл}}^{\text{після}} = 96000 + 9600 + 23232 = 128832 \text{ грн.}$$

Розрахунок амортизаційних відрахувань визначається за формулою:

$$C_a = K_{\text{кко}} \cdot H_a / 100$$

Де H_a – норма амортизаційних відрахувань ($H_a = 60 \%$).

$$C_a = 5000 \cdot 0,6 = 3000 \text{ грн.}$$

Річна вартість споживаної електроенергії, визначається за формулою:

$$C_{\text{ел}} = M_y \cdot T_{\text{ко}} \cdot Ц_{\text{э}} \cdot K_{\text{и}} \quad (4.12)$$

M_y – установлена сумарна потужність комп'ютерного устаткування:

$$M_y = 0,6 \text{ кВт};$$

$T_{\text{ко}}$ – річний фонд роботи ЕОМ з урахуванням часу на профілактичні огляди:

$$T_{\text{ко}} = 5500 \text{ год};$$

$Ц_{\text{э}}$ – вартість 1 кВт – години ел. енергії 3,45 грн;

$K_{\text{и}}$ – коефіцієнт інтенсивного використання потужності $K_{\text{п}} = 0.9$.

$$C_{\text{ел}} = 0,6 \cdot 5500 \cdot 3,45 \cdot 0,9 = 10246,5 \text{ грн.}$$

Разом:

$$C = C_{\text{опл}} + C_a + C_{\text{ел}} + C_{\text{р}} + C_{\text{п}}:$$

$$C_{\text{до}} = 386496 \text{ (тільки вартість труда)}$$

$$C_{\text{після}} = 128832 + 3000 + 10246,5 + 300 + 100 + 2500 = 144950,5 \text{ грн.}$$

Де \mathcal{E}_r – річна економія на поточних витратах;

$K_{\text{п}}$ – передвиробничі витрати на проект: $K_{\text{п}} = 120\,839,45 \text{ грн.};$

$E_{\text{н}}$ – нормативний коефіцієнт: $E_{\text{н}} = 0,25;$

$$\Xi_r = (C^{\text{до}} - C^{\text{після}}) + \Delta\Pi.$$

$\Delta\Pi$ – приріст прибутку господарчого суб'єкта та його структурного розподілу $\Delta\Pi = 0$

$$\Xi_r = 386496 - 144950,5 + 0 = 241545 \text{ грн.}$$

$$\Xi_o = 241545,2 - 0,25 \cdot 120\,839,45 = 211\,335 \text{ грн.}$$

Рентабельність інвестицій:

$$E = \frac{\Xi_r}{K_n} \cdot 100\% = \frac{241545,2}{K_n} \cdot 100\% = 199\% \quad (4.14)$$

199% > 25% - проект ефективний.

Строк окупності витрат проекту складає - до одного року

Таблиця 4. 7 - Техніко-економічні показники

Показники	Одиниця виміру	Значення показника	
		до впровадження	після впровадження
Трудомісткість розробки проекту	дні	-	151
Ціна ПП	грн.	-	115 339,45
Капітальні витрати	грн.	-	120 839,45
Поточні витрати	грн./рік	386494	144950
Економічний ефект від реалізації	грн./рік	-	241545,2
Строк окупності	роки	-	0,5
Рентабельність проекту	%	-	199

Висновки до четвертого розділу

Техніко-економічні показники проекту показують:

- Трудомісткість розробки не перевищують запланований термін;
- Ціна ПП, капітальні та поточні витрати дозволяють окупити проект в термін до одного року;
- Рентабельність проекту показує високу якість ценової та комерційності стійкості проекту

Економічна ефективність цього проекту є перевагою порівняно зі створенням власного розробленого симулятора або використанням комерційних рішень.

Економічні показники проекту дають можливість пропонувати доступні та конкурентні ціни, які дозволять клієнтам зекономити кошти при отриманні якісного симулятора.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ

5.1 Загальні вимоги до особистого робочого місця

Сучасний розвиток технічного та технологічного стану виробництва передбачає постійну автоматизацію та оптимізацію виробничих процесів. Сьогодні, напевно, важко уявити компанію, господарська діяльність в якій здійснювалась би без використання комп'ютерної техніки. Через масовий характер робіт, що виконуються працівниками за допомогою комп'ютера, законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки на підприємстві, безпосередньо й охорона праці при роботі з комп'ютером.

Вимоги до приміщення. Приміщення, в яких планується установка та подальша робота з комп'ютером, повинні відповідати проектній документації будинку, погодженій з уповноваженими державними органами. Крім того, роботодавець повинен враховувати санітарні нормативи освітлення, вимоги до параметрів мікроклімату (температура, відносна вологість), ступеня і сили вібрації, звукового шуму і вогнестійкості приміщення, а також характеристики електромагнітного, ультрафіолетового та інфрачервоного полів. Конкретні показники зазначених санітарних норм див. в Державних санітарних правилах і нормах роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98, затверджених Постановою Головного державного санітарного лікаря України №7 від 10 грудня 1998 року.

Правила поширюються на умови й організацію праці при роботі з візуальними дисплейними терміналами (ВДТ) усіх типів вітчизняного та зарубіжного виробництва на основі електронно-променевиx трубок (ЕПТ), що використовуються в електронно-обчислювальних машинах (ЕОМ) колективного використання та персональних ЕОМ (ПЕОМ).

Так, наприклад, роботодавцю заборонено встановлювати комп'ютери в приміщеннях, розташованих у підвалах будинків. Для уникнення можливих

аварій та замикань, поряд з приміщеннями, де вестиметься робота з комп'ютером (над чи під ними), також не дозволяється проведення робіт, що потребують здійснення надмірно вологих технологічних процесів. Відповідне приміщення повинно бути укомплектоване системами центрального або індивідуального опалення, кондиціонування чи вентиляції повітря. Але при установці зазначених систем, необхідно переконатись, що батареї опалення, водопровідні труби, вентиляційні кабелі тощо, надійно сховані під захисними щитками, які перешкоджатимуть можливому потраплянню робітника під напругу.

У кожній кімнаті, де обладнуватимуться робочі місця співробітників, що працюватимуть на комп'ютері, повинні бути наявні елементи природного та штучного освітлення. При цьому, на вікнах слід встановити легко регульовані жалюзі чи штори, які дозволять працівникам коригувати рівень освітлення в приміщенні. Бажано розмістити комп'ютери в кімнаті таким чином, щоб світло потрапляло на екрани моніторів з півдня чи північного сходу. З метою досягнення максимального рівня безпеки і охорони праці при роботі з комп'ютером, виробничі приміщення необхідно обладнати аптечками першої медичної допомоги, системами автоматичної пожежної сигналізації і вогнегасниками. В приміщенні, в якому разом працюють 5 або більше комп'ютерів, на видимому місці встановлюється службовий вимикач, який у разі потреби дозволить повністю відключити електричне живлення кімнати.

Вимоги до особистого робочого місця працівника. Роботодавець, який використовує найману працю робітників, повинен забезпечити відповідність їхніх робочих місць комфортним та безпечним умовам. Розмір одного робочого місця має становити не менше 6 квадратних метрів. При необхідності, суміжні робочі місця співробітників, що працюють з комп'ютером, слід розділити перегородками висотою до 2 метрів. При визначенні достатнього розміру приміщення і робочого місця на одну особу необхідно додатково враховувати шафи, сейфи, тумби або інші предмети меблів чи обладнання, які знаходяться

в кімнаті. На столі працівника можливо розмістити допоміжні для роботи пристрої (принтери, колонки, сканери), а також місця для зберігання документів, за умови, що це не обмежуватиме видимість екрану і не заважатиме працівнику.

У разі надмірного шуму чи вібрації технічного обладнання, роботодавець повинен забезпечити працівників антивібраційними килимками. Робочий стілець співробітника має бути підйомно-поворотним, легко регульованим за висотою та забезпечувати належну підтримку та зручне положення спини і хребта особи. Щодня необхідно проводити вологе прибирання приміщення, та очищати робоче місце та безпосередньо монітор комп'ютера від запиленості.

На підприємстві забороняється: проводити ремонт та технічне обслуговування комп'ютера за робочим місцем працівника; самочинно ремонтувати або намагатись здійснити технічне налагодження комп'ютера без залучення компетентних спеціалістів; складувати на робочому місці зайві документи, деталі та предмети, що не потрібні для роботи; використовувати монітори з нечітким зображенням та монітори, у яких наявні поламки екрану; працювати з матричним принтером без антивібраційного покриття та зі знятою кришкою. Допускати до роботи осіб, які не пройшли затвердження на підприємстві курс охорони праці для роботи з комп'ютером, не дозволяється.

Соціальні та профілактичні засоби захисту робітників, які працюють з комп'ютером. При прийнятті на роботу кожна особа має пройти лікарський огляд. Окрім того, при подальшій трудовій діяльності в компанії, така особа підлягає регулярному лікарському огляду не рідше ніж раз на 2 роки. Обов'язковим є проходження таких лікарів як терапевта, невропатолога та офтальмолога.

В компанії мають бути чітко встановлені перерви для відпочинку працівників (окрім обідньої), як правило, тривалістю 10-15 хвилин раз на годину або дві, в залежності від складності роботи.

В будь-якому випадку, роботодавець повинен передбачити такий розпорядок роботи на підприємстві, щоб час неперервної роботи з комп'ютером був не більше ніж 4 години. Додатково, для збереження належного рівня здоров'я та професійної придатності робітників, рекомендується виділити на підприємстві окреме побутове приміщення для перепочинку працівників і зняття ними нервово-емоційного напруження, що виникає при роботі з комп'ютером.

5.2 Загальні вимоги безпеки при роботі з комп'ютером та іншою оргтехнікою

1. До самостійної роботи з комп'ютером, ноутбуком, принтером, ксероксом, сканером, плазмовою панеллю, LCD-дисплеєм та іншою оргтехнікою допускаються особи, які досягли 18 річного віку, пройшли медичний огляд, ознайомлені з інструкцією з охорони праці при роботі з оргтехнікою, не мають протипоказань за станом здоров'я.

2. Під час роботи на комп'ютері та іншій оргтехніці на робітники можуть впливати наступні небезпечні та шкідливі фактори:

- a. електрострум і випромінювання;
- b. перенапруження зору під час роботи з електронними пристроями, монітором, особливо при нераціональному розташуванні екрана по відношенню до очей.

3. Освітлювальні установки повинні забезпечувати рівномірне освітлення і не повинні утворювати засліплюючих відблисків на клавіатурі, а також на екрані монітора за напрямом очей.

4. При роботі з комп'ютером, принтером, ксероксом та іншою периферійною технікою не допускається розташування робочого місця в приміщеннях без природного освітлення, без наявності природної або штучної вентиляції.

5. Робоче місце з комп'ютером та оргтехнікою повинно розміщуватися на відстані не менше 1м від стіни, від стіни з віконними

отворами - на відстані не менше 1,5 м.

6. Кут нахилу екрана монітора або ноутбука по відношенню до вертикалі повинен складати 10-15 градусів, а відстань до екрана – 500-600 мм.

7. Кут зору екрана повинен бути прямим і становити 90 градусів.

8. Для захисту від прямих сонячних променів повинні передбачатися сонцезахисні пристрої (плівка з металізованим покриттям, регульовані жалюзі з вертикальними панелями та ін).

9. Освітлення повинно бути змішаним (природним та штучним).

10. У приміщенні кабінету і на робочому місці необхідно підтримувати чистоту і порядок, проводити систематичне провітрювання.

11. Про всі виявлені під час роботи несправності обладнання необхідно доповісти керівнику, у випадку поломки необхідно припинити роботу до усунення аварійних обставин. При виявленні можливої небезпеки, попередити оточуючих та негайно повідомити керівнику; утримувати в чистоті робоче місце, не захащувати його сторонніми предметами.

12. Про нещасний випадок очевидець, працівник, який його виявив, або сам потерпілий повинні доповісти безпосередньо керівникові установи і вжити заходів з надання медичної допомоги.

13. Особи, винні в порушенні вимог, вимагаємих даною інструкцією з охорони праці при роботі з комп'ютером, принтером, ксероксом та іншою оргтехнікою, притягаються до дисциплінарної відповідальності у відповідності з чинним законодавством.

5.3 Вимоги безпеки перед початком роботи з комп'ютером (ноутбуком) та іншою оргтехнікою

1. Оглянути і переконатися у справності обладнання, електропроводки. У разі виявлення несправностей, до роботи не приступати. Повідомити про це керівника і, тільки після усунення несправностей і його дозволу, приступити до роботи.

2. Перевірити освітлення робочого місця, за необхідності, вжити

заходів до його нормалізації.

3. При виявленні будь-яких несправностей, комп'ютер та оргтехніку не вмикати і негайно повідомити про це головному з охорони безпеки.

4. Ретельно провітрити приміщення з персональним комп'ютером та оргтехнікою, переконатися, що мікроклімат у приміщенні знаходиться в допустимих межах: температура повітря в холодний період року 22-24°C, в теплий період року – 23-25°C, відносна вологість повітря – 40-60%.

5. Включити монітор і перевірити стабільність і чіткість зображення на екрані, переконатися у відсутності запаху диму від комп'ютера та оргтехніки.

5.4 Вимоги безпеки під час роботи з комп'ютером та іншою оргтехнікою

Вмикайте і вимикайте комп'ютер, ноутбук та іншу оргтехніку тільки вмикачами, забороняється проводити вимкнення витягуванням вилки з розетки.

— Забороняється знімати захисні пристрої з обладнання і працювати без них.

— Не допускати до комп'ютера та оргтехніки сторонніх осіб, які не беруть участі в роботі

— Забороняється переміщати та переносити системний блок, монітор, принтер, будь-яке обладнання, яке знаходиться під напругою.

— Забороняється під час роботи пити будь-які напої, приймати їжу.

— Забороняється будь-яке фізичне втручання у пристрій комп'ютера, принтера, сканера, ксерокса під час їх роботи.

— Забороняється залишати включене обладнання без нагляду.

— Забороняється класти предмети на комп'ютерне обладнання, монітори, екрани та оргтехніку.

— Суворо виконувати загальні вимоги з електробезпеки та пожежної

Змн.	Арк.	№ докум.	Підпис	Дата	

безпеки.

—Під час усунення застрявання паперу на ксероксі чи принтері, задля уникнення ураження електрострумом, необхідно відключити обладнання від електромережі. Необхідно також вимикати обладнання від мережі при тривалому простої.

—Під час регламентованих перерв, з метою зниження нервово-емоційного напруження, стомлення зорового аналізатора, усунення впливу гіподинамії та гіпокінезії, запобігання розвитку познотонічного стомлення, слід виконувати комплекси вправ для очей або організувати фізкультурні паузи.

—Комп'ютер, будь-які його периферійні пристрої, оргтехніку необхідно використовувати у суворій відповідності з експлуатаційною документацією до них.

—Під час виконання роботи необхідно бути уважним, не звертати уваги на сторонні речі.

—Про всі виявлені несправності та збої в роботі апаратури необхідно повідомити безпосередньо інженера з обслуговування комп'ютерної техніки.

5.5 Вимоги безпеки після закінчення роботи з комп'ютером, принтером, ксероксом, сканером та іншою оргтехнікою

1. Вимкнути комп'ютер, ноутбук, телевізор, плазмову панель, LCD-екран, принтер, ксерокс, сканер, колонки та іншу оргтехніку від електромережі, для чого необхідно вимкнути тумблери, а потім акуратно витягнути штепсельні вилки з розетки.

2. Протерти зовнішню поверхню комп'ютера чистою вологою тканиною. При цьому не допускайте використання розчинників, одеколону, препаратів в аерозольній упаковці.

3. Прибрати робоче місце. Скласти диски у відповідне місце зберігання.

3. негайно відключити електроприлад від мережі живлення;
4. повідомити про те, що трапилась пожежа адміністрації установи і в найближчу пожежну частину за телефоном 101;
5. приступити до гасіння осередка загоряння за допомогою первинних засобів пожежогасіння.

При отриманні травми слід надати першу допомогу, при необхідності відправити його до лікувальної установи і повідомити про це адміністрацію.

Висновки до п'ятого розділу

Сформовані вимоги безпеки, а саме:

- вимоги безпеки при роботі з комп'ютером та іншою оргтехнікою;
- вимоги безпеки перед початком роботи з комп'ютером;
- вимоги безпеки під час роботи з комп'ютером, ноутбуком;
- вимоги безпеки після закінчення роботи з комп'ютером;
- вимоги техніки безпеки та безпеки життєдіяльності в аварійних ситуаціях при роботі з комп'ютером та іншою оргтехнікою;
- вимоги безпеки в аварійних ситуаціях при роботі з комп'ютером

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Introduction in coaching and professional management" - Valeria Topchu
2. Learn PyCharm course [Електронний ресурс]. – Режим доступу:
<https://www.jetbrains.com/help/pycharm/quick-start-guide.html>
3. Laragon Documentation [Електронний ресурс]. – Режим доступу:
<https://laragon.org/docs/index.html>
4. SQLite Documentation [Електронний ресурс]. – Режим доступу:
[://www.sqlite.org/docs.html](https://www.sqlite.org/docs.html)
5. Бази даних – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85
6. SQLite Architecture – SQLite – Вікіпедія [Електронний ресурс].
[:https://ru.wikipedia.org/wiki/SQLite#/media/%D0%A4%D0%B0%D0%B9%D0%BB:SQLite_Architecture.svg](https://ru.wikipedia.org/wiki/SQLite#/media/%D0%A4%D0%B0%D0%B9%D0%BB:SQLite_Architecture.svg)
7. Laravel – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Laravel>
8. Парсер: що це таке простими словами, приклади реклами [Електронний ресурс]. – Режим доступу: <https://www.activetraffic.ru/wiki/parser/>
9. Фактори ціноутворення [Електронний ресурс]. – Режим доступу:
<https://sendpulse.ua/ru/support/glossary/pricing>
10. Ціноутворення Вікіпедія [Електронний ресурс]. – Режим доступу:
<https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D0%BE%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5>
11. Робота з MongoDB, її основні функції [Електронний ресурс] -
<https://www.freecodecamp.org/news/learn-mongodb-a4ce205e7739/>
12. Робота з MySQL, створення та редагування записів [Електронний ресурс] - <https://blog.devart.com/mysql-command-line-client.html>
13. Принцип роботи криптовалюти [Електронний ресурс] -

<https://www.forbes.com/advisor/in/investing/cryptocurrency/what-is-cryptocurrency-and-how-does-it-work/>

14. Вибір технологій для криптобіржи [Електронний ресурс] - <https://qoden.com/what-makes-a-good-cryptocurrency-exchange-programming-language/>

15. Принцип роботи криптобіржі [Електронний ресурс] - <https://cointelegraph.com/learn/what-is-a-cryptocurrency-a-beginners-guide-to-digital-money>

16. IntelliJ Idea швидкий старт, курс для роботи [Електронний ресурс] - <https://www.udemy.com/course/intellij-idea-secrets-double-your-coding-speed-in-2-hours/>

17. Java, огляд мови [Електронний ресурс] - https://www.java.com/ru/download/help/whatis_java.html

18. Ознайомлення з синтаксисом джави [Електронний ресурс] - <https://metanit.com/java/tutorial/>

19. Визначення підходу програмування для обраної мови програмування [Електронний ресурс] - <https://highload.today/oop-v-java-chetyre-printsipa-s-primerami/>

20. Робота многопоточної програми [Електронний ресурс] - <https://javarush.com/groups/posts/1992-mnogopotchnostjh-v-java-sutjh-pljusih-i-chastihe-lovushki->

21. Spring, огляд фреймворку [Електронний ресурс] - <https://habr.com/ru/articles/490586/>

22. Взаємодія з базами даних через спринг [Електронний ресурс] - <https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa>

23. Будівництво REST архітектури [Електронний ресурс] - <https://www.baeldung.com/rest-with-spring-series>

24. Вивчення роботи по розкладу [Електронний ресурс] - <https://www.baeldung.com/spring-scheduled-tasks>

25. Binance документація [Електронний ресурс] - <https://binance->

