

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Управління проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

На тему:

**“Дослідження характеристик проекту створення додатку для торгівлі
криптовалютою на платформі Android”**

Студентки 2-го курсу групи УПз-21

Кравець Анастасії Станіславівни

Науковий керівник:

Кандидат технічних наук, професор

Морозов Віктор Володимирович

(підпис студента)

(дата)

(підпис)

Попередній захист:

(Висновок: “До захисту в Екзаменаційній комісії”)

**Завідувач кафедри
технологій управління**

(підпис)

(прізвище, ініціали)

(дата)

Київ-2023

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій**

Кафедра технологій управління
Освітній рівень магістр
Спеціальність 122 Комп'ютерні науки
Освітня програма Управління проектами

ЗАТВЕРДЖУЮ
Завідувач кафедри
Професор Морозов В. В.

“ ___ ” _____ 20__ року

**З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студентка: Кравець Анастасія Станіславівна

Група: УПз -21

1. Тема кваліфікаційної роботи:

Дослідження характеристик проекту створення додатку для торгівлі криптовалютою на платформі Android

Затверджена на засіданні кафедри технологій управління "26" червня 2023 року, протокол №14

2. Строк подання студентом готової роботи – “ ___ ” _____ 20__ р.

3. Цільова установка та вихідні дані до роботи: Цільова установка проекту полягає в розробці високоефективного та зручного додатку для торгівлі криптовалютою на платформі Android, спрямованого на задоволення потреб сучасних інвесторів. Ми прагнемо забезпечити користувачам швидкий та безпечний доступ до криптовалютних ринків, надаючи зручний інтерфейс для виконання операцій купівлі, продажу та моніторингу портфеля. Вихідні дані включають детальний аналіз потреб цільової аудиторії, вимоги до функціоналу, а також вивчення конкурентної обстановки в галузі торгівлі криптовалютою на мобільних платформах.

4. Зміст роботи: дослідження та обґрунтування доцільності та життєздатності проекту, опис концепції проекту, етап проектування, розробки, тестування та впровадження додатку, випуск готового продукту на ринок.

5. Перелік графічного матеріалу (слайдів): аналіз предметної області, опис концепції проекту, кошторис проекту, організаційна структура команди, цілі

проекту, структура робіт проекту, діаграма Ганта, графік проведення мітингів, ризики проекту, прототип інтерфесу.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	План виконання роботи
1	Вивчення літературних джерел з предмету дослідження	18.09.2023 - 25.09.2023
2	Збір і вивчення матеріалів для розробки додатку	26.09.2023 - 02.10.2023
3	Складання розгорнутого плану кваліфікаційної роботи	03.10.2023 – 06.10.2023
4	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін.	09.10.2023
5	Підготовка розділу 1 «Дослідження та обґрунтування доцільності та життєздатності проекту»	10.10.2023 – 23.10.2023
6	Підготовка розділу 2 «Опис концепції проекту»	24.10.2023 – 07.11.2023
7	Підготовка розділу 3 «Етап проектування, розробки, тестування та впровадження додатку»	08.11.2023 – 14.11.2023
8	Підготовка розділу 4 «Випуск готового продукту на ринок»	15.11.2023 – 30.11.2023
9	Оформлення кваліфікаційної роботи	01.12.2023 – 07.12.2023
10	Передача кваліфікаційної роботи рецензенту для рецензування	08.12.2023
11	Передача кваліфікаційної роботи науковому керівнику	09.12.2023
12	Попередній захист кваліфікаційної роботи	12.12.2023
13	Захист кваліфікаційної роботи	26.12.2023

Дата видачі завдання “_____” _____ 20__ р.

Керівник роботи кандидат технічних наук, професор

Морозов Віктор Володимирович

(підпис)

Завдання прийняла до виконання студентка групи УПз-21

Кравець Анастасія Станіславівна

(підпис)

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЕКТУ	13
1.1. Аналіз предметної галузі.....	13
1.2. Характеристика індустрії та тенденції розвитку	13
1.3. Аналіз продуктів конкурентних установ.....	16
1.4. Формулювання проблемної області.....	18
1.5. Формулювання наукової новизни та інноваційності проекту.....	19
1.6. PEST-аналіз проекту	21
РОЗДІЛ 2. ОПИС КОНЦЕПЦІЇ ПРОЕКТУ	23
2.1. Визначення цілей та обсягу проекту.....	23
2.2. Етап збору вимог.....	26
2.2.1. Ідентифікація зацікавлених сторін.....	28
2.2.2. Збір функціональних та нефункціональних вимог.....	30
2.3. Постановка задачі, формулювання технічного завдання на розробку у вигляді паспорту проекту.....	31
2.4. Розробка концептуальної моделі мобільного додатку.....	35
2.5. Опис математичної моделі проекту та математична постановка задачі.....	38
2.6. Аналіз та вибір методології управління проектом. Класифікація моделей управління ІТ проектами.	39
2.6.1. Обґрунтування вибору моделі управління для проекту розробки ..	41
2.7. Команда та кошторис проекту	43
2.7.1. Організаційна структура команди проекту	43
2.7.2. Кошторис проекту	44
2.8. Управління змістом проекту. WBS і роботи проекту	48
2.9. Управління часом проекту. Діаграма Ганта.....	50
2.9.1. Календар мітингів	52
2.10. Управління ризиками проекту.....	53
РОЗДІЛ 3. ЕТАП ПРОЕКТУВАННЯ, РОЗРОБКИ, ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ ДОДАТКУ	57
3.1. Вибір технологій та архітектури системи	57
3.2. Проектування прототипу, інтерфейсу, та бази даних.....	62

3.2.1. Розробка концептуальної моделі бази даних проєкту	68
3.2.2. Побудова логічної моделі бази даних проєкту	74
3.3. Написання коду	78
3.4. Планування тестів та тестування.....	81
3.5. виправлення помилок	85
РОЗДІЛ 4. ВИПУСК ГОТОВОГО ПРОДУКТУ НА РИНОК	88
4.1. Впровадження продукту.....	88
4.2. Підтримка користувачів	90
4.3. Завершення проєкту.....	92
4.4. Експлуатація та технічне обслуговування додатку	94
4.4.1. Моніторинг ефективності	95
4.4.2. Вдосконалення та оновлення.....	97
ВИСНОВКИ	99
СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.....	101

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:
“Дослідження характеристик проекту створення додатку для торгівлі криптовалютою на платформі Android”

Студентка: Кравець Анастасія Станіславівна
Науковий керівник: Морозов Віктор Володимирович
Рік захисту - 2023.

Темою даної роботи було обрано **“Дослідження характеристик проекту створення додатку для торгівлі криптовалютою на платформі Android”**, предметною областю якої є розробка програмного забезпечення, технічні аспекти, безпека, інтерфейс користувача, інтеграція з біржами, вивчення ринку, правові аспекти, тестування та маркетинг.

Метою підготовки роботи є дослідження та аналіз характеристик проекту створення додатку для торгівлі криптовалютою на платформі Android з метою визначення оптимальних стратегій розробки, інтеграції та впровадження, спрямованих на ефективне використання в умовах сучасного ринку криптовалют.

Ціль проекту - розробка та успішне впровадження функціонального та безпечного додатку для торгівлі криптовалютою на платформі Android, сприяючи зручності користувачів та відповідаючи вимогам сучасного ринку цифрових фінансових послуг.

Наукова новизна роботи полягає у комплексному дослідженні технічних, безпекових, інтерфейсних та маркетингових аспектів розробки додатку для торгівлі криптовалютою на платформі Android, що враховує високі стандарти безпеки фінансових операцій та споживацької зручності. Новизною також є впровадження та оптимізація механізмів інтеграції з блокчейн-технологією для ефективного ведення та моніторингу транзакцій у режимі реального часу, що робить проект актуальним та конкурентоспроможним на ринку цифрових фінансових послуг.

Кваліфікаційна робота складається з анотації, вступу, основної частини, яка включає чотири розділи, висновків, списку використаних джерел та додатків.

В першому розділі проводиться аналіз ринкових та соціальних потреб для визначення актуальності розробки додатку для торгівлі криптовалютою на платформі Android; вивчаються технічні та економічні вигоди проекту, зокрема його конкурентоспроможність та можливість задоволення потреб цільової аудиторії; аналізуються ризики та можливості, пов'язані з ринковим середовищем та конкретною специфікою торгівлі криптовалютою. Також обґрунтовується життєздатність проекту, визначаються стратегії мінімізації ризиків та забезпечення сталого розвитку; формулюються ключові висновки, які визначають важливість та доцільність реалізації даного проекту у визначених умовах.

Другий розділ присвячено детальному опису функціональних можливостей розроблюваного додатку та його унікальних особливостей. Визначається цільова аудиторія та створюється портрет користувача з урахуванням специфіки торгівлі криптовалютою, розглядається концепція взаємодії з користувачем, включаючи інтерфейс та важливі елементи користувацького досвіду. Визначаються ключові технічні та інноваційні рішення, які визначають унікальність та конкурентоспроможність проекту. Формулюється загальний концептуальний фреймворк проекту, що служить основою для подальших етапів розробки.

В третьому розділі наводиться опис проектування, розглядається архітектура системи та обираються необхідні технічні засоби для реалізації функціоналу. Здійснюється розробка програмного забезпечення, включаючи вибір програмних інструментів та мов програмування. Проводяться етапи тестування, включаючи модульне, інтеграційне та системне тестування, з метою визначення та усунення можливих помилок. Додатково описується процес впровадження додатку, включаючи підготовку до запуску та взаємодію

з потенційними користувачами, визначається план підтримки та подальшого розвитку додатку, а також оцінюються його перспективи на ринку.

В четвертому розділі визначаються стратегії маркетингу та рекламної кампанії для привертання уваги цільової аудиторії. Розглядається підготовка матеріалів для випуску на ринок, включаючи документацію та рекламні матеріали. Проводиться аналіз реакції ринку та коригування стратегій, якщо необхідно. Також, визначаються критерії успіху та проводяться остаточні тестування перед релізом на ринку. Формулюється план подальшого розвитку та підтримки продукту після випуску.

За результатами роботи було зроблено висновки. Детальний комплексний аналіз продукту, плану та процесів проекту, здійснений в результаті даного дослідження, допомагає прийти до висновку, що подальший розгляд проекту, як такого що має високий потенціал, інноваційність та дуже перспективного з великою ймовірністю успішної реалізації вважається доцільним. У роботі представлено акт впровадження даного додатку.

Робота містить 100 сторінок без додатків, 24 рисунки та 3 таблиці. Додатки складають 33 сторінки.

Ключові слова: криптовалюта; торгівля криптовалютою; мобільний додаток; Android; фінансові технології; тенденції розвитку ринку криптовалют; характеристики проекту; технологічні аспекти розробки; планування проекту; етапи розробки та впровадження; економічний аналіз; фінансові додатки; ефективність додатку; ризики та заходи безпеки; інновації в галузі фінтеху; стратегії маркетингу для фінансових додатків; регулювання криптовалютного ринку; глибокий аналіз ринкових конкурентів.

ВСТУП

В епоху стрімкого розвитку технологій та фінансових інновацій, ринок криптовалют стає важливим компонентом глобальної фінансової системи. У цьому контексті мобільні додатки, зокрема ті, які дозволяють здійснювати торгівлю криптовалютою на платформі Android, набувають особливого значення для інвесторів та учасників фінансового ринку. Аналіз характеристик проекту створення таких додатків стає ключовим завданням, спрямованим на вивчення та оптимізацію їхньої ефективності.

Актуальність теми полягає у тому, що в сучасному світі технології та фінанси тісно переплітаються, і ринок криптовалют стає не тільки інструментом для збереження та зростання капіталу, але й об'єктом для розвитку інноваційних технологій. Мобільні додатки, спрямовані на торгівлю криптовалютою, стають каталізатором для швидкого та зручного доступу до цього ринку.

Мета дослідження – провести глибокий аналіз характеристик проекту створення мобільного додатку для торгівлі криптовалютою на платформі Android. З'ясувати тенденції розвитку ринку криптовалют, вивчити технологічні аспекти розробки фінансових додатків, а також визначити оптимальні етапи планування, розробки та впровадження подібного проекту.

Предмет дослідження - характеристики та перспективи проекту створення мобільного додатку для торгівлі криптовалютою на платформі Android: аналіз тенденцій розвитку ринку криптовалют, технологічні аспекти розробки фінансових додатків і визначення оптимальних етапів планування, розробки та впровадження.

Реалізація поставлених завдань передбачає проведення комплексного підходу до розробки та впровадження додатку для торгівлі криптовалютою. Детальніше розглянемо кожен етап цього процесу:

1. Огляд існуючих додатків для торгівлі криптовалютою:
 - Проведення аналізу популярних додатків на ринку для з'ясування їхніх особливостей та недоліків.

- Визначення ключових функціональних та дизайнерських аспектів, які привертають користувачів.
2. Визначення цілей та обсягу проекту:
 - Розроблення чіткої та обгрунтованої мети проекту, що визначає його основні завдання та підсумки.
 - Визначення обсягу проекту, включаючи функціональні та технічні вимоги.
 3. Аналіз технологічних аспектів:
 - Дослідження сучасних технологій, які використовуються у фінансових додатках та торгівлі криптовалютою.
 - Визначення оптимальних технічних рішень для забезпечення швидкодії, безпеки та зручності користування.
 4. Збір функціональних та нефункціональних вимог:
 - Проведення зустрічей з зацікавленими сторонами для отримання детальних вимог до функціоналу додатку.
 - Формалізація вимог у вигляді документу, який визначає функції та характеристики системи.
 5. Проектування архітектури додатку:
 - Визначення оптимальної структури для продукту, враховуючи вимоги до його функціоналу.
 - Створення дизайну користувацького інтерфейсу та розробка структури бази даних.
 6. Вибір технологій та розробка ПЗ:
 - Визначення технологій та інструментів для реалізації проекту.
 - Написання програмного забезпечення відповідно до визначених вимог та проекту.
 7. Тестування та впровадження на ринок:
 - Розроблення стратегії тестування та створення плану тестування.

- Проведення різноманітних тестів, включаючи функціональне, інтеграційне та валідаційне тестування.
- Впровадження додатку на ринок через розгортання та маркетингові заходи.

Об'єкт дослідження – це процес створення та функціонування додатку для торгівлі криптовалютою на Android. Предмет – характеристики проекту, які визначають його успішність та ефективність в умовах сучасного фінансового ринку.

Матеріал дослідження включатиме як теоретичні аспекти – аналіз ринку, огляд технологічних підходів, так і практичні результати – огляд існуючих додатків та розробка власного.

Основні методи дослідження включають в себе аналіз літературних джерел, вивчення існуючих додатків, аналіз технологічних вимог, тестування та апробацію результатів.

Теоретичне значення даного дослідження полягає у глибокому аналізі та систематизації актуальної інформації про ринок криптовалют та технологій розробки додатків для платформи Android. Дослідження ринку криптовалют дозволить зрозуміти сучасні тенденції, різновиди криптовалют, їхню популярність та основні технічні аспекти.

Аналіз технологій розробки додатків для Android зорієнтований на виявлення найбільш ефективних інструментів та підходів у цьому контексті. Враховуючи швидкий розвиток технологій, такий аналіз стане важливим джерелом інформації для розробників та компаній, що планують створення або оптимізацію фінансових додатків для Android.

Практичне значення полягатиме у можливості надання конкретних рекомендацій для оптимізації розробки та функціонування подібних проектів. Отримані результати дослідження нададуть базу для розроблення ефективних стратегій вибору технологій, визначення ключових функцій додатку та підвищення його конкурентоспроможності на ринку. Рекомендації також

зможуть сприяти зменшенню ризиків та покращенню загального виконання проектів у сфері розробки фінансових додатків для Android.

Структура роботи: робота складається зі вступу, чотирьох розділів, висновку, списку використаної літератури (56 джерел) та 27 додатків. Повний обсяг дослідження – 138 сторінок.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЕКТУ

1.1. Аналіз предметної галузі

Зростання популярності мобільних додатків в галузі криптотрейдингу стає невід'ємною частиною сучасного інвестиційного середовища. Трейдери шукають зручні та ефективні інструменти для ведення операцій з криптовалютами, і мобільні додатки дозволяють їм торгувати в будь-який час та в будь-якому місці.

Криптовалюти, такі як Bitcoin, Ethereum та інші, привертають увагу інвесторів своєю унікальністю та потенціалом для високих прибутків. Огляд характеристик криптовалют та їх взаємодії з мобільними додатками для трейдингу допомагає зрозуміти, як краще використовувати ці технології для досягнення інвестиційних цілей.

Існує різноманітність мобільних додатків для криптотрейдингу, від простих портфельних додатків до складних торгових платформ. Різноманіття функцій та можливостей додатків визначається потребами користувачів, і вивчення даного розділу допомагає з'ясувати, які саме характеристики є важливими для трейдерів.

1.2. Характеристика індустрії та тенденції розвитку

Криптовалюта — цифровий або віртуальний актив, що використовує шифрування для забезпечення безпечних транзакцій та контролю створення нових одиниць. Вона заснована на технології блокчейн, що дозволяє безпечно записувати та відстежувати транзакції [25].

Основні риси криптовалют:

- *Децентралізація:* Функціонує поза централізованими фінансовими установами, такими як банки.
- *Шифрування:* Забезпечує конфіденційність та безпеку транзакцій шляхом криптографічного шифрування.

- *Обмежена емісія*: Більшість криптовалют мають обмежену кількість одиниць - це означає що в майбутньому їх кількість не збільшиться.

Як працює криптовалюта:

- *Блокчейн*: Транзакції записуються в блоках та формують ланцюжок, який неможливо змінити.
- *Майнінг*: Майнери вирішують складні завдання для підтвердження транзакцій та створення нових блоків.
- *Ключі*: Кожен користувач має публічний та приватний ключі для отримання та відправлення коштів.

Застосування криптовалют:

- *Міжнародні транзакції*: Швидкі та ефективні грошові перекази без посередництва банків.
- *Інвестиції*: Використовується як інвестиційний актив для зберігання та зростання капіталу.
- *Смарт-контракти*: Автоматизовані програми, які виконують угоди без посередництва.

Криптовалюти представляють різноманітний світ цифрових активів з унікальними характеристиками та застосуванням. Деякі із найбільш актуальних різновидів включають:

- *Bitcoin (BTC)*: Піонер та лідер ринку, використовується як цифровий актив та засіб обміну вартості.
- *Ethereum (ETH)*: Відомий своєю платформою для смарт-контрактів та децентралізованих додатків (DApps).
- *Binance Coin (BNB)*: Спочатку призначений для внутрішніх операцій на біржі Binance, здобуває популярність іншими застосуваннями.
- *Ripple (XRP)*: Спрямований на оплату міжнародних транзакцій та співпрацює з фінансовими установами.
- *Cardano (ADA)*: Зацікавлений у наданні більшої безпеки та масштабованості.





















#	МОНЕТА	ЦІНА	КАПІТАЛІЗАЦІЯ	ВИПУЩЕНО	ОБ'ЄМ ЗА 24Ч	ПОПРАВКА 24 ГОДИНИ	ГРАФІК ЦІНИ (30Д)
1	 Bitcoin BTC	41 717.65	816 053 004 571\$	19 561 362 BTC	37 885 126 998\$	+2.55% +1 063.80\$	
2	 Ethereum ETH	2 227.82\$	267 851 686 736\$	120 230 295 ETH	15 757 370 421\$	+0.67% +14.93 \$	
3	 Прив'язати USDT	0.999755 \$	89 792 109 359\$	89 814 116 890 доларів США	73 750 270 434\$	-0.06% -	
4	 Binance Coin BNB	231.345 \$	35 094 598 185\$	151 698 076 BNB	2 140 911 254\$	+0.88% +2.04\$	
5	 XRP XRP	0.619037\$	33 358 999 156\$	53 888 571 585 XRP	1 954 973 148\$	-1.65% -0.01 \$	
6	 Солана СОНЦЕ	60.1698\$	25 533 151 357\$	424 351 910 SOL	2 207 021 361\$	-5.84% -3.51 \$	
7	 Монета доларів ... USDC	0.999763 \$	24 368 314 462\$	24 374 081 974 доларів США	7 211 062 729\$	-0.02% -	
8	 Кардано €	0.403795 \$	14 259 030 893\$	35 312 508 598 ADA	540 586 166\$	-0.27% -	
9	 Dogecoin ДОЖ	0.090083 \$	12 801 028 831\$	142 103 066 383 DOGE	1 407 234 606\$	+4.16% -	
10	 Трон TRX	0.103408\$	9 152 521 659\$	88 509 215 013 TRX	328 235 560\$	-0.23% -	

Рис. 1.1. Рейтинг криптовалют станом на 2023р. (Додаток А)

Динамічний світ криптовалют визначають певні тенденції та явища:

- *Децентралізовані Фінанси (DeFi)*: Зростання інтересу до проектів DeFi, які надають фінансові послуги поза рамками традиційних інститутів.
- *NFT (Non-Fungible Tokens)*: Популярність унікальних цифрових активів, таких як мистецькі та ігрові NFT.
- *Екосистеми Блокчейнів*: Розширення екосистем, об'єднуючи різні блокчейни для підтримки обміну та інтеграції.

На сучасному ринку криптовалют спостерігається висока волатильність (це показник, який характеризує тенденції зміни ринкових цін і доходів впродовж певного часу). Тенденції включають активний розвиток децентралізованих фінансів (DeFi), популярність унікальних цифрових активів (NFT) та розширення екосистем блокчейнів [42].

Ситуація на ринку є динамічною. Волатильність залишається високою через вплив різних факторів, таких як регуляторні рішення, інституційні

інвестиції та глобальні події. Інституційний інтерес викликає зростання легітимізації та визнання криптовалют.

Також продовжують розвиватися технології DeFi та концепції NFT, що формує нові фінансові можливості та ринки. Регуляційний підхід різних країн визначає динаміку ринку та його подальший розвиток.

1.3. Аналіз продуктів конкурентних установ

Огляд існуючих мобільних додатків для торгівлі криптовалютою на платформі Android показує, що на сьогоднішній день існує значна кількість додатків, які надають користувачам можливість торгувати криптовалютами прямо зі своїх мобільних пристроїв. Деякі з найпопулярніших та надійних додатків в цій категорії включають:

1. *Binance*:

- Один з найпопулярніших та широко використовуваних додатків для торгівлі криптовалютою.
- Забезпечує доступ до ринків широкого спектру криптовалют.
- Функціональність мобільного додатку повністю відповідає веб-версії, що забезпечує однаковий функціонал та зручність використання.

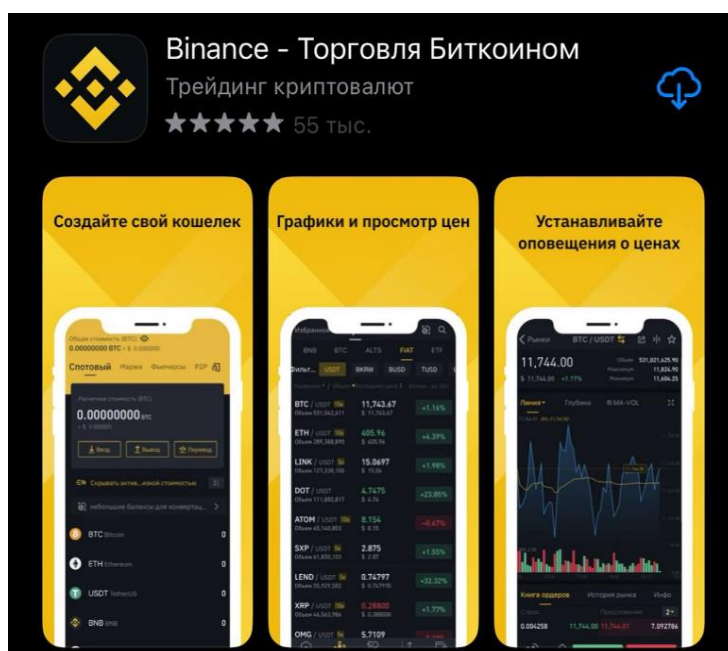


Рис. 1.2. Мобільний додаток «Binance» (Додаток Б)

2. Coinbase:

- Популярний сервіс для купівлі, продажу та зберігання криптовалют.
- Має інтуїтивно зрозумілий інтерфейс, що робить його ідеальним для початківців.
- Забезпечує безпечну та швидку торгівлю популярними криптовалютами.

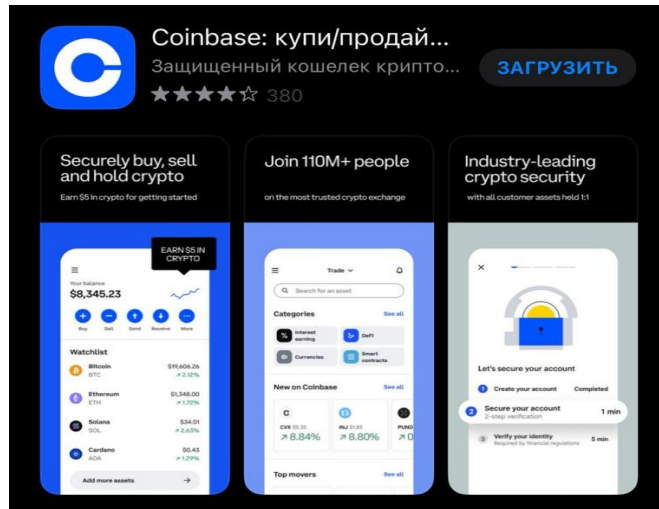


Рис. 1.3. Мобільний додаток «Coinbase» (Додаток В)

3. Kraken:

- Платформа, що відзначається великим вибором криптовалют та просунутими функціями торгівлі.
- Надає різні типи ордерів та аналітичні інструменти для трейдерів.
- Забезпечує високий рівень безпеки та захисту особистих даних користувачів.

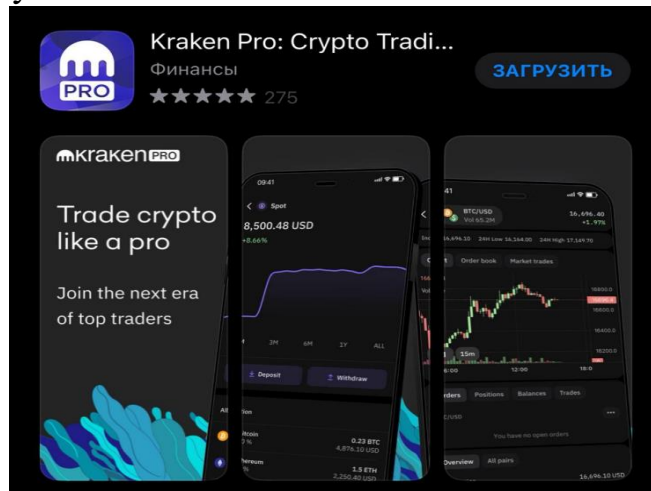


Рис. 1.4. Мобільний додаток «Kraken» (Додаток Г)

Зростання конкуренції на ринку крипто додатків сприяє впровадженню новітніх технологій. Використання штучного інтелекту, блокчейну та інших інноваційних рішень допомагає поліпшити функціонал додатків та забезпечити їхню ефективність.

Безпека та конфіденційність даних – важливі аспекти для мобільних додатків, які працюють з фінансовою інформацією. Визначення заходів, спрямованих на забезпечення безпеки та конфіденційності, є ключовим етапом аналізу.

Завершуючи аналіз та огляд популярних додатків для торгівлі криптовалютою, враховуючи відгуки користувачів, можна узагальнити ключові переваги та недоліки кожного з розглянутих застосунків. Це допомагає визначити кращі практики та уникнути можливих недоліків при розробці нового додатку для крипто трейдингу на Android.

1.4. Формулювання проблемної області

Проблемна область, яку розглядає дана робота, полягає в потребі в розробці нового мобільного додатку для торгівлі криптовалютою на платформі Android, який би відповідав потребам та вимогам сучасних трейдерів. Незважаючи на наявність існуючих мобільних додатків для торгівлі криптовалютами, існують певні недоліки та обмеження, які можуть обмежувати користувачів у їх торговельних операціях.

Огляд існуючих додатків виявив такі ключові проблеми:

- *Технічні проблеми:* Деякі додатки можуть стикається з недоліками у швидкодії, нестабільною роботою або відсутністю підтримки оновлень. Це може призводити до затримок в обробці замовлень, втрати з'єднання з ринком або навіть аварійного завершення програми, що створює нестабільні умови для трейдерів.
- *Проблеми функціональності:* Деякі додатки можуть бути обмежені у функціоналі, не надаючи повний спектр інструментів для аналізу ринку, стеження за портфелем або доступу до певних типів замовлень.

Обмежена функціональність може обмежувати можливості трейдерів у реалізації своїх стратегій та обмежувати їх потенціал у досягненні бажаних результатів.

- *Проблеми зручності використання:* Інтерфейси деяких додатків можуть бути складними у використанні або мають неінтуїтивну навігацію. Нестача простоти та зрозумілості може створювати перешкоди для користувачів, ускладнювати процес торгівлі та призводити до помилок. Крім того, недостатня зручність використання може займати багато часу для оволодіння додатком та утримувати користувачів від досягнення оптимального досвіду торгівлі.
- *Проблеми безпеки:* Безпека є однією з найважливіших аспектів додатків для торгівлі криптовалютою. Деякі додатки можуть мати слабкі механізми захисту даних та особистої інформації користувачів, що ставить під загрозу їх криптовалютні активи. Проблеми безпеки можуть включати недостатній рівень шифрування, незахищене зберігання паролів або недостатні заходи безпеки щодо доступу до облікових записів трейдерів [6].

Ці проблеми обмежують ефективність та зручність торгівлі криптовалютою на платформі Android, а також піддають ризику безпеку та конфіденційність користувачів. Вирішення цих проблем є актуальним завданням для розробки нового мобільного додатку, який пропонуватиме удосконалені технічні можливості, розширений функціонал, зручний інтерфейс та надійні механізми безпеки для задоволення потреб трейдерів та забезпечення їх успіху у світі криптовалют.

1.5. Формулювання наукової новизни та інноваційності проекту

Наукова новизна та інноваційність проекту "Розробка проекту створення андроїд додатку для торгівлі криптовалютою" проявляється в наступних аспектах:

1. *Впровадження нових технологій:* Проект передбачає використання сучасних технологій розробки мобільних додатків на платформі Android. Це включає використання мов програмування, фреймворків та бібліотек, які дозволяють створити потужну, швидку та стабільну програму для торгівлі криптовалютою.
2. *Розширений функціонал:* Проект має на меті розробити додаток з розширеним функціоналом, який включатиме можливості аналізу ринку, моніторингу цін, виконання різних типів замовлень, управління портфелем та багато іншого. Це дозволить трейдерам здійснювати торговельні операції з більшою ефективністю та точністю.
3. *Забезпечення безпеки:* Однією з ключових складових проекту є забезпечення високого рівня безпеки користувачів та їх криптовалютних активів. Це включає розробку механізмів шифрування даних, захисту від несанкціонованого доступу, безпечного зберігання облікових записів та інших заходів, які гарантують конфіденційність та безпеку трейдерів.
4. *Інтуїтивний інтерфейс:* Проект ставить за мету розробку інтуїтивно зрозумілого та зручного інтерфейсу додатку. Це дозволить трейдерам легко орієнтуватись у функціоналі, швидко здійснювати операції та виконувати аналітичні завдання. Для досягнення цього мети, в проекті передбачається детальне проектування інтерфейсу з урахуванням потреб та звичок користувачів.
5. *Інтеграція з різними криптовалютними біржами:* Додаток буде підтримувати інтеграцію з різними криптовалютними біржами, що дозволить трейдерам здійснювати торговельні операції на різних ринках та використовувати різні торгові платформи. Це забезпечить більший вибір та гнучкість для трейдерів, що покращить їх можливості для отримання прибутку [7].

Загалом, проект виходить за рамки існуючих додатків для торгівлі криптовалютою на платформі Android, пропонуючи нові можливості, підвищений рівень безпеки та зручності використання. Він орієнтований на

покращення досвіду трейдерів і надання їм інструментів для успішної торгівлі криптовалютою [14].

1.6. PEST-аналіз проекту

PEST-аналіз (аналіз політичних, економічних, соціокультурних та технологічних чинників) допомагає оцінити зовнішнє середовище, в якому розвиватиметься проект "Розробка проекту створення андроїд додатку для торгівлі криптовалютою". Основна мета аналізу - визначення факторів, які можуть вплинути на успішність та стабільність проекту [28]. Нижче наведено загальний огляд PEST-аналізу:

Політичні фактори:

- Регулювання криптовалют: Зміни у законодавстві та політичному підході до криптовалют можуть вплинути на торгівлю та використання криптовалют на платформі Android.
- Криптовалютні обмеження: Політика та регулювання, що обмежують доступ до криптовалют або обмежують їх використання, можуть становити виклик для проекту.

Економічні фактори:

- Фінансова стабільність: Економічна ситуація та фінансова стабільність можуть вплинути на інтерес користувачів до торгівлі криптовалютою та їх готовність вкладати кошти у додаток.
- Зміна курсу криптовалют: Волатильність курсу криптовалют може вплинути на дохідність трейдерів та популярність додатку для торгівлі криптовалютою.

Соціокультурні фактори:

- Зростання інтересу до криптовалют: Загальний інтерес до криптовалют та їх використання може створити сприятливе соціокультурне середовище для розвитку додатку.

- Сприйняття безпеки: Свідомість користувачів щодо безпеки та конфіденційності у сфері криптовалют може вплинути на їх готовність використовувати додаток для торгівлі.

Технологічні фактори:

- Розвиток мобільних технологій: Постійне вдосконалення мобільних платформ та технологій може створити сприятливі умови для розробки та використання мобільного додатку для торгівлі криптовалютою.
- Конкуренція на ринку додатків: Присутність інших мобільних додатків для торгівлі криптовалютою на платформі Android створює конкурентне середовище, яке потрібно врахувати та знайти особливості, що забезпечать конкурентоспроможність проекту.

Аналіз політичних, економічних, соціокультурних та технологічних чинників допоможе ідентифікувати важливі та впливові фактори, які слід враховувати при розробці та впровадженні додатку для торгівлі криптовалютою на платформі Android. Оцінка цих факторів допоможе забезпечити успішну реалізацію проекту та виявити можливості для покращення та конкурентного переваги [21].

РОЗДІЛ 2. ОПИС КОНЦЕПЦІЇ ПРОЕКТУ

2.1. Визначення цілей та обсягу проекту

Ціль проекту є основною метою, яку необхідно досягти через реалізацію проекту. Це конкретне, визначене твердження про те, що ви плануєте досягти після завершення всіх робіт. Ціль часто формулюється як результат, який можна виміряти, і відображає головні прагнення команди або організації [13].

Мета нашого дослідження – провести глибокий аналіз характеристик проекту створення мобільного додатку для торгівлі криптовалютою на платформі Android. З'ясувати тенденції розвитку ринку криптовалют, вивчити технологічні аспекти розробки фінансових додатків, а також визначити оптимальні етапи планування, розробки та впровадження подібного проекту.

Обсяг проекту визначає роботи, які потрібно виконати для досягнення цілі. Це включає в себе всі завдання, функції, функціональності та доставки, необхідні для успішного завершення проекту. Обсяг допомагає уникнути невизначеності та чітко визначити межі того, що входить до проекту і що за його межами [36].

Також важливо визначити ролі та обов'язки учасників проекту, розподілити ресурси, визначити графік виконання робіт, а також розробити систему контролю якості та оцінки результатів. Усі ці аспекти входять в поняття планування проекту, що є ключовим етапом перед початком виконання робіт [53].

Цільовими завданнями даного проекту є:

1. Створення функціонального додатку та забезпечення безпеки та конфіденційності операцій.
2. Інтеграція ключових функцій для ефективної торгівлі та відстеження торговельних операцій.
3. Підтримка різних криптовалют та платіжних систем.
4. Аналіз та визначення поточних тенденцій ринку для прогнозу ринкових тенденцій.

5. Оцінка ризиків та розробка стратегій управління для забезпечення стабільності та надійності фінансового додатку.
6. Визначення вимог до безпеки та конфіденційності даних користувачів і розробка заходів для їх забезпечення.
7. Реалізація механізмів взаємодії з іншими фінансовими платформами для покращення функціональності.
8. Розробка стратегії підтримки користувачів та створення каналів комунікації для вирішення їхніх питань та проблем.
9. Вивчення та врахування законодавчого середовища для забезпечення відповідності вимогам законодавства.
10. Забезпечення масштабованості для розширення функціоналу та підтримки зростання користувачів.
11. Впровадження моніторингових засобів для стеження за ефективністю та безперервної роботи.
12. Розробка маркетингової стратегії для просування на ринку та залучення нових користувачів.
13. Вивчення та імплементація новітніх технологій для підтримки конкурентоспроможності.
14. Постійне вдосконалення функціоналу та інтерфейсу на основі звітів від користувачів та аналізу трендів.

Обсяг проекту: Обсяг проекту включає в себе розробку та впровадження повноцінного мобільного додатку для оптимальної торгівлі криптовалютою на платформі Android. Ключові етапи охоплюють:

1. Планування проекту:

Процес планування проекту розпочинається із чіткого визначення його цілей та обсягу. Першим етапом є збір вимог, включаючи ідентифікацію зацікавлених сторін і визначення їхніх очікувань від продукту. На цьому етапі також проводяться ряд зустрічей із зацікавленими сторонами для отримання детальних вимог.

Одночасно з цим формується команда проекту, обирається керівник і встановлюються комунікаційні процедури. Важливо набрати в команду фахівців із різних областей, які доповнюють один одного та забезпечать всебічний підхід до вирішення завдань.

Цей етап визначає основні параметри проекту, що будуть служити основою для подальших етапів розробки та впровадження [19].

2. Проектування та розробка додатку:

Визначення технічних засобів, які будуть використані для розробки додатку, включаючи мови програмування, фреймворки та загальну архітектуру системи. На цьому етапі вирішуються ключові аспекти взаємодії компонентів та вибір оптимальних технічних рішень [2].

Створення дизайну та інтерфейсу користувача для забезпечення зручного та ефективного взаємодії з додатком. Розробка структури бази даних, включаючи визначення типів даних та забезпечення безпеки даних. Обидва аспекти спрямовані на забезпечення ефективної роботи додатку та задоволення потреб користувачів [46].

3. Реалізація ключових модулів:

Розробка механізмів торгівлі та аналізу ринкових даних включає в себе створення механізмів, які дозволяють користувачам взаємодіяти з ринковими даними в режимі реального часу. Це включає в себе розробку інструментів аналізу, графіків та засобів прийняття рішень для ефективної торгівлі [16].

Інтеграція систем безпеки та аутентифікації - забезпечення високого рівня безпеки та конфіденційності шляхом впровадження механізмів аутентифікації та авторизації користувачів. Включає розробку систем захисту даних, щоб запобігти несанкціонованому доступу та забезпечити безпеку операцій [38].

4. Тестування та виправлення помилок:

Проведення різноманітних тестів для перевірки функціональності включає в себе проведення модульних, інтеграційних та системних тестів для перевірки коректності роботи окремих компонентів, їх взаємодії та загальної функціональності продукту.

Виправлення виявлених помилок та оптимізація продукту - після виявлення помилок на попередньому етапі виконується їх аналіз та виправлення. Крім того, проводиться оптимізація продукту з метою підвищення продуктивності, ефективності та коректності роботи [11].

5. *Впровадження та підтримка:*

Запуск додатку на ринок включає в себе етапи розгортання та впровадження додатку, забезпечуючи його доступність для користувачів. Важливо враховувати маркетингові стратегії та взаємодію з аудиторією для забезпечення успішного запуску.

Організація сервісної підтримки для користувачів - розробляється стратегія та система обслуговування користувачів, включаючи надання відповідей на їхні питання, вирішення проблем та забезпечення зручних каналів зв'язку. Оновлення та підтримка продукту відбувається відповідно до отриманих відгуків та змін у вимогах користувачів [33].

Цей проект має на меті створення високоякісного та конкурентоздатного додатку для торгівлі криптовалютою, що відповідає вимогам сучасного ринку та враховує потреби користувачів.

2.2. Етап збору вимог

Етап збору вимог у проекті є ключовим етапом, що передує розробці. На цьому етапі здійснюється детальний аналіз та уточнення потреб користувачів та визначення очікувань щодо функціоналу та характеристик системи. Бізнес-вимоги описують, як проект вирішує бізнес-задачі та допомагає досягти поставлених цілей [8].

Вимоги до проекту поділяються на кілька видів, кожен з яких визначає різні аспекти функціональності та властивостей системи:

1. Функціональні вимоги:

- Описують конкретні функції та операції, які система повинна виконувати.
- Визначають, як система має взаємодіяти з користувачем та іншими системами.
- Включають у себе функції, які мають бути доступні, опціональні функції та умови їх виклику.

2. Нефункціональні вимоги:

- Спрямовані на характеристики та якості системи, а не на конкретні функції.
- Включають у себе аспекти продуктивності, безпеки, доступності, надійності, сумісності тощо.
- Ці вимоги можуть визначати обмеження щодо продуктивності системи або технічні стандарти, якими вона повинна дотримуватися.

3. Бізнес-вимоги:

- Визначають, як проект сприяє досягненню стратегічних цілей організації.
- Можуть включати економічні аспекти, такі як вартість проекту та очікувану прибутковість.

4. Зовнішні вимоги:

- Пов'язані з зовнішніми факторами, які можуть вплинути на систему, такими як легісляційні вимоги, стандарти промисловості, екологічні обмеження тощо.

5. Операційні вимоги:

- Визначають, як система повинна функціонувати в реальному середовищі.

- Включають у себе вимоги до продуктивності, відновлюваності, обслуговування та інші аспекти, які забезпечують успішну експлуатацію системи [27].

Кожен з цих видів вимог грає важливу роль у визначенні, проектуванні та впровадженні проекту, забезпечуючи повноту та точність визначення функціональності та характеристик системи.

Збір вимог включає консультації з зацікавленими сторонами, аналіз існуючих документів, проведення інтерв'ю та опитувань. Отримані вимоги документуються в специфікаціях вимог, які узгоджуються з усіма стейкхолдерами.

Важливо забезпечити зрозумілість вимог, валідувати їх правильність, а також врегулювати конфлікти та невідповідності. Також важливим є захист від змін вимог, який передбачає стійкість вимог та документування будь-яких змін з урахуванням впливу на проект.

Цей етап визначає основу для успішної реалізації проекту, враховуючи потреби та очікування всіх учасників процесу.

2.2.1. Ідентифікація зацікавлених сторін

Зацікавлені сторони в проекті - це особи, групи чи організації, які мають безпосередній інтерес у проекті або можуть бути впливовими на його результати. Вони можуть включати замовника, який ініціює проект; користувачів, які будуть використовувати продукт; команду проекту, яка реалізує завдання; фінансові сторони, які надають фінансову підтримку; менеджмент організації, який визначає стратегічні рішення; співпрацюючі організації, співтовариства, громадські організації та регулюючі органи, які можуть впливати на різні аспекти проекту. Розуміння потреб і очікувань зацікавлених сторін є важливим для успішного управління проектом, оскільки це дозволяє враховувати різноманітність інтересів та максимізувати підтримку всіх сторін у процесі виконання завдань [5].

Нижче наведені зацікавлені сторони даного проекту:

- *Замовники (Клієнти):* Організації або особи, які ініціюють та фінансують проект, зацікавлені в отриманні продукту, який відповідає їхнім потребам та очікуванням. Вони прагнуть високої якості, вчасної поставки та дотримання бюджету.
- *Кінцеві Користувачі:* Особи чи групи, які будуть використовувати продукт, зацікавлені в його функціональності, зручності використання та відповідності їхнім потребам. Вони оцінюють інтерфейс, продуктивність та якість взагалі.
- *Проектний Менеджер:* Особа, відповідальна за планування, виконання та успішне завершення проекту. Зацікавлений у виконанні завдань в межах бюджету та графіку, ефективній комунікації та розв'язанні конфліктів.
- *Розробники:* Інженери, які відповідають за технічну реалізацію продукту. Зацікавлені в ефективному використанні технічних ресурсів, написанні якісного коду та відповідності продукту вимогам.
- *Тестувальники:* Спеціалісти, які проводять тестування продукту для виявлення помилок та забезпечення його якості. Зацікавлені в високій надійності та безпеці продукту.
- *Фінансові Аналітики:* Особи, які вивчають та аналізують фінансові аспекти проекту. Зацікавлені в ефективному використанні бюджету, оцінці ризиків та забезпеченні фінансової стабільності.
- *Маркетологи:* Особи, відповідальні за позиціонування продукту на ринку та просування його серед потенційних користувачів. Зацікавлені в успішному впровадженні продукту та його конкурентоспроможності.

Врахування глибини інтересів кожної зацікавленої сторони допомагає створити продукт, який задовольняє різноманітні потреби та очікування всіх учасників проекту.

2.2.2. Збір функціональних та нефункціональних вимог

Під час зустрічей із зацікавленими сторонами проекту проводився ретельний аналіз та обговорення функціональних та нефункціональних вимог до майбутнього додатку для торгівлі криптовалютою на платформі Android. Запитання та обговорення на зустрічах допомогли визначити ключові аспекти, які мають бути враховані під час розробки.

Функціональні вимоги:

1. *Реєстрація та Авторизація:* Забезпечення безпечного та зручного входу для користувачів.
2. *Перегляд Балансу та Історії Транзакцій:* Можливість відстеження стану рахунку та перегляд історії усіх фінансових операцій.
3. *Торгівельний Функціонал:* Можливість виконання купівлі та продажу криптовалют за актуальними курсами.
4. *Аналітика та Звітність:* Забезпечення користувачам інструментів аналізу та генерації звітів.

Нефункціональні вимоги:

1. *Безпека та Конфіденційність:* Забезпечення високого рівня захисту особистих даних та транзакцій користувачів.
2. *Швидкодія та Висока Доступність:* Забезпечення ефективності та стабільної роботи додатку без затримок.
3. *Крос-платформенність:* Можливість використання додатку на різних пристроях з операційною системою Android.
4. *Інтуїтивний Інтерфейс:* Забезпечення зручного та легкого користування інтерфейсом додатку для широкого кола користувачів.

Цей процес гарантує повноту та точність визначених вимог, створюючи основу для формулювання чітких завдань для команди розробників. Він також сприяє уникненню непорозумінь та забезпечує взаєморозуміння між усіма учасниками проекту, що сприяє ефективній реалізації завдань та досягненню успішних результатів. Цей важливий етап дозволяє детально визначити

очікування замовника, узгоджувати їх із можливостями та реаліями проекту та визначати пріоритети для успішної реалізації [51].

2.3. Постановка задачі, формулювання технічного завдання на розробку у вигляді паспорту проекту

Задача проекту полягає в розробці андроїд додатку для торгівлі криптовалютою, який надає зручний та надійний спосіб купувати, продавати і обмінювати різні криптовалюти на платформі Android. Цей додаток має на меті вирішити потреби активних учасників криптовалютного ринку, надаючи їм інтуїтивно зрозумілий інтерфейс, а також широкий функціонал для зручного управління їхніми фінансовими операціями. Користувачі зможуть отримати доступ до актуальних курсів криптовалют, здійснювати торгові операції в режимі реального часу та отримувати детальну статистику щодо їхніх інвестицій.

Реєстрація та авторизація користувачів:

- Розробка механізму реєстрації нових користувачів, який включатиме заповнення обов'язкових полів, перевірку правильності введених даних та збереження інформації про користувачів у базі даних.
- Розробка системи авторизації, яка передбачатиме перевірку введеного логіну та пароля, а також використання захисту даних, наприклад, шифрування паролів.

Управління гаманцями:

- Реалізація функції створення нових гаманців для різних криптовалют, яка передбачатиме генерацію унікальних адрес для кожного гаманця.
- Забезпечення можливості додавання коштів до гаманця, переказу та вилучення коштів з гаманця, враховуючи безпекові протоколи та стандарти криптовалют.

Взаємодія з криптовалютними біржами:

- Інтеграція з популярними криптовалютними біржами для отримання актуальної інформації про ціни, статус замовлень та інші ринкові дані.

- Розробка механізму для розміщення заявок на купівлю або продаж криптовалюти на вибраній біржі, з можливістю відстеження та виконання замовлень.

Візуалізація та аналітика:

- Розробка графічного інтерфейсу, який забезпечить зручне візуальне представлення даних про криптовалютні ринки, гаманці та торговельні операції.
- Реалізація аналітичних інструментів, таких як графіки, діаграми, статистика тощо, для детального аналізу динаміки цін, торгових обсягів та інших показників.

Забезпечення безпеки:

- Використання шифрування та захисту даних користувачів.
- Застосування механізмів автентифікації, авторизації та контролю доступу до функцій додатку.
- Впровадження заходів з протидії шахрайству, шпигунству та іншим видам кіберзлочинності.

Тестування та оптимізація:

- Проведення ретельного тестування додатку на різних пристроях, версіях операційних систем та умовах використання.
- Виявлення та усунення помилок, оптимізація роботи додатку для забезпечення швидкості та ефективності його роботи.

Технічне завдання включає в себе детальні вимоги до функціональності, дизайну, безпеки, продуктивності, а також умови та обмеження розробки проекту. Воно визначає основні параметри і характеристики додатку, дозволяючи розробникам реалізувати проект згідно визначених вимог. Технічне завдання також може включати умови і обмеження, пов'язані з інтеграцією з іншими системами, сумісністю з певними версіями операційних систем або пристроїв. Ці аспекти грають важливу роль у забезпеченні успішності розробки та впровадження проекту [20].

Таблиця 2.1. (Додаток Д)

Паспорт проекту

№	Пункт	Опис
1	2	3
1.	Назва проекту	Дослідження характеристик проекту створення додатку для торгівлі криптовалютою на платформі Android
2.	Мета проекту	Метою проекту є розробка та реалізація андроїд додатку, який забезпечує користувачам зручну та безпечну платформу для торгівлі криптовалютою на пристроях з операційною системою Android. Додаток повинен надавати широкий функціонал для торгівлі, забезпечувати надійність, швидкість та конфіденційність операцій, а також забезпечувати користувачів необхідною інформацією про курси криптовалют та тренди ринку.
3.	Основні задачі проекту	<ul style="list-style-type: none"> – Аналіз вимог та визначення функціональності додатку. – Розробка інтерфейсу користувача та реалізація основних функцій. – Інтеграція із зовнішніми API для отримання актуальних даних. – Розробка механізмів безпеки та захисту даних. – Тестування та налагодження додатку. – Впровадження та підтримка додатку.

Продовження табл. 2.1.

1	2	3
4.	Очікувані результати	<ul style="list-style-type: none"> – Функціональний та стабільний андроїд додаток для торгівлі криптовалютою на платформі Android. – Забезпечення безпеки та конфіденційності даних користувачів. – Зручний та інтуїтивно зрозумілий інтерфейс користувача. – Інтеграція із зовнішніми сервісами та API для отримання актуальних даних. – Підтримка різних платіжних методів та криптовалютних бірж. – Повна документація проекту, включаючи детальний опис функціональності та технічних аспектів.
5.	Обмеження проекту	<ul style="list-style-type: none"> – Бюджет проекту обмежений певною сумою. – Потрібно забезпечити сумісність додатка з мобільною платформою Android, що може обмежити вибір технологій або підходів до розробки. <p>Проект має бути завершений протягом 12 місяців.</p>
6.	Ключові зацікавлені сторони	<ul style="list-style-type: none"> – Замовники – Кінцеві користувачі – Криптовалютні біржі – Команда проекту додатку – Фінансові установи, аналітики

Продовження табл. 2.1.

1	2	3
7.	Основні етапи проекту	<ul style="list-style-type: none"> – Необхідно провести аналіз вимог та розробити концептуальну модель протягом 1-2 місяців. – Досягнення дизайну інтерфейсу та структури додатку потребує 2-3 місяців. – Розробка алгоритмів та програмного забезпечення займе від 3 до 6 місяців. – Проведення тестування додатку та внесення коректив займає період від 6 до 8 місяців. – Запуск додатку планується на протязі 8-12 місяців.
8.	Ризики	<ul style="list-style-type: none"> – Технічні ризики – Фінансові ризики – Ризики конкуренції – Правові ризики – Виклики з підтримки та обслуговування
9	Основні метрики успіху	<ul style="list-style-type: none"> – Кількість завантажень додатку – Кількість активних користувачів – Обсяг торгових операцій – Рівень задоволеності користувачів – Час використання додатку – Конверсія використання в операції – Задоволеність замовників

2.4. Розробка концептуальної моделі мобільного додатку

Концептуальна модель мобільного додатку для торгівлі криптовалютою включає визначення основних компонентів додатку, їх взаємодію та

функціональні можливості. Цей етап допомагає визначити структуру додатку та основні функції, необхідні для ефективної торгівлі криптовалютою. Концептуальна модель враховує потреби користувачів та розв'язує проблеми, пов'язані з безпекою, зручністю використання та надійністю торговельного процесу [43].

Основні компоненти мобільного додатку для торгівлі криптовалютою на платформі Android включають:

- *Аутифікація та реєстрація:* Модуль, який дозволяє користувачам створювати акаунти та входити в систему за допомогою своїх облікових даних. Цей модуль забезпечує безпеку та конфіденційність користувачів.
- *Головний екран:* Модуль, який відображає основну інформацію про криптовалютні ринки, таку як актуальні курси, графіки, новини та інші статистичні дані. Користувач може отримувати швидкий доступ до основних функцій додатку з цього екрану.
- *Портфель:* Модуль, який дозволяє користувачам відстежувати свої криптовалютні активи. Він надає можливість додавати та видаляти криптовалюту зі свого портфеля, переглядати їх поточну вартість та отримувати статистику про зміни вартості.
- *Торгівля:* Модуль, що дозволяє користувачам здійснювати купівлю та продаж криптовалют. Він забезпечує доступ до ринкових ордерів, виконання торгових операцій, перегляд історії торгів та інші функції, пов'язані з торгівлею.
- *Оповіщення:* Модуль, який надає користувачам можливість налаштувати сповіщення про зміни вартості криптовалют або інші важливі події на ринку. Це допомагає користувачам вчасно реагувати на зміни та приймати відповідні рішення.
- *Аналітика та звіти:* Модуль, що надає користувачам аналітичні інструменти для вивчення ринку криптовалют та аналізу даних. Він

може включати графіки, діаграми, звіти про торгові операції та інші інструменти для сприяння прийняттю обґрунтованих рішень.

- *Спільнота та соціальні функції*: Модуль, що надає можливість користувачам спілкуватися, обмінюватися думками, коментувати та вподобати записи, створювати обговорення та вступати в групи, пов'язані з криптовалютами.

Кожен з цих модулів має свою внутрішню структуру та взаємодіє з іншими модулями для забезпечення цілісного та зручного досвіду користувача. Концептуальна модель мобільного додатку для торгівлі криптовалютою допомагає визначити, як ці модулі повинні взаємодіяти між собою та як вони повинні відповідати вимогам користувачів [29].

Наприклад, коли користувач використовує модуль перегляду ринкових даних, він може отримати актуальну інформацію про курси криптовалют та графіки їх змін. Користувач може також використовувати модуль купівлі/продажу, де він може здійснювати операції купівлі та продажу криптовалют за актуальними ринковими цінами.

Модуль управління гаманцем дозволяє користувачам зберігати свої криптовалюти та виконувати операції з переказу та отримання коштів. Крім того, додаток може включати модуль аналітики, який надає користувачам звіти та статистику про їх торговельну діяльність та результативність [22].

Окрім основних модулів, концептуальна модель може включати модуль авторизації та безпеки, що забезпечує безпечний доступ до облікового запису користувача, а також модуль сповіщень, який надсилає повідомлення про важливі події та оновлення.

Таким чином, концептуальна модель мобільного додатку для торгівлі криптовалютою визначає основні модулі та їх функціональність, які дозволяють користувачам ефективно торгувати криптовалютою, отримувати актуальну інформацію та керувати своїми фінансовими операціями.

Нижче представлена концептуальна модель мобільного додатку:

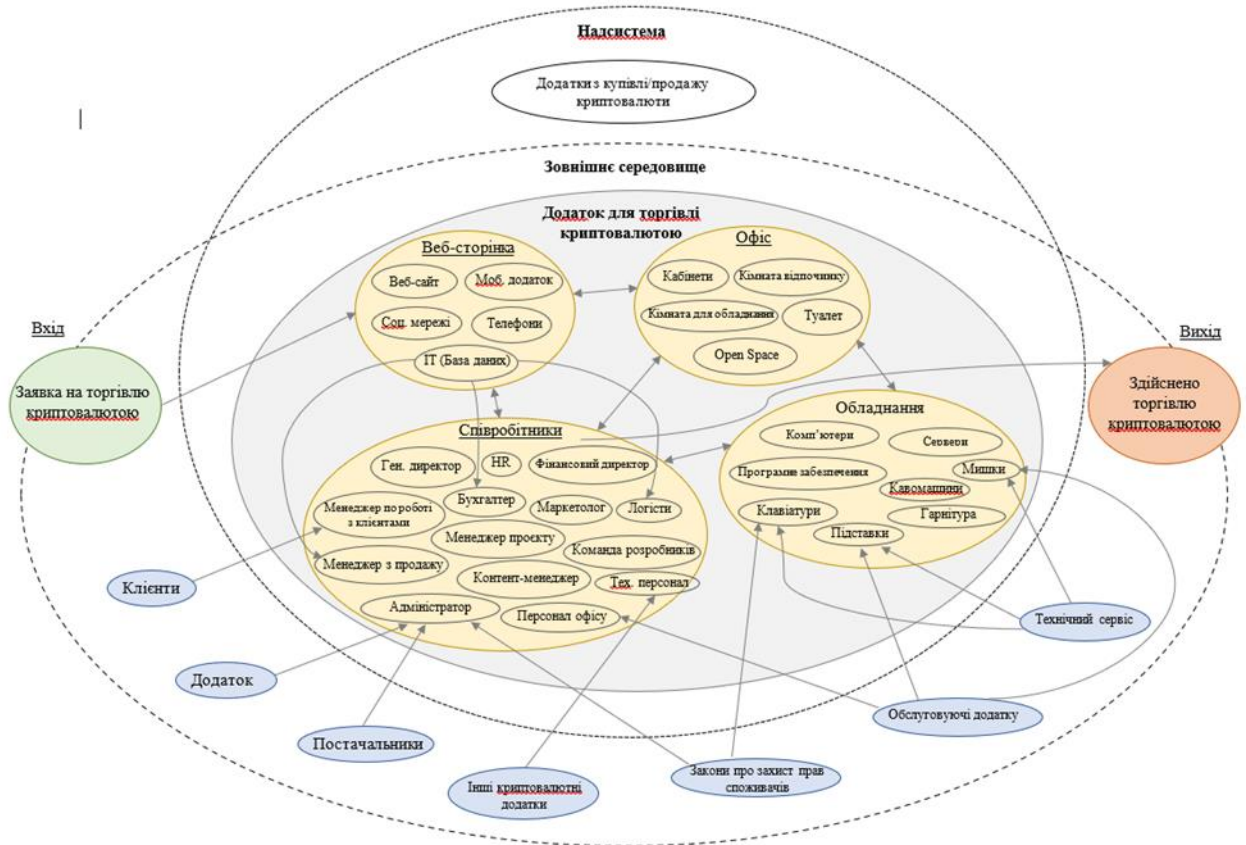


Рис. 2.1. Концептуальна модель мобільного додатку (Додаток E)

2.5. Опис математичної моделі проекту та математична постановка задачі

Задача розробки проекту створення додатку для торгівлі криптовалютою на платформі Android може бути сформульована за допомогою математичних моделей і постановки задачі. Розглянемо детальніше математичну модель та постановку задачі згідно з нашою темою.

Математична модель:

Нехай у нас є певна кількість криптовалют, доступних для торгівлі на платформі Android. Позначимо цю кількість як N . Крім того, маємо M різних торгових платформ, на яких можна здійснювати торгівлю [1].

Для кожної криптовалюти i ($1 \leq i \leq N$) та кожної торгової платформи j ($1 \leq j \leq M$) вводимо змінну x_{ij} , яка вказує, чи включена ця криптовалюта в торговельний портфель на платформі Android. Якщо $x_{ij} = 1$, то криптовалюта i включена до торговельного портфелю на платформі j ; якщо $x_{ij} = 0$, то криптовалюта i не включена до торговельного портфелю на платформі j .

Цільова функція:

Основною метою є максимізація прибутку від торгівлі криптовалютами на платформі Android. Цільова функція може бути виражена наступним чином: $Max Z = \sum_{i=1}^N \sum_{j=1}^M p_{ij} \times x_{ij}$, де p_{ij} - прибуток від торгівлі криптовалютою i на платформі j .

Обмеження:

В процесі розробки додатку для торгівлі криптовалютою, можуть бути деякі обмеження, які потрібно враховувати. Деякі з можливих обмежень включають:

Обмеження на кількість доступних ресурсів:

$\sum_{j=1}^M x_{ij} \times r_{ij} \leq R_j$, для всіх i ($1 \leq i \leq N$), де r_{ij} - ресурси, необхідні для торгівлі криптовалютою i на платформі j , R_j - максимальна кількість ресурсів, доступних для торгівлі криптовалютою i .

Обмеження на ризики:

$\sum_{j=1}^M x_{ij} \times r_{skij} \leq R_{skj}$, для всіх j ($1 \leq i \leq N$), де r_{skij} – ризик торгівлі криптовалютою i на платформі j , R_{skj} – максимальний припустимий ризик для криптовалюти j .

Математична постановка задачі:

Задача полягає в тому, щоб вибрати оптимальний торговельний портфель криптовалют на платформі Android, який максимізує прибуток, враховуючи обмеження на ресурси та ризики. Математична постановка задачі полягає в максимізації цільової функції Z при виконанні всіх обмежень.

Така модель та постановка задачі дозволять розробити додаток для торгівлі криптовалютою на платформі Android, який допоможе ефективно керувати торговельними операціями та досягати максимального прибутку, враховуючи ресурси та ризики.

2.6. Аналіз та вибір методології управління проектом. Класифікація моделей управління ІТ проектами.

Методи управління проектами - це систематичні та структуровані підходи до планування, виконання та контролю проектних завдань для

досягнення поставлених цілей. Їх вибір залежить від характеристик проекту, його складності, розміру та інших факторів [26]. Основні методи включають:

1. *Каскадна модель (Waterfall)*: Каскадна модель - це традиційний та лінійний підхід до управління проектами, де розробка розділяється на послідовні фази. Кожна фаза починається лише після завершення попередньої. Перевагами такого підходу є структурованість та чіткість визначених етапів, що дозволяє краще контролювати процес. Однак, він менш гнучкий у відношенні змін вимог та може бути менш ефективним у сучасному швидкозмінюваному середовищі.
2. *Agile* - це досить гнучка система управління, характерними ознаками якої є надання кінцевого продукту на кожному етапі роботи. Основний принцип роботи – поділ проекту на короткі цикли (ітерації), наприкінці кожного з яких замовник отримує певний продукт. У agile широка сфера застосування, однак найкраще він прижився в ІТ. За Agile етапи можуть не йти послідовно, а відбуватися паралельно або в різному порядку. Ключовий момент – на кожному етапі реалізації проекту його продуктом можна користуватися. Етапи прийнято називати спринтами, кожен з яких містить набір операцій і строки їх реалізації. Строки виконання плануються безпосередньо перед спринтом.
3. *Scrum*: Scrum - це Agile методологія, яка використовує короткі ітерації, називані sprints, для розробки та управління проектом. Вона акцентується на ефективній комунікації в команді та замовнику. Регулярні мітинги (Daily Scrum) сприяють вирішенню проблем, взаємодії та вирішенню питань шляхом спільних обговорень.
4. *Kanban*: Канбан - це методологія, орієнтована на візуалізацію та оптимізацію процесів. Завдання візуалізуються на дошці, де кожен етап роботи представлений колонкою. Це дозволяє команді бачити поточний стан завдань та швидко реагувати на зміни.
5. *Extreme Programming (XP)*: Extreme Programming (XP) - це Agile методологія, яка ставить на увагу короткі цикли розробки, велику

взаємодію та високий ступінь автоматизації тестування. Вона прагне досягти високої якості продукту та швидкої реакції на зміни в вимогах [55].

2.6.1. Обґрунтування вибору моделі управління для проекту розробки

На основі нашої теми проекту, ми розглянули різні підходи до розробки програмного забезпечення і обрали Agile-розробку програмного забезпечення, як найбільш відповідну для проекту. Якщо ми оберемо цей підхід, то зможемо розробити програмне забезпечення ефективно та виконати вимоги проекту.

Agile-розробка програмного забезпечення передбачає ітеративний та інкрементальний підхід до розробки. Це означає, що ми будемо працювати над проектом у невеликих ітераціях або спринтах. Кожен спринт буде фокусуватися на розробці конкретної функціональності або компонента програмного забезпечення [24].

Використання Agile-розробки програмного забезпечення допоможе нам забезпечити якісну розробку програмного забезпечення, зменшити ризики та забезпечити задоволення потреб замовника [31].

Основні етапи Agile-розробки програмного забезпечення включають:

- *Планування:* Збір вимог та формулювання коротких та конкретних цілей для кожного ітераційного циклу розробки.
- *Розробка ітерацій:* Розробка програмного забезпечення відбувається на протязі коротких ітерацій (зазвичай 1-4 тижні), під час яких розробники фокусуються на конкретних функціональних можливостях або компонентах системи.
- *Постійний зворотній зв'язок:* Регулярні зустрічі між командою розробників та замовником для обговорення прогресу, зворотного зв'язку та уточнення вимог.
- *Тестування та оцінка якості:* Виконання тестування програмного забезпечення на кожному етапі розробки з метою виявлення та виправлення помилок.

- *Інкрементальна поставка*: Готові функціональні частини програмного забезпечення постійно впроваджуються та надаються замовнику, що дозволяє отримувати швидкий фідбек та забезпечує більш ефективне управління ризиками.
- *Постійне вдосконалення*: Аналіз роботи команди розробників, виявлення недоліків та можливостей для покращення процесу розробки програмного забезпечення [18].

Переваги Agile-розробки програмного забезпечення включають:

- *Гнучкість*: Здатність швидко адаптуватися до змінних вимог і впроваджувати нові функціональності на кожному етапі розробки.
- *Залученість замовника*: Замовник має можливість активно співпрацювати з командою розробників, давати зворотний зв'язок та контролювати процес розробки.
- *Швидкість впровадження*: Завдяки ітеративному підходу нові функціональності можуть бути впроваджені швидше, що дозволяє отримувати швидкий фідбек від замовника та коригувати напрямок розробки.
- *Покращена якість*: Регулярне тестування та оцінка якості допомагають виявляти та виправляти помилки на ранніх етапах розробки.

Цей підхід дозволить нам швидко відгукуватися на зміни і впроваджувати зміни у програмному забезпеченні на ранніх етапах проекту. Ми зможемо працювати в тісному співробітництві із замовником, отримувати його зворотний зв'язок та вносити корективи до розробки в процесі [44].

Також, Agile-розробка передбачає гнучкість і адаптивність, тому ми зможемо пристосовувати наш підхід до нових вимог і викликів, що можуть виникнути під час розробки. Це дозволить нам бути більш готовими до змін і забезпечити успішну реалізацію проекту [10].

2.7. Команда та кошторис проекту

2.7.1. Організаційна структура команди проекту

Людські ресурси відіграють важливу роль у успіху будь-якого проекту чи підприємства. Вони забезпечують доступ до різноманітних фахівців, визначають ефективність комунікації та взаємодії в команді. Здатність мотивувати персонал та забезпечувати ефективне лідерство впливає на результативність проекту.

Людські ресурси відповідають за управління та планування ресурсами, ефективне розподілення завдань і часу для досягнення цілей. Креативність та інновації, що притаманні залученим співробітникам, допомагають виробити нові ідеї та підходи.

Також, вони грають ключову роль у вирішенні конфліктів, створенні сприятливого робочого середовища та забезпеченні загального благополуччя працівників. Управління людськими ресурсами визначає успіх та стабільність проекту в довгостроковій перспективі [41].

Для успішної реалізації проекту необхідна компетентна команда, яка включає в себе:

1. Проектний менеджер - керує всім проектом, включаючи планування, виконання та контроль за термінами та бюджетом. Також він організовує комунікацію в команді та вирішує конфлікти.
2. Бізнес-аналітик - збирає та аналізує вимоги зацікавлених сторін та клієнтів, розробляє стратегії вдосконалення бізнес-процесів і дає рекомендації щодо функціональності продукту.
3. Дизайнер інтерфейсу - відповідає за створення естетичного та функціонального дизайну продукту та розробляє інтерфейс для зручності користувачів.
4. Розробник Android - пише програмний код для мобільного додатку на платформі Android і забезпечує виконання вимог щодо функціональності та продуктивності.

5. Бекенд-розробник - відповідає за розробку та підтримку серверної частини додатку та забезпечує взаємодію між мобільним додатком та сервером.
6. Інженер з баз даних - розробляє та підтримує базу даних для зберігання інформації та оптимізує запити для ефективного доступу до даних.
7. Тестувальник - виконує різноманітні тести для перевірки функціональності та виявлення помилок, а також допомагає виправляти їх та оптимізувати продукт.
8. Продакт менеджер - визначає стратегію та функціональність продукту, а також керує його життєвим циклом та взаємодіє з розробницькою командою.
9. Маркетолог - розробляє маркетингові стратегії для просування продукту та вивчає конкурентне середовище, визначаючи цільову аудиторію.

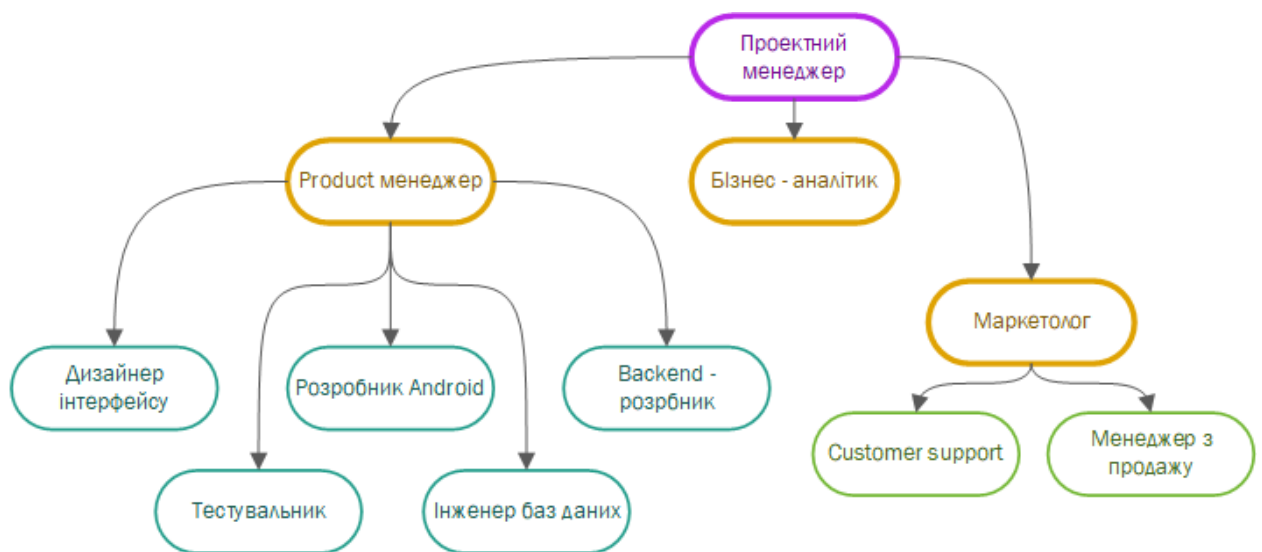


Рис. 2.2. Організаційна структура команди проекту (Додаток Ж)

2.7.2. Кошторис проекту

Матеріальні ресурси включають у себе фізичні об'єкти та матеріали, які використовуються для реалізації проекту. Це може охоплювати такі аспекти, як обладнання, технології, інфраструктура та інше. Наприклад, комп'ютери, сервери, офісне обладнання, програмне забезпечення, витрати на інтернет-з'єднання, а також будь-які інші матеріальні активи, необхідні для виконання завдань проекту. Ефективне управління матеріальними ресурсами важливо

для забезпечення ефективності та успішності проекту [48]. Важливими матеріальними ресурсами є:

1. Інтернет-з'єднання - для забезпечення безперебійного доступу до мережі, обміну інформацією та спілкування в команді.
2. Програмне забезпечення - необхідне для розробки, тестування та впровадження програмного продукту.
3. Серверне обладнання - якщо необхідно забезпечити роботу великої кількості користувачів або використовувати серверні послуги.
4. Офіс, офісні прилади та засоби комфорту - для підтримки робочого середовища та комфорту команди.
5. Засоби зв'язку - телефони, відеоконференційні системи для ефективної комунікації між учасниками проекту.
6. Спеціальне обладнання - якщо для реалізації проекту необхідне спеціалізоване або індивідуальне обладнання.
7. Робочі меблі та аксесуари - для забезпечення комфортної робочої обстановки.
8. Резервні матеріали та компоненти - для уникнення перебоїв у виробництві в разі виникнення непередбачуваних ситуацій.

Бюджет проекту - це фінансовий план, що визначає усі витрати, пов'язані з реалізацією конкретного проекту. Його створення є важливою частиною проектного управління і включає в себе оцінку та аналіз різних фінансових аспектів. У процесі розробки можуть використовуватися різні методи оцінки вартості, а також інструменти та техніки управління бюджетом проекту. Бюджет розробляється з урахуванням усіх наступних аспектів і повинен бути оптимізованим для ефективного використання ресурсів та досягнення поставлених цілей [15]:

1. Оцінка обладнання та інфраструктури: Визначення витрат, пов'язаних із закупівлею обладнання, необхідного для реалізації проекту, а також витрат на інфраструктуру, наприклад, сервери, комп'ютери, програмне забезпечення та інше.

2. Витрати на навчання та розвиток персоналу: Забезпечення коштів для навчання та розвитку команди, включаючи тренінги, семінари та інші освітні заходи.
3. Витрати на логістику та транспорт: Визначення бюджету для транспортних потреб, включаючи витрати на доставку обладнання, відрядження команди та інші логістичні витрати.
4. Витрати на інновації та дослідження: Забезпечення коштів для впровадження новітніх технологій та проведення досліджень для підтримки інновацій у проекті.
5. Витрати на правову підтримку: Ресурси для оплати юридичних послуг, включаючи укладання угод, реєстрацію прав інтелектуальної власності та інші юридичні аспекти.
6. Витрати на управління ризиками: Забезпечення резервів для управління ризиками, включаючи можливі збитки або несподівані обставини.

Загальний бюджет проекту розробляється з урахуванням усіх вищезазначених аспектів та повинен бути оптимізованим для ефективного використання ресурсів та досягнення поставлених цілей.

Таблиця 2.2. (Додаток II)

Бюджет проекту

Посада	Ціна (\$/міс)	Тривалість (міс)	Загальні витрати (\$)
1	2	3	4
Проектний менеджер	\$5000	12	60000\$
Бізнес-аналітик	\$4000	12	48000\$
Дизайнер інтерфейсу	\$3500	6	21000\$
Розробник Android	\$4500	12	54000\$
Бекенд-розробник	\$4500	12	54000\$

Продовження табл 2.2.

1	2	3	4
Інженер з баз даних	\$4000	12	48000\$
Тестувальник	\$3500	10	35000\$
Продакт менеджер	5000\$	12	60000\$
Маркетолог	4000\$	8	32000\$
Додаткові витрати			
Оренда офісного простору	2000\$	12	24000\$
Обладнання та програмне забезпечення	1050\$	12	12600\$
Реклама і маркетинг	800\$	12	9600\$
Логістика та транспорт	500\$	12	6000\$
Витрати на управління ризиками	5000\$	12	60000\$
Правова підтримка	1000\$	12	12000\$
Інновації та дослідження	1000\$	10	10000\$
Загальні витрати			557000\$

Загалом вартість трудових ресурсів складає: 412 000 \$. Нижче наведена кругова діаграма розподілу вартості трудових ресурсів проекту розробки Android-додатку для торгівлі криптовалютою.



Рис. 2.3. Діаграма вартості трудових ресурсів (Додаток К)

2.8. Управління змістом проекту. WBS і роботи проекту

WBS, або структура розбиття робіт (Work Breakdown Structure), визначається як інструмент управління проектами, який сприяє розкладенню проекту на менші більш керовані компоненти. Це ієрархічна структура, яка включає в себе всі роботи, необхідні для досягнення цілей проекту.

WBS вирізняється тим, що дозволяє ефективно організувати та контролювати весь проект, а також визначати залежності між окремими роботами та розподіляти їх між командою проекту. Кожен рівень ієрархії WBS представляє собою все більш деталізовані етапи чи елементи проекту, що дозволяє керівникам та членам команди легше розуміти та виконувати свої завдання.

WBS виявляється корисним інструментом для визначення обсягу проекту, планування ресурсів, встановлення контрольних точок та

забезпечення здійснення проекту в рамках чітко визначених меж. Цей метод сприяє збільшенню ефективності управління та зменшенню ризиків, пов'язаних із реалізацією проектних завдань [30].

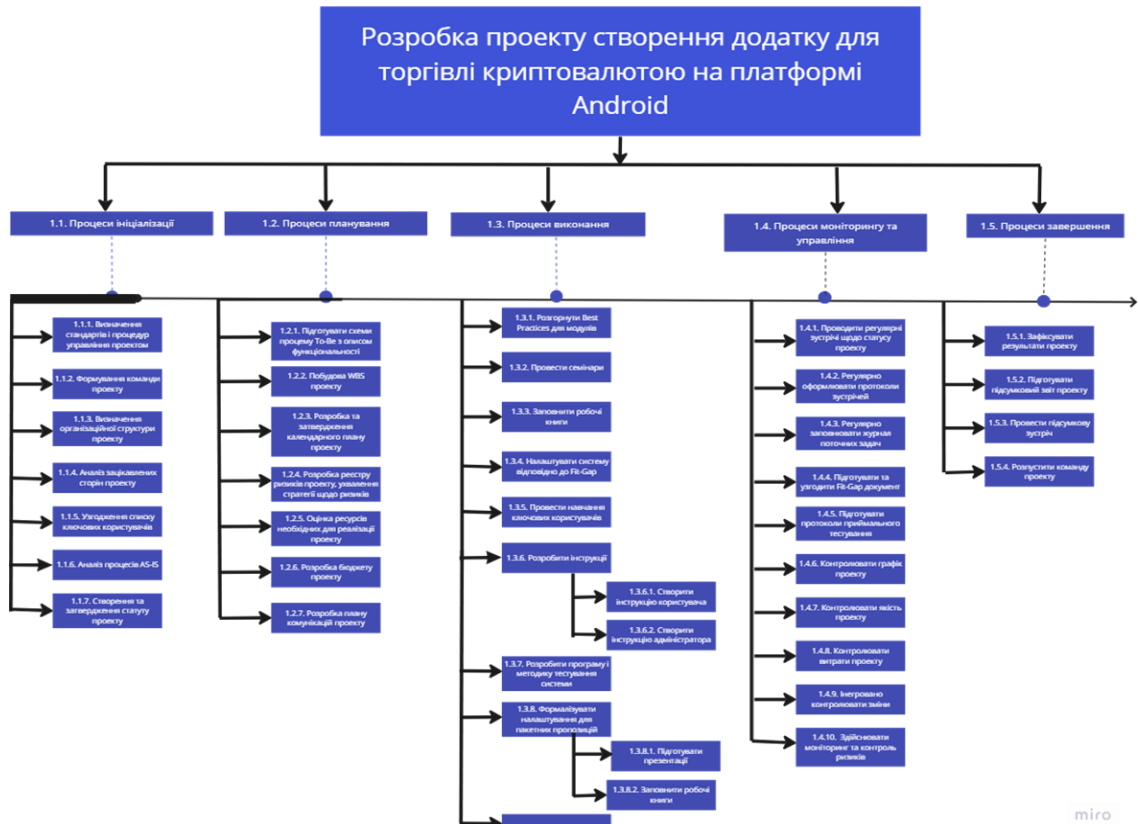


Рис. 2.4. WBS проекту по процесам (Додаток Л)

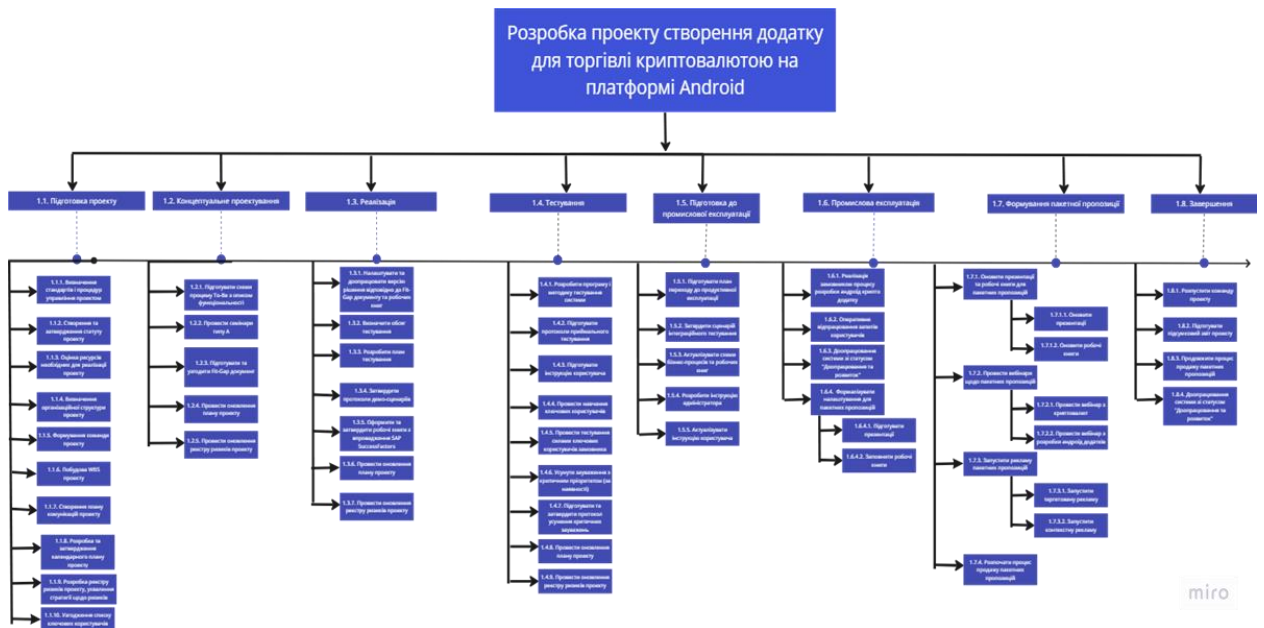


Рис. 2.5. WBS проекту по фазам життєвого циклу (Додаток М)



Рис. 2.6. WBS проекту по продуктам (Додаток Н)

2.9. Управління часом проекту. Діаграма Ганта.

Для ефективного виконання проекту необхідно створити докладний графік, що включає в себе:

- *Визначення термінів та етапів реалізації:* Розподіл завдань на конкретні етапи з визначеними термінами виконання. Це включає ретельне планування, визначення пріоритетів та встановлення критичних завдань.
- *Вибір моделі управління проектами:* Аналіз різних моделей управління проектами та вибір тієї, яка найкраще відповідає конкретним потребам та характеру проекту.

- *Створення графіка*: Розробка детального календарного плану з урахуванням завдань, термінів та залежностей. Визначення можливостей оптимізації графіка для забезпечення ефективного виконання завдань.
- *Графік проведення мітингів*: Організація регулярних зустрічей для обговорення прогресу, вирішення питань та планування наступних кроків. Створення чіткого плану мітингів із зазначенням цілей та відповідальностей.
- *Контроль якості*: Систематичний моніторинг та перевірка відповідності вимогам з метою забезпечення високої якості виконання завдань на всіх етапах проекту.
- *Контроль виконання задач*: Регулярне відстеження прогресу та виявлення можливих затримок. Вжиття заходів для вирішення труднощів та підтримки виконання завдань у визначені строки.

Цей підхід є ключовим для забезпечення успішності проекту. Детальне визначення термінів та етапів реалізації дозволяє команді чітко розуміти, які кроки потрібно виконати і в які строки. Вибір оптимальної моделі управління проектами забезпечує ефективну координацію та взаємодію всіх учасників.

Створення докладного графіка із визначенням завдань, термінів та залежностей сприяє плануванню ресурсів та оптимізації робочого процесу. Регулярні мітинги дозволяють оперативно вирішувати питання, вислуховувати ідеї команди та вносити корективи до плану, що сприяє адаптації до змін у середовищі проекту.

Контроль якості та виконання завдань забезпечує високу якість виробленого продукту. Систематичний моніторинг прогресу та оперативна реакція на затримки дозволяють уникнути можливих проблем та забезпечити вчасне завершення проекту. Такий комплексний підхід гарантує досягнення поставлених цілей та задоволення вимог усіх учасників проекту [35].

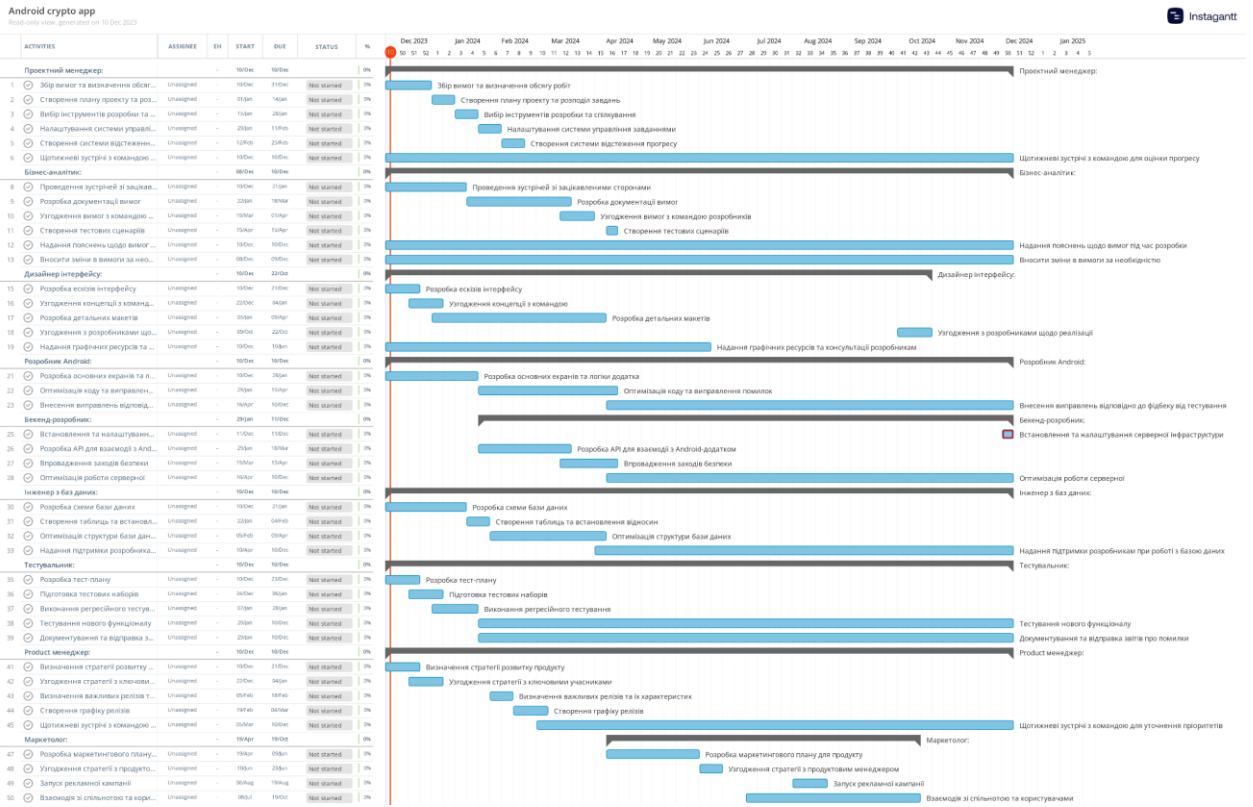


Рис. 2.7. Діаграма Ганта (Додаток П)

2.9.1. Календар мітингів

Календар мітингів охоплює різні види зустрічей, кожен з яких виконує певні функції у процесі управління проектом. До звичайних видів мітингів відносяться:

1. Щоденні мітинги (Daily Standups або Daily Scrums): Мають на меті швидкий облік прогресу та планів на день. Вони проводяться щоденно.
2. Спринт-планування (Sprint Planning): Спрямовані на планування завдань на наступний спринт. Ці мітинги відбуваються перед кожним спринтом.
3. Спринт-ретроспектива (Sprint Retrospective): Служать для огляду попереднього спринту, виділення позитивних та негативних аспектів та пошуку шляхів вдосконалення. Їх проводять в кінці кожного спринту.
4. Мітинги із замовником (Client Meeting або Stakeholder Meeting): Призначені для обговорення вимог та очікувань замовника чи зацікавлених сторін. Ці мітинги проводяться за необхідності.
5. Планування релізів (Release Planning): Спрямовані на визначення стратегії випуску продукту та розподіл завдань на кілька спринтів.

Зазвичай вони відбуваються під час початкового етапу проекту та перед випуском нової версії.

- б. Мітинги з вирішення проблем (Problem-Solving Meetings): Призначені для виявлення та вирішення проблем, що можуть виникнути в ході проекту. Ці мітинги проводяться при потребі.

Всі ці мітинги взаємодіють для забезпечення ефективного спілкування, вирішення проблем та планування дій для досягнення поставлених цілей [53].

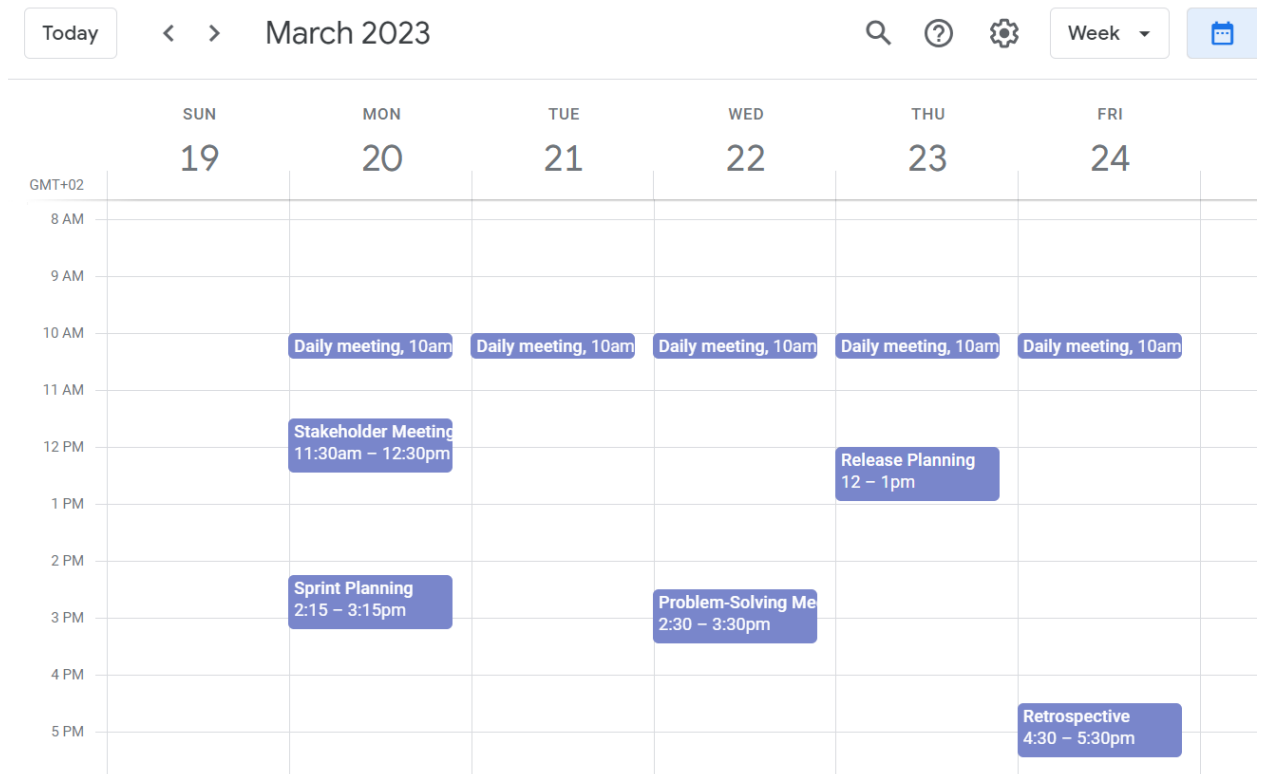


Рис. 2.8. Приклад графіку проведення мітингів (Додаток Р)

2.10. Управління ризиками проекту

Ризики - це потенційні події або обставини, які можуть мати негативний вплив на досягнення цілей проекту чи організації. У контексті інформаційних технологій (ІТ) проектів, ризики можуть включати технічні проблеми, витрати, зміни в вимогах, проблеми безпеки, конфлікти в команді та інші фактори, які можуть стати перешкодою успішному виконанню проекту. Управління ризиками передбачає ідентифікацію, оцінку та прийняття заходів для зменшення чи контролювання можливих негативних наслідків.

Ризики у проекті можуть мати різні властивості та виникати на різних етапах проектного циклу. Важливо ідентифікувати, аналізувати та вирішувати ці ризики для забезпечення успішного виконання проекту [12].

Властивості ризиків:

1. Ймовірність виникнення: Оцінка ймовірності того, що конкретний ризик стане реальністю. Висока ймовірність може вказувати на необхідність докладного вивчення та управління цим ризиком.
2. Величина впливу: Оцінка масштабу впливу ризику на проект. Це може включати фінансові втрати, затримки у графіку, втрату ресурсів тощо.
3. Ступінь контролю: Оцінка того, наскільки ефективно можна контролювати або уникнути конкретний ризик. Деякі ризики можуть бути керовані краще, ніж інші.

Різновиди ризиків:

1. Внутрішні ризики: Ті, що пов'язані зі специфікою самого проекту, наприклад, технічні обмеження, непрогнозовані труднощі в розробці або зміни вимог.
2. Зовнішні ризики: Походять від зовнішніх факторів, таких як зміни в законодавстві, економічні коливання або тиск з боку конкурентів.
3. Ризики людського фактору: Виникають через недостатню кваліфікацію, конфлікти в команді, втрату ключового персоналу або зміни в команді.
4. Технічні ризики: Зумовлені технічними складнощами, вибором несправжніх технологій, або непрогнозованими аспектами в розробці.
5. Фінансові ризики: Включають в себе можливі фінансові втрати або витрати, неспроможність залучити достатньо коштів для завершення проекту.

Етапи управління ризиками:

1. Ідентифікація ризиків: Аналіз проекту для визначення можливих ризиків. Використання методів, таких як SWOT-аналіз, допомагає виявити потенційні проблеми.

2. Аналіз ризиків: Оцінка ймовірності та величини впливу кожного ризику. Спроба розуміти, як ризики можуть вплинути на проект.
3. Планування відповіді на ризики: Розроблення стратегій для управління ризиками, таких як уникнення, перенесення, зменшення чи прийняття ризику.
4. Втілення стратегій: Впровадження запланованих стратегій для управління ризиками та вирішення проблем, які вони можуть викликати.
5. Моніторинг і контроль ризиків: Систематичне відстеження ризиків протягом усього проекту та коригування стратегій, якщо ситуація змінюється [39].

№	Тип ризику	Ризикова подія	Сила впливу	Керованість
1	Програмні ризики	Виявлення критичного програмного багу	Висока	Середня
2		Неправильна інтеграція з API	Висока	Середня
3		Зміни в вимогах користувача	Середня	Низька
4		Затримка у випуску нової версії ОС	Низька	Низька
5	Апаратні ризики	Виход з ладу апаратного забезпечення	Висока	Висока
6		Недоступність необхідного обладнання	Висока	Середня
7		Втрата даних через дефект жорсткого диску	Середня	Середня
8		Проблеми зі стабільністю живлення	Висока	Низька
9	Внутрішні ризики проекту (команда)	Відсутність ключового члена команди	Висока	Середня
10		Порушення графіка роботи команди	Середня	Середня
11		Конфлікти в команді	Висока	Висока
12		Недостатня кваліфікація персоналу	Середня	Висока
13	Зовнішні (оточення)	Зміни в законодавстві про криптовалюту	Низька	Низька
14		Конкуренція на ринку криптовалют	Середня	Низька
15		Зміни у валютному ринку	Висока	Низька
16		Втрата доступу до платформи Google Play	Висока	Низька
17	Форс мажори	Природні катастрофи	Висока	Низька
18		Політичні кризи	Висока	Низька
19		Техногенні аварії	Висока	Низька
20		Збої в роботі інфраструктури	Середня	Низька
21	Ризики кібербезпеки	Вторгнення хакерів	Висока	Низька
22		Витік конфіденційної інформації	Висока	Низька
23		Спроби впливу на роботу додатку зловмисників	Висока	Низька
24		Недостатній захист даних користувачів	Середня	Низька

Рис. 2.9. Ідентифікація ризиків проекту (Додаток С)

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовірність		Частота (за проєкт)		Важливість ризиків (компл. показ ник)
		Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.
		1	Ризик подорожчання ліцензійного ПЗ	СВ	6	ВС	8	СН	4	СН
2	Ризик втрати ключового члена команди	СВ	7	ВС	8	СН	3	СН	3	24
3	Ризик зміни вимог до додатку	СС	5	ВС	7	СН	2	СН	2	14
4	Ризик недостатньої кваліфікації персоналу	СН	4	СС	6	СН	3	СС	3	18
5	Ризик виходу з ладу сервера	СН	8	СС	7	СС	2	СС	2	14
6	Ризик змін у законодавстві про криптовалюту	СС	6	СВ	7	СН	1	СН	1	7
7	Ризик конфліктів в команді	СВ	7	ВС	8	СВ	2	СВ	2	16
8	Ризик змін на фінансовому ринку	СС	6	СВ	7	СВ	1	СВ	1	7
9	Ризик неправильної інтеграції з API	СВ	7	СС	6	СС	3	СС	3	18
10	Ризик природної катастрофи	ВВ	9	СН	5	СН	1	СН	1	5
11	Ризик збоїв в роботі інфраструктури	СС	8	СС	7	СС	2	СС	2	14
12	Ризик вторгнення хакерів	СВ	8	ВВ	9	СС	3	СС	3	27
13	Ризик витоку конфіденційної інформації	СС	7	ВВ	8	СВ	2	СС	2	16
14	Ризик спроб впливу на роботу додатку зловмисників	ВВ	8	ВВ	9	СВ	3	СС	3	27
15	Ризик політичної кризи	СВ	6	СН	5	ВС	1	СН	1	5
16	Ризик змін у валютному ринку	СС	7	СС	6	СС	2	СС	2	12
17	Ризик втрати доступу до платформи Google Play	СВ	7	ВВ	8	СС	3	СВ	3	24
18	Ризик техногенної аварії	СС	8	СС	7	СН	3	СС	3	21
19	Ризик недоступності необхідного обладнання	СН	5	СН	6	СН	1	СН	1	6
20	Ризик виходу з ладу апаратного забезпечення	СН	7	СН	8	СН	2	СН	2	16

Рис. 2.10. Оцінювання ризиків проєкту (Додаток Т)

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		профілактика		при симптомі	при проблемі
1	Ризик подорожчання ліцензійного ПЗ	Перевірка альтернативного програмного забезпечення та ліцензій на ринку	Збір інформації про доступні альтернативи	Розгляд можливості використання безкоштовного ПЗ	Спілкування з постачальником ліцензійного ПЗ
2	Ризик втрати ключового члена команди	Пошук і найм резервного персоналу	Моніторинг роботи членів команди	Проведення тренінгів та розвиток кадрів	Регулярні зустрічі з ключовим членом команди
3	Ризик зміни вимог до додатку	Постійний моніторинг вимог та оновлення документації з вимог	Аналіз вимог та змін до вимог	Регулярне оновлення документації з вимог	Заплановане оновлення функціональної специфікації
4	Ризик недостатньої кваліфікації персоналу	Планування та проведення навчання та розвитку персоналу	Визначення слабких місць у кваліфікації персоналу	Надання навчальних матеріалів і засобів для самостійного навчання	Регулярні оцінки кваліфікації персоналу

Рис. 2.11. Розробка протиризикових заходів (ПРЗ) проєкту (Додаток У)

Управління ризиками є важливою складовою успішного виконання проєктів, оскільки воно дозволяє проактивно взаємодіяти з невизначеністю та забезпечити ефективну реакцію на можливі труднощі.

РОЗДІЛ 3. ЕТАП ПРОЕКТУВАННЯ, РОЗРОБКИ, ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ ДОДАТКУ

Проектування - це ключовий етап у розробці проектів, який передує фактичній реалізації. Під час цього процесу фахівці з дизайну та розробки працюють над створенням концепцій, архітектури та інтерфейсів продукту. Метою проектування є не лише створення естетично привабливого дизайну, але й врахування функціональних, ергономічних та інших важливих аспектів.

Проектування включає в себе аналіз потреб користувачів, визначення функціональності та розробку структури продукту. Дизайнери розробляють концепції інтерфейсів, визначають колірну палітру, шрифти та інші елементи візуального оформлення.

Прототипування є важливим етапом, що передує розробці фінальної версії продукту. Під час прототипування створюються моделі або прототипи продукту, які дозволяють оцінювати його функціональність та дизайн. Це допомагає замовнику та команді проекту отримати відчуття від інтерфейсу та функцій, вносити корективи та вдосконалювати продукт перед фінальною реалізацією.

Загалом, проектування та прототипування є процесами, спрямованими на те, щоб забезпечити успішний старт розробки, враховуючи як вимоги користувачів, так і технічні аспекти [4].

3.1. Вибір технологій та архітектури системи

Архітектура системи - це стратегічно розроблена організація та структура програмного чи апаратного забезпечення, яка визначає взаємодію між її різними компонентами. Це важливий аспект у розробці програмних продуктів та систем, оскільки правильно обрана архітектура може визначити ефективність, масштабованість та стабільність системи.

Архітектура системи визначає, як компоненти системи взаємодіють між собою, як вони організовані та як вони виконують свої функції в рамках системи. Вона включає в себе визначення структури додатку, вибір

технологій, патерни проектування та інші аспекти, що визначають архітектурний ландшафт системи.

Вибір архітектури системи — ключовий етап у розробці програмного продукту. Від цього вибору залежать продуктивність, масштабованість та безпека системи. Ретельне визначення структури системи визначає, як вона буде взаємодіяти з користувачами та іншими системами, як буде забезпечено розподілена обробка даних та які можливості для масштабування та розширення функціоналу будуть доступні.

Один із ключових моментів при виборі архітектури - це забезпечення відповідності функціональним та нефункціональним вимогам проекту. Різні типи архітектур (наприклад, клієнт-серверна, мікросервісна, багатошарова) можуть відповідати різним потребам проекту в залежності від його характеру та масштабу [54].

Отже, архітектура системи є фундаментальним елементом у процесі проектування програмних та апаратних систем, впливаючи на їх ефективність та здатність відповідати вимогам користувачів і бізнесу.

Існує кілька основних типів архітектур програмних систем, кожен з яких має свої унікальні характеристики та переваги. Ось деякі з них:

- Монолітна архітектура характеризується наявністю одного великого програмного забезпечення, в якому об'єднано весь функціонал. Це спрощує розробку та тестування, зокрема для невеликих проектів.
- Клієнт-Серверна архітектура передбачає розділення функціоналу між клієнтською (frontend) та серверною (backend) частинами. Це сприяє покращеній масштабованості та полегшує управління та підтримку.
- Мікросервісна архітектура полягає в розділенні функціоналу на невеликі, автономні мікросервіси. Це забезпечує гнучкість, масштабованість та можливість використання різних технологій для кожного сервісу.
- Багатошарова (N-Tier) архітектура передбачає розділення функціоналу на різні рівні або шари, такі як представлення, логіка додатку та база

даних. Це забезпечує модульність, полегшує підтримку та масштабованість.

- Схема Ядро-Плагін (Plugin Architecture) включає в себе основне ядро програми з можливістю додавання функціоналу за допомогою плагінів. Це забезпечує розширюваність та гнучкість, дозволяючи додавати та вилучати функціонал без перезапуску основної системи.

При визначенні оптимальної архітектури системи для розробки додатку для торгівлі криптовалютою на платформі Android були враховані ключові аспекти та вимоги проекту:

1. *Модульність*: Вибір модульної архітектури, яка дозволяє розбити систему на незалежні модулі, що спрощує розробку, тестування та збереження коду.
2. *Масштабованість*: Розгляд можливостей горизонтальної та вертикальної масштабованості для забезпечення ефективності та здатності системи впоратися з ростом обсягів даних та користувацького трафіку.
3. *Безпека*: Впровадження засобів захисту даних та транзакцій, зокрема використання шифрування та механізмів аутентифікації.
4. *Сумісність*: Врахування сумісності із різними версіями операційної системи Android та різними пристроями для максимального охоплення аудиторії користувачів.
5. *Відкритість для розширень*: Створення гнучкої архітектури, що дозволяє легко внести зміни та розширити функціонал додатку у майбутньому [32].

Обрана для проекту архітектура Model-View-Controller (MVC) є популярним підходом до розробки програмного забезпечення, включаючи мобільні додатки для Android. В рамках цієї архітектури додаток розбивається на три ключові компоненти:

1. *Модель (Model)*: Відповідає за управління даними та бізнес-логікою. Модель представляє собою об'єкти даних та функції, які взаємодіють із зовнішнім світом.
2. *Вид (View)*: Представляє інтерфейс користувача і відповідає за відображення інформації користувачу. Він отримує дані від моделі та відображає їх у вигляді, зрозумілому для користувача.
3. *Контролер (Controller)*: Обробляє введення користувача та керує взаємодією між моделлю та виглядом. Контролер приймає вхідні дані від користувача, взаємодіє з моделлю для оновлення даних та сповіщає вид про зміни.

Однією з ключових переваг MVC є розділення відповідальностей між цими компонентами, що спрощує управління кодом та робить додаток більш гнучким. Це полегшує розширення функціоналу, тестування окремих компонентів та сприяє підтримці додатку на різних етапах розробки.

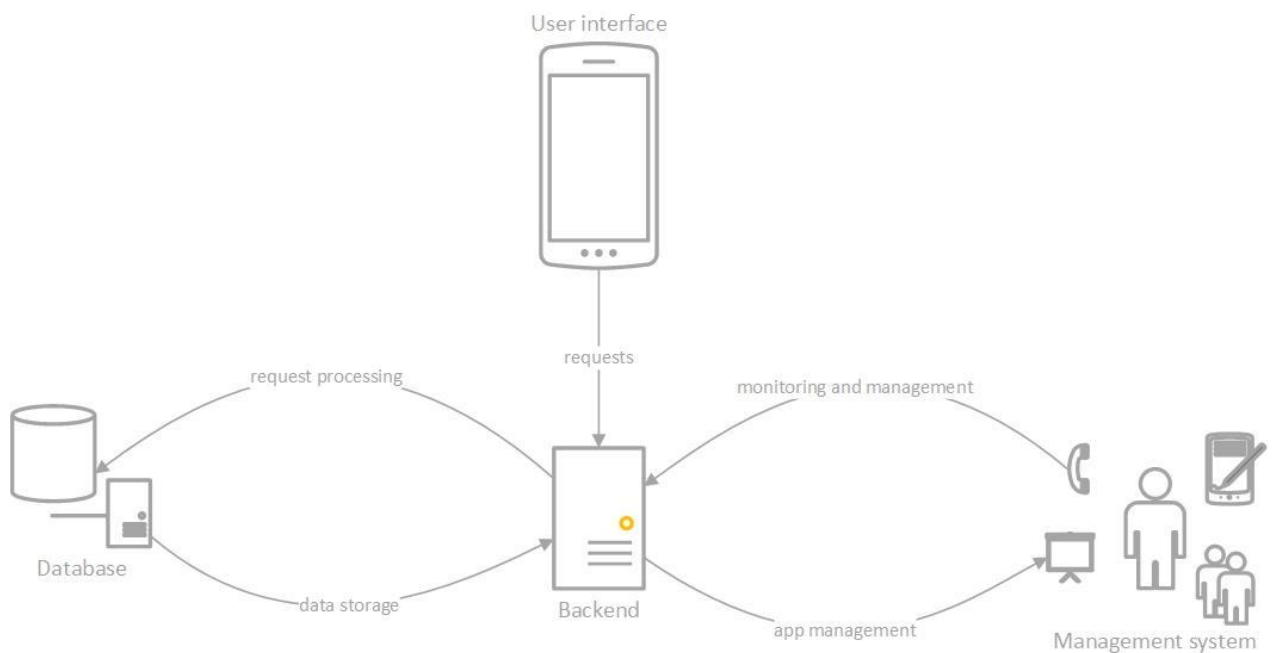


Рис. 3.1. Схема структури ПЗ додатку (Додаток Ф)

На етапі визначення технологій та інструментів для реалізації проекту обираються оптимальні технічні засоби, які відповідають завданням додатку для торгівлі криптовалютою на платформі Android. Основні аспекти включають:

1. Мови програмування: Вибір мови програмування, яка найбільше підходить для реалізації функцій додатку на Android. Наприклад, використання Java, Kotlin або C# для розробки під Android-платформу. Наш вибір спрямований на мову C#, оскільки вона ідеально поєднується з платформою Xamarin. Xamarin дозволяє розробникам писати один код, який може використовуватися для створення додатків для Android, iOS та Windows. Це забезпечує ефективну крос-платформенну розробку, а також підтримку різних мов програмування, таких як Objective-C, Java, C та C++.
2. Фреймворки та бібліотеки: Визначення необхідних фреймворків та бібліотек для швидкої та ефективної розробки. Наприклад, використання Android SDK, Retrofit для роботи з мережевими запитами, або Room для роботи з базою даних. Ось деякі з інструментів що використані в проекті:
 - Xamarin.Forms: Це фреймворк для створення крос-платформенних мобільних додатків.
 - Xamarin.Android: потрібен для більш тонкого налаштування Android-додатку.
 - Xamarin.Essentials: Бібліотека надає багато основних API для різних функцій, таких як робота з камерою, геолокацією, датчиками...
 - Xamarin.Auth: Для взаємодії з API криптовалютних бірж або вимагає аутентифікації.
 - Xamarin.Forms.Chart: Потрібна для візуалізації даних для відстеження курсів криптовалют та іншої статистики, можна використовувати бібліотеку для побудови графіків.
3. Інструменти для розробки: Вибір середовища розробки, такого як Android Studio, Xcode, або Visual Studio, яке забезпечить зручний розробницький процес, інструменти для налагодження та аналізу коду. Для розробки додатку обрано середовище Visual Studio, оскільки воно

володіє потужними можливостями та інтеграцією з платформою Xamarin. Visual Studio дозволяє комфортно працювати з кодом, забезпечуючи широкі можливості для налагодження, тестування та оптимізації додатку на різних платформах.

4. Версії Android: Визначення підтримуваних версій операційної системи Android для забезпечення максимальної широкої аудиторії користувачів. Наш вибір спрямований на максимальну сумісність із різними пристроями, тому ми встановлюємо мінімальну версію Android 8.0 (Oreo) як базову. Це дозволяє охопити значну частину ринку та забезпечити користувачам різних пристроїв можливість використання нашого додатку.
5. Безпека: Розгляд можливостей використання шифрування та інших засобів для забезпечення безпеки обробки та зберігання конфіденційної інформації. Для проекту було обрано SSL/TLS для шифрування з'єднань між Android додатком та сервером. Це запобігає перехопленню пакетів та забезпечує конфіденційність даних, які передаються через мережу.
6. Хмарні сервіси: Визначення можливостей інтеграції з хмарними сервісами для забезпечення резервного копіювання та синхронізації даних між пристроями користувачів [49].

Вибір правильних технологій є стратегічно важливим етапом, який визначить продуктивність, функціональність та успіх розробленого додатку для торгівлі криптовалютою.

3.2. Проектування прототипу, інтерфейсу, та бази даних

Прототип в розробці програмного забезпечення є функціональним взірцем або моделлю продукту, яка використовується для тестування, оцінки та вдосконалення його концепції перед фактичною реалізацією. Прототип дозволяє команді проекту та зацікавленим сторонам отримати візуальне уявлення про те, як повинен функціонувати кінцевий продукт та як відповідає вимогам користувачів.

Існує два основних види прототипів:

1. *Живий (робочий) прототип*: Це фактично функціонуючий варіант деякої частини або всього продукту. Він дозволяє користувачам та розробникам активно тестувати та експериментувати з функціоналом.
2. *Нерухомий прототип*: Це статичне зображення або макет інтерфейсу без реальної функціональності. Використовується для візуалізації дизайну, розміщення елементів та структури продукту.

Переваги використання прототипів включають можливість залучення відгуків в ранніх етапах розробки, уточнення вимог, а також зменшення ризику невірної сприйняття вимог між командою розробників та клієнтом. Прототипування сприяє покращенню якості та пришвидшенню процесу розробки [17].

Структура програмного забезпечення мобільного додатку для торгівлі криптовалютою на платформі Android може бути наступною:

1. *Користувацький інтерфейс (User Interface)*:

- *Домашній екран*: Показує основну інформацію про рахунок, портфелі, криптовалюту та активні замовлення.
- *Сторінка криптовалют*: Відображає список доступних криптовалют з їхніми цінами та статистикою.
- *Портфель*: Показує інформацію про позиції користувача в різних криптовалютах та загальну вартість портфеля.
- *Сторінка замовлень*: Дозволяє користувачу переглядати, редагувати та створювати замовлення на купівлю або продаж криптовалют.
- *Налаштування*: Містить налаштування користувача, такі як профіль, безпека, підписки тощо.

2. *Управління користувачем (User Management)*:

- *Реєстрація та аутентифікація*: Дозволяє новим користувачам створювати облікові записи та існуючим користувачам увійти в систему за допомогою електронної пошти та пароля.

- *Авторизація*: Контролює доступ до різних функцій додатку залежно від ролі та прав користувача.

3. Логіка додатку:

- *Керування портфелем*: Дозволяє користувачам додавати, видаляти та редагувати позиції в своєму портфелі криптовалют.
- *Обробка замовлень*: Забезпечує функціональність створення, виконання та скасування замовлень на купівлю або продаж криптовалют.
- *Отримання та оновлення даних*: Здійснює запити до зовнішніх сервісів для отримання актуальної інформації про ціни криптовалют та іншу статистику.
- *Обробка транзакцій*: Записує історію та деталі кожної транзакції, включаючи купівлю та продаж криптовалют.

4. Доступ до бази даних:

- *Зберігання користувацьких даних*: Зберігання інформації про користувачів, портфелі, замовлення та транзакції в базі даних.
- *Зберігання даних криптовалют*: Зберігання інформації про доступні криптовалюти та їхні ціни в базі даних.

5. Зв'язок з зовнішніми сервісами:

- *Інтеграція з API криптовалютних бірж*: Дозволяє отримувати актуальні дані про ціни, виконувати замовлення та отримувати статистику.
- *Взаємодія з платіжними шлюзами*: Забезпечує можливість здійснення фінансових операцій для купівлі та продажу криптовалют.

Ця структура програмного забезпечення дозволяє користувачам зручно керувати своїми криптовалютними активами, робити замовлення на покупку або продаж, отримувати актуальну інформацію про ринкові ціни та виконувати транзакції в безпечному та зручному способі [45].

Для ефективного розроблення додатку для торгівлі криптовалютою на платформі Android був реалізований прототип з метою візуалізації функціоналу та отримання згоди від зацікавлених сторін, зокрема від майбутніх користувачів та замовника проекту. Основні моменти у створенні прототипу включають:

1. *Дизайн інтерфейсу*: Розробка вигляду та розташування елементів на екрані для досягнення оптимального користувацького досвіду.
2. *Навігація*: Визначення логіки переходів між різними екранами та функціональними частинами додатку.
3. *Функціонал*: Реалізація базового функціоналу, який демонструє основні можливості додатку, зокрема перегляд балансу, виконання торгових операцій тощо.
4. *Збір фідбеку*: Проведення тестувань прототипу серед потенційних користувачів та замовника для збору відгуків та корекції дефектів.

Створення прототипу є необхідним етапом перед початком активної розробки, оскільки це дозволяє вчасно виявити та врахувати побажання та вимоги сторін, що максимально підвищує ефективність процесу розробки та зменшує ризики непорозумінь [23].

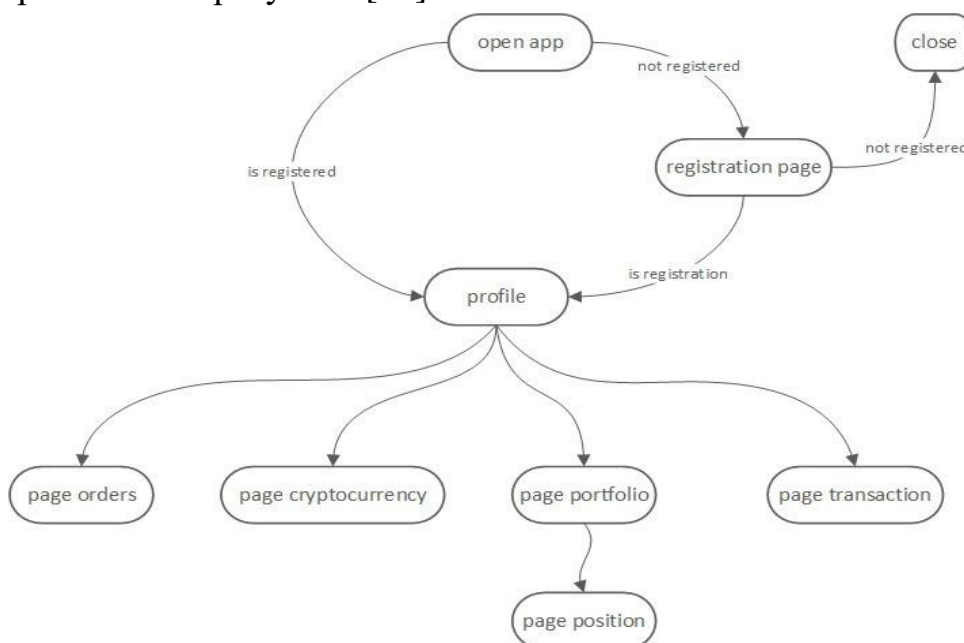


Рис. 3.2. Блок-схема алгоритму взаємодії користувача з додатком (Додаток X)

На етапі проектування дизайну користувацького інтерфейсу та структури бази даних визначаються ключові аспекти, що впливають на зручність та ефективність взаємодії користувача з додатком, а також на ефективне зберігання та обробку інформації.

Проектування інтерфейсу:

1. *Вигляд та стиль*: Розроблення естетичного та сучасного дизайну, який відповідає актуальним трендам і враховує психологію користувача.
2. *Логіка взаємодії*: Визначення логіки роботи кожного екрану та взаємодії користувача з різними функціональними елементами.
3. *Адаптивність*: Забезпечення адаптивності інтерфейсу до різних розмірів екранів пристроїв для максимального комфорту користувачів.

Проектування бази даних:

1. *Структура*: Визначення оптимальної структури бази даних для зберігання інформації про користувачів, транзакції, баланс та інші важливі дані.
2. *Безпека*: Розроблення заходів для захисту конфіденційної інформації, таких як шифрування даних та встановлення обмежень доступу.
3. *Швидкодія*: Врахування оптимізацій для швидкого доступу до даних та ефективної обробки запитів.
4. *Резервне копіювання*: Визначення стратегії резервного копіювання для запобігання втраті важливих даних у випадку непередбачуваних ситуацій.

Проектування інтерфейсу та бази даних є ключовим етапом у розробці програмного продукту. Інтерфейс фокусується на створенні зручного користувацького інтерфейсу, а база даних — на структурі та організації даних для ефективного звертання до них. Ці аспекти визначають подальші етапи розробки, тестування та впровадження додатку [9].

Графічний інтерфейс (GUI) є засобом взаємодії користувача з системою, який базується на використанні графічних елементів, таких як кнопки, вікна, меню та інші візуальні об'єкти, замість текстових команд. Головною метою

GUI є спрощення використання програм та операційних систем, роблячи їх більш інтуїтивно зрозумілими для користувачів.

Основні елементи GUI включають:

1. Вікна: Основні робочі області, в яких відображається інформація та яка може містити інші елементи.
2. Меню: Набір опцій та команд, які користувач може вибрати для виклику певних функцій.
3. Кнопки: Візуальні елементи, які користувач може натискати для виклику певних дій або функцій.
4. Текстові поля: Області для введення тексту або відображення інформації.
5. Списки та таблиці: Елементи, які відображають структуровані дані у вигляді списків або таблиць.

GUI дозволяє взаємодіяти з системами за допомогою екрану, миші, клавіатури або інших введених пристроїв, забезпечуючи інтуїтивно зрозуміле та зручне середовище для користувачів. Використання графічного інтерфейсу значно полегшує навігацію та використання програм та систем [14].

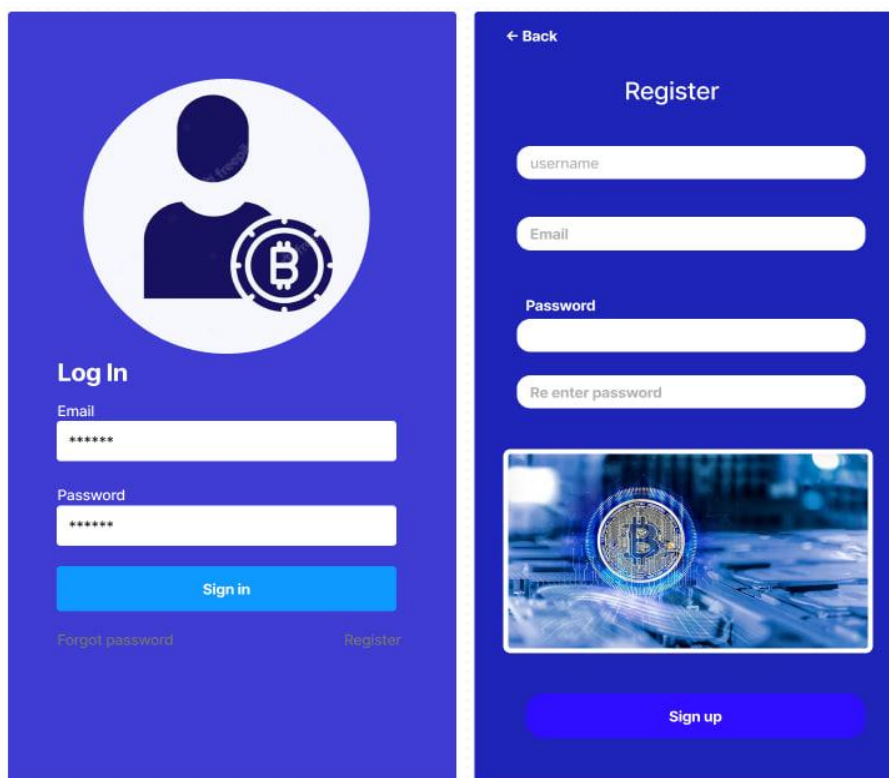


Рис. 3.3. Вхід/реєстрація користувача (Додаток Ц)

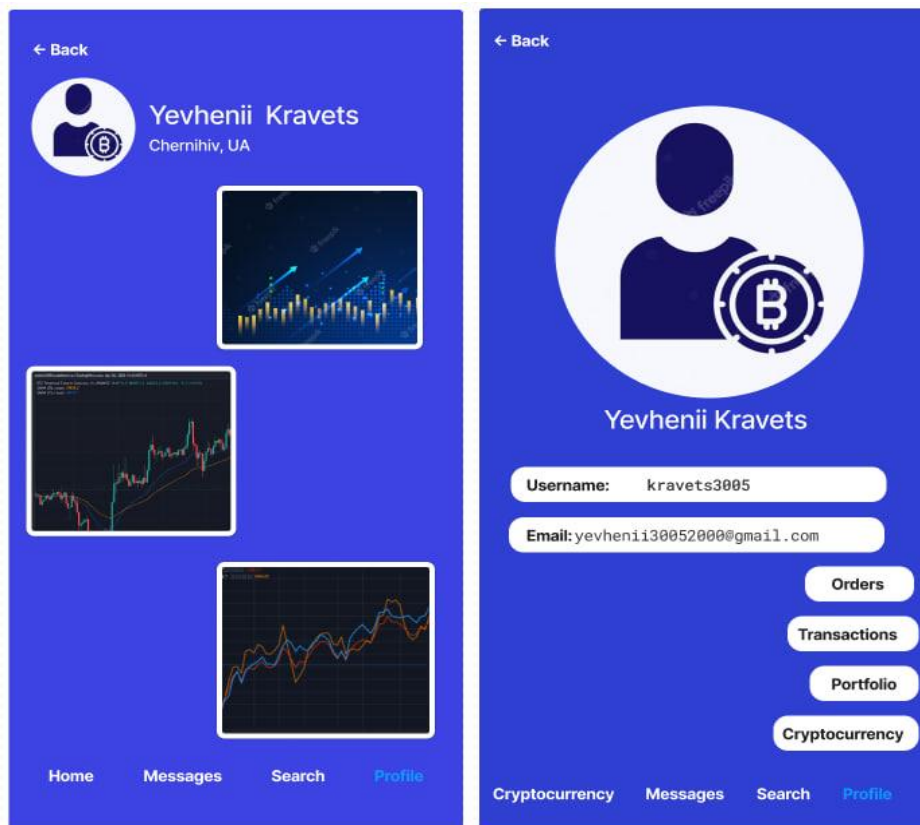


Рис. 3.4. Профіль користувача (Додаток Ш)

3.2.1. Розробка концептуальної моделі бази даних проєкту

База даних (БД) представляє собою систематизовану колекцію даних, яка зберігається та обробляється за допомогою комп'ютерної системи. Основна мета створення баз даних полягає в ефективному та структурованому зберіганні інформації для подальшого доступу та обробки.

Розробка концептуальної моделі бази даних є критичним етапом у процесі проектування інформаційної системи. Концептуальна модель бази даних визначає організацію даних, які будуть зберігатися у базі даних, а також встановлює взаємозв'язки між цими даними [52].

Основні характеристики баз даних включають:

1. *Дані*: База даних містить дані різних типів, які можуть бути числовими, текстовими, графічними тощо.
2. *Таблиці*: Дані в базі організовані у вигляді таблиць, де кожен рядок представляє запис, а кожна колонка - поле або атрибут.
3. *Відносини*: Зв'язки між таблицями визначаються для уникнення дублювання інформації та забезпечення консистентності даних. Це

важливий аспект реляційних баз даних, який дозволяє ефективно та економічно управляти інформацією.

4. *Мова запитів*: Бази даних використовують мови запитів, такі як SQL (Structured Query Language), для взаємодії з даними, пошуку необхідної інформації та внесення змін.
5. *Система управління базами даних (СУБД)*: Це програмне забезпечення, яке дозволяє створювати та управляти базами даних. СУБД надає інтерфейс для взаємодії між користувачами та базою даних, дозволяючи виконувати операції, такі як додавання, вилучення, оновлення та пошук даних.

Таблиця 3.1. (Додаток Ш)

Сутності бази даних

Сутність	Атрибути
1	2
Користувач (User)	<ul style="list-style-type: none"> – Ідентифікатор користувача (User ID) – Ім'я користувача (Username) – Дата реєстрації (Registration Date) – Баланс рахунку (Account Balance)
Криптовалюта (Cryptocurrency)	<ul style="list-style-type: none"> – Ідентифікатор криптовалюти (Cryptocurrency ID) – Назва (Name) – Скорочена назва (Symbol) – Поточна ціна (Current Price) – Кількість одиниць в обігу (Circulating Supply)

Продовження табл. 3.1.

1	2
Портфель (Portfolio)	<ul style="list-style-type: none"> – Ідентифікатор портфеля (Portfolio ID) – Ідентифікатор користувача (User ID) – Назва портфеля (Portfolio Name) – Дата створення (Creation Date)
Позиція (Position)	<ul style="list-style-type: none"> – Ідентифікатор позиції (Position ID) – Ідентифікатор портфеля (Portfolio ID) – Ідентифікатор криптовалюти (Cryptocurrency ID) – Кількість одиниць (Quantity) – Середня вартість (Average Cost)
Замовлення (Order)	<ul style="list-style-type: none"> – Ідентифікатор замовлення (Order ID) – Ідентифікатор користувача (User ID) – Ідентифікатор криптовалюти (Cryptocurrency ID) – Тип замовлення (Order Type) – Кількість одиниць (Quantity) – Ціна замовлення (Order Price) – Дата замовлення (Order Date)

Продовження табл. 3.1.

1	2
Транзакція (Transaction)	<ul style="list-style-type: none"> – Ідентифікатор транзакції (Transaction ID) – Ідентифікатор користувача (User ID) – Ідентифікатор криптовалюти (Cryptocurrency ID) – Тип транзакції (Transaction Type) – Кількість одиниць (Quantity) – Ціна транзакції (Transaction Price) – Дата транзакції (Transaction Date)
Журнал входу (Login Log)	<ul style="list-style-type: none"> – Ідентифікатор запису журналу (Log ID) – Ідентифікатор користувача (User ID) – Дата та час входу (Login DateTime) – IP-адреса (IP Address)
Новина (News)	<ul style="list-style-type: none"> – Ідентифікатор новини (News ID) – Заголовок (Title) – Текст новини (News Text) – Дата публікації (Publication Date)

Продовження табл. 3.1.

1	2
Повідомлення (Message)	<ul style="list-style-type: none">– Ідентифікатор повідомлення (Message ID)– Ідентифікатор користувача відправника (Sender User ID)– Ідентифікатор користувача отримувача (Recipient User ID)– Текст повідомлення (Message Text)– Дата надсилання (Sending Date)

Нижче наведені відносини між сутностями:

1) *Користувач (User):*

- Має один або багато Портфелів (Portfolio).
- Робить одне або багато Замовлень (Order).
- Здійснює одну або багато Транзакцій (Transaction).
- Має один або багато Відгуків (Review).
- Має один або багато записів в Журналі входу (Login Log).

2) *Криптовалюта (Cryptocurrency):*

- Має одну або багато Позицій (Position) у різних Портфелях.
- Має одну або багато Замовлень (Order).
- Має одну або багато Транзакцій (Transaction).
- Має одну або багато Відгуків (Review).

3) *Портфель (Portfolio):*

- Належить одному Користувачеві (User).
- Містить одну або багато Позицій (Position).

4) *Позиція (Position):*

- Міститься в одному Портфелі (Portfolio).

- Пов'язана з однією Криптовалютою (Cryptocurrency).
- 5) *Замовлення (Order)*:
- Створено одним Користувачем (User).
 - Відноситься до однієї Криптовалюти (Cryptocurrency).
- 6) *Транзакція (Transaction)*:
- Здійснена одним Користувачем (User).
 - Включає одну Криптовалюту (Cryptocurrency).
- 7) *Відгук (Review)*:
- Написаний одним Користувачем (User).
 - Стосується однієї Криптовалюти (Cryptocurrency).
- 8) *Новина (News)*:
- Не має прямих відносин з іншими сутностями. Може бути використана для збереження інформації про новини без прив'язки до конкретних користувачів або криптовалют.
- 9) *Повідомлення (Message)*:
- Містить інформацію про взаємодію між двома Користувачами (User).
- 10) *Журнал входу (Login Log)*:
- Містить історію входів Користувачів (User) до системи.
- Ці відносини відображають зв'язки між різними сутностями бази даних та визначають, як вони взаємодіють між собою [37].

Бази даних використовуються для забезпечення надійного та ефективного зберігання великих обсягів інформації, спрощення доступу до неї та підтримки різноманітних операцій, що виконуються в інформаційних системах. Вони є важливою складовою багатьох застосунків, від бізнес-систем до наукових досліджень. Бази даних також відіграють ключову роль у підтримці консистентності, цілісності та безпеки даних, що робить їх невід'ємним інструментом для забезпечення надійності та ефективності інформаційних процесів.

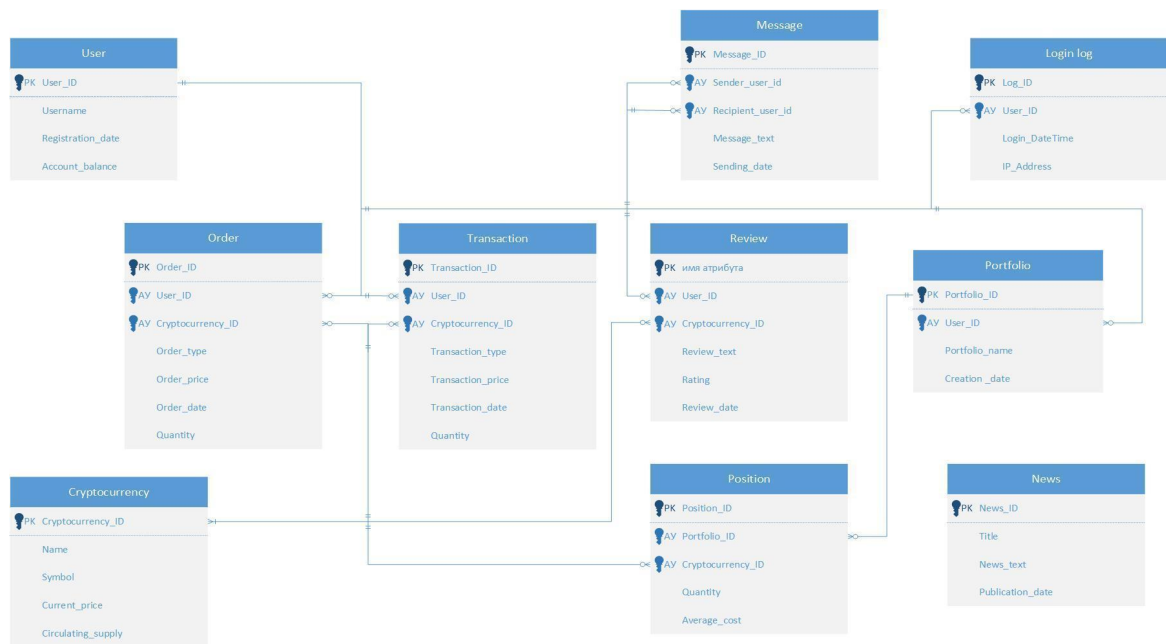


Рис. 3.5. Діаграма типу сутність-зв'язок концептуальної моделі бази даних (Додаток Ю)

3.2.2. Побудова логічної моделі бази даних проєкту

Будуючи логічну модель бази даних для проєкту створення додатку для торгівлі криптовалютою на платформі Android, необхідно детально визначити сутності, їх атрибути та відносини, враховуючи специфіку функціональності та потреб користувачів, щоб забезпечити ефективне та оптимальне зберігання та обробку даних у базі даних додатку.

Нижче наведена логічна модель бази даних:

Таблиця 3.2. (Додаток Я)

Опис таблиць бази даних

Таблиця	Опис
1	2
"Користувач" (User)	Містить основну інформацію про користувачів, таку як їх унікальний ідентифікатор (user_id), ім'я користувача (username), електронну пошту (email), пароль (password) та дату реєстрації (registration_date).

Продовження табл. 3.2

1	2
"Позиція" (Position)	Містить інформацію про позиції в портфелях користувачів, таку як їх унікальний ідентифікатор позиції (position_id), ідентифікатор портфеля, до якого належить позиція (portfolio_id), ідентифікатор криптовалюти, яка знаходиться в позиції (cryptocurrency_id)_ та кількість криптовалюти в позиції (quantity).
"Замовлення" (Order)	Містить інформацію про замовлення користувачів на купівлю або продаж криптовалют, таку як унікальний ідентифікатор замовлення (order_id), ідентифікатор користувача, який розмістив замовлення (user_id), ідентифікатор криптовалюти, на яку оформлено замовлення (cryptocurrency_id), тип замовлення, що може бути "купівля" або "продаж" (order_type), кількість криптовалюти в замовленні (quantity), ціна за одиницю криптовалюти в замовленні (price) та дата замовлення (order_date).

Продовження табл. 3.2

1	2
"Транзакція" (Transaction)	Містить інформацію про транзакції з криптовалютами, які здійснюються користувачами, таку як унікальний ідентифікатор транзакції (transaction_id), ідентифікатор користувача, який здійснив транзакцію (user_id), ідентифікатор криптовалюти, якою здійснено транзакцію (cryptocurrency_id), тип транзакції, що може бути "купівля" або "продаж" (transaction_type), кількість криптовалюти в транзакції (quantity), ціна за одиницю криптовалюти в транзакції (price) та дата транзакції (transaction_date).
"Портфель" (Portfolio)	Містить інформацію про портфелі користувачів, таку як їх унікальний ідентифікатор портфеля (portfolio_id) та ідентифікатор користувача, до якого належить портфель (user_id).
"Криптовалюта" (Cryptocurrency)	Містить основну інформацію про криптовалюти, таку як їх унікальний ідентифікатор (cryptocurrency_id), назву (name), символ (symbol) та ринкову ціну (market_price).

Таблиці у базі даних проекту "Створення додатку для торгівлі криптовалютою на платформі Android" пов'язані між собою за допомогою

зовнішніх ключів. Відносини між таблицями дозволяють встановлювати зв'язки та залежності між даними, що зберігаються в цих таблицях.

Опис відносин між таблицями:

1. Таблиця "Користувач" (User) пов'язана з таблицею "Портфель" (Portfolio) через зовнішній ключ `user_id`. Кожен користувач може мати один або більше портфелів.
2. Таблиця "Портфель" (Portfolio) пов'язана з таблицею "Позиція" (Position) через зовнішній ключ `portfolio_id`. У кожному портфелі можуть бути різні позиції з криптовалютами.
3. Таблиця "Позиція" (Position) пов'язана з таблицею "Криптовалюта" (Cryptocurrency) через зовнішній ключ `cryptocurrency_id`. Кожна позиція в портфелі пов'язана з конкретною криптовалютою.
4. Таблиця "Замовлення" (Order) пов'язана з таблицями "Користувач" (User) і "Криптовалюта" (Cryptocurrency) через зовнішні ключі `user_id` і `cryptocurrency_id`. Кожне замовлення пов'язане з користувачем, який його розмістив, та криптовалютою, на яку оформлено замовлення.
5. Таблиця "Транзакція" (Transaction) пов'язана з таблицями "Користувач" (User) і "Криптовалюта" (Cryptocurrency) через зовнішні ключі `user_id` і `cryptocurrency_id`. Кожна транзакція пов'язана з користувачем, який її здійснив, та криптовалютою, якою була здійснена транзакція.

Ці відносини дозволяють здійснювати зв'язки між різними сутностями бази даних і забезпечувати цілісність даних. Наприклад, зв'язок між таблицями "Портфель" (Portfolio) і "Позиція" (Position) дозволяє відстежувати, які криптовалюти знаходяться в конкретному портфелі. Зв'язок між таблицями "Замовлення" (Order) і "Користувач" (User) дозволяє відстежувати, який користувач розмістив певне замовлення. Зв'язок між таблицями "Транзакція" (Transaction) і "Криптовалюта" (Cryptocurrency) дозволяє відстежувати, якою криптовалютою була здійснена кожна транзакція.

Нижче наведена логічна схема цих взаємозв'язків:

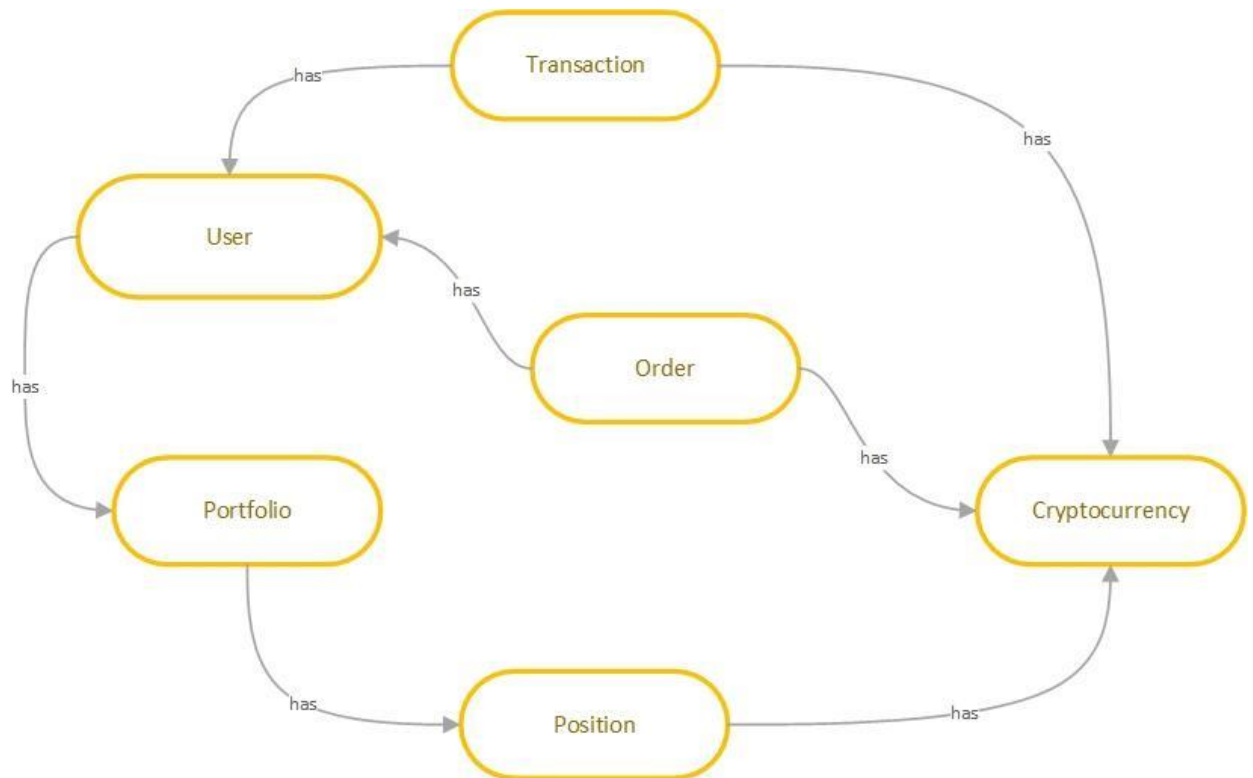


Рис. 3.6. Схема взаємозв'язків логічної бази даних (Додаток D)

3.3. Написання коду

На етапі написання коду розпочинається активний процес реалізації проекту, який включає в себе декілька ключових аспектів:

1. *Розробка основних функцій:* Програмісти приступають до реалізації основних функцій додатку згідно з визначеними вимогами. Це може включати створення графічного інтерфейсу, роботу з базою даних, обробку мережевих запитів та інші ключові функції.
2. *Кодування згідно стандартів:* Під час написання коду дотримання стандартів кодування, наприклад SOLID є важливим елементом. Це полегшує спільну роботу розробників, підтримку та розуміння коду.
3. *Використання патернів проектування:* Використання патернів проектування дозволяє створити ефективний та легко зрозумілий код, сприяє його масштабованості та обслуговуваності.

4. *Тестування в процесі розробки*: Проведення модульного тестування на кожному етапі написання коду допомагає виявляти та виправляти помилки та недоліки на ранніх стадіях.
5. *Документування коду*: Кожна частина коду повинна бути добре задокументованою для полегшення розуміння та співпраці між розробниками.

Алгоритми в програмуванні є основними будівельними блоками, які визначають послідовність кроків для вирішення конкретної задачі чи виконання певного процесу. Вони є невід'ємною частиною розробки програмного забезпечення, надаючи структуру, логіку та ефективність виконання коду. Алгоритми сприяють абстракції завдань, полегшуючи розуміння та управління складністю програм, а також забезпечують повторне використання в кодї для оптимізації розробки та підтримки [47]. Для додатку для торгівлі криптовалютою на платформі Android ми розробили наступні ключові алгоритми:

1. *Алгоритм отримання криптовалютних даних*. Цей алгоритм використовується для: встановлення з'єднання з API криптовалютних бірж, що надає доступ до актуальних котирувань та інших даних; запиту необхідної інформації, такої як ціна, обсяг, часові маркери тощо; отримання відповіді та обробка отриманих даних.
2. *Алгоритм аналізу криптовалютних даних*. Цей алгоритм використовується для: розпарсування отриманих даних та створення об'єктів або моделей, що представляють криптовалютні дані; виконання обробки даних, такої як фільтрація за певними критеріями, обчислення показників, розрахунки вартості тощо; використання алгоритмів аналізу даних, таких як технічний аналіз або статистичні моделі, для отримання корисних інсайтів з криптовалютних даних.
3. *Алгоритм керування торговельними операціями*. Цей алгоритм використовується для: розробки алгоритму для вибору оптимальної стратегії торгівлі криптовалютою, враховуючи різні фактори, такі як

ризика, ринкові умови, налаштування користувача тощо; визначення критеріїв для виконання торговельних операцій, таких як купівля, продаж або установа замовлень на певному рівні ціни; реалізації автоматичного виконання торговельних операцій на основі заданого алгоритму та вхідних параметрів [15].

Написання коду є ключовим етапом у розробці, оскільки від якості та ефективності коду залежить подальша успішна реалізація проекту. Також це вимагає дотримання кращих практик програмування та врахування принципів сумісності, що сприяє підтримці, розширенню та зменшенню можливих помилок у майбутньому.

```
30 +     [HttpGet]
31 +     public IActionResult Range(DateTime startDate, DateTime endDate)
32 +     {
33 +         if (endDate < startDate)
34 +             return Json(new { success = false, error = "Start date is less than end date" });
35 +
36 +         var events = _service.GetInRange(startDate, endDate);
37 +         return Ok(events);
38 +     }
39 +
40 +     [HttpGet]
41 +     public IActionResult Details(Guid id)
42 +     {
43 +         try
44 +         {
45 +             var @event = _service.Details(id);
46 +
47 +             if (@event == null)
48 +                 return Json(new { success = false });
49 +             else
50 +                 return Ok(@event);
51 +         }
52 +         catch (Exception e)
53 +         {
54 +             _logger.LogInformation($"Operation error: details [{e}]");
55 +         }
56 +         return Json(new { success = false });
57 +     }
```

Рис. 3.7. Частина коду контролера для *Http*-запитів (Додаток F)

```

62 +     [InlineData(null)]
63 +     [InlineData("123456789012345678901234567890 invalid")]
64 +     public void TaskController_Create_WithInvalidTitle_ReturnsError(string taskTitle)
65 +     {
66 +         // Arrange
67 +         var model = new TaskRequestModel()
68 +         {
69 +             Title = taskTitle,
70 +             Description = "desc",
71 +             Date = DateTime.Now,
72 +             Color = "#ffffff"
73 +         };
74 +
75 +         var logger = new Mock<Microsoft.Extensions.Logging.ILogger<TaskController>>();
76 +         var context = InitHelpers.GetDbContext();
77 +         var taskService = new Mock<IServiceItem<Task>>();
78 +
79 +         var validator = new TaskRequestModelValidator();
80 +         var service = new TaskController(logger.Object, context.Object, taskService.Object, validator);
81 +
82 +         //Act
83 +         var result = service.Create(model);
84 +
85 +         //Assert
86 +         Assert.NotNull(result);
87 +
88 +         var validationResult = validator.Validate(model);
89 +
90 +         Assert.NotNull(validationResult);
91 +         Assert.False(validationResult.IsValid);
92 +     }

```

Рис. 3.8. Unit-тест для методу контролеру (Додаток G)

3.4. Планування тестів та тестування

Тестування - це процес перевірки, валідації та верифікації програмного продукту з метою забезпечення його якості, відповідності вимогам та виявлення можливих дефектів чи помилок. Цей процес включає в себе різні види тестів та стратегій, спрямованих на різні аспекти програми.

Однією з основних мет тестування є впевненість, що програма працює коректно та відповідає функціональним та нефункціональним вимогам. Це включає перевірку функціональності, продуктивності, надійності, безпеки та інших характеристик програмного забезпечення.

Тестування допомагає виявити та виправити помилки, забезпечуючи високу якість програмного продукту та зменшуючи ризик неполадок при його використанні.

Планування тестів - ключовий етап в процесі розробки програмного забезпечення, оскільки від його якісного виконання залежить виявлення та виправлення помилок, а також забезпечення високої якості продукту.

1. *Стратегія тестування*: Визначення загальної стратегії тестування, включаючи обрані методи та підходи. Особливу увагу слід звернути на покриття тестування - як широко охоплені функціональні та нефункціональні вимоги.
2. *Обладнання тестового середовища*: Забезпечення необхідними ресурсами для проведення тестів. Це може включати в себе конфігурації обладнання, налаштування середовища, визначення тестових наборів.
3. *Вибір тестових кейсів*: Аналіз функціональності та визначення конкретних тестових кейсів для випробування різних аспектів продукту. Важливо враховувати різні сценарії використання та варіації вхідних даних. Основні елементи тестового кейсу включають:
 - *Опис тесту*: Короткий опис тесту або ціль тестування.
 - *Вхідні дані*: Дані, які потрібні для запуску тесту.
 - *Кроки виконання*: Кроки, які тестер повинен виконати для проведення тесту. Кроки повинні бути чіткими та послідовними.
 - *Очікувані результати*: Опис того, що має статися після виконання кожного кроку або в кінці тесту. Ці очікувані результати порівнюються з фактичними результатами для визначення успішності тесту.
 - *Умови виконання*: Будь-які передумови або обмеження, які повинні бути виконані для успішного проведення тесту.
 - *Тестові дані*: Конкретні дані, які використовуються для тестування. Це може включати в себе як вхідні дані, так і вже існуючі дані в системі.

– *Статус тесту*: Позначка, що вказує на стан тесту (пройдено, не пройдено, потребує перевірки тощо).

4. *Планування ресурсів*: Оцінка потреб у ресурсах, таких як людські, фінансові та технічні. Визначення кількості і кваліфікації тестувальників, а також виділення фінансових ресурсів для проведення тестів.
5. *Тестові дати та терміни*: Встановлення графіка проведення тестів, включаючи дати початку і завершення різних етапів тестування. Важливо враховувати можливі взаємодії з іншими етапами розробки.
6. *Метрики тестування*: Визначення ключових метрик, які дозволять визначити ефективність тестів. Це може включати в себе покриття коду, кількість виявлених помилок, час тестування тощо.
7. *Процедури та відповідальності*: Розподіл завдань та визначення ролей між учасниками тестування. Чітко визначені процедури виявлення, документування та виправлення помилок.
8. *Засоби для відстеження тестування*: Визначення інструментів для відстеження процесу тестування та результатів. Це може бути використання систем управління помилками, тестовими звітами, а також засобами автоматизованого тестування. Для даного проекту було обрано Jira - це один з найбільш популярних засобів для керування проектами та відстеження тестування. Він надає можливість створення тестових циклів, відстеження дефектів, а також гнучкість у створенні звітів та інтеграції з іншими інструментами.
9. *Процедури відновлення*: Розроблення плану відновлення в разі виявлення критичних помилок, в тому числі процедур відкату до попередніх версій системи. Обрано платформу GitHub для контролю версій з метою ефективного моніторингу та відновлення кодової бази.
10. *Оцінка результатів*: Визначення процедур та критеріїв оцінки ефективності проведених тестів та прийняття рішення про готовність продукту до випуску [34].

Інтеграція всіх аспектів у єдиний план тестування є ключовим етапом, спрямованим на забезпечення високої якості та надійності програмного забезпечення. Гармонійне взаємодія між проведенням різноманітних тестів, виявленням та виправленням помилок, а також впровадженням ефективних процедур відновлення покращує процес розробки та забезпечує задоволення високих стандартів якості продукту.

Тестування може включати різні види випробувань, такі як функціональне тестування, що перевіряє, чи виконують функції програми відповідно до визначених вимог; тестування продуктивності, яке оцінює швидкість та ефективність програми; тестування безпеки, яке визначає стійкість до атак та інші види тестувань.

Тестування також може бути проведене на різних етапах розробки, включаючи модульне тестування окремих компонентів, інтеграційне тестування взаємодії між компонентами, та системне тестування в рамках всього програмного продукту. Такий підхід допомагає виявити та виправити помилки на ранніх етапах розробки, зменшуючи ризик неполадок при впровадженні.

На етапі тестування коду проводиться ретельна перевірка функціональності окремих компонентів для забезпечення високої якості програмного забезпечення. Цей етап включає в себе кілька важливих аспектів:

1. *Модульне тестування:* Кожен модуль або компонент коду тестується ізольовано, щоб перевірити його коректність та взаємодію з іншими частинами системи.
2. *Валідація відповідності вимогам:* Код перевіряється на відповідність визначеним вимогам, зокрема функціональним та нефункціональним.
3. *Тестування реакції на помилки:* Сценарії тестування розробляються для перевірки того, як система реагує на можливі помилки, такі як некоректні дані чи втрата з'єднання.
4. *Тестування продуктивності:* Оцінка швидкодії та ефективності коду під час великих обсягів даних чи великої кількості користувачів.

5. *Автоматизоване тестування:* Використання інструментів автоматизації для швидкого та ефективного виконання тестів [40].

Тестування коду є важливою частиною розробки, оскільки воно дозволяє виявити та усунути потенційні проблеми ще до випуску продукту. Тестова покриття коду покращує його стабільність та надійність.

3.5. Виправлення помилок

Після проведення тестування програмного забезпечення та аналізу результатів, виявлені помилки вимагають негайного виправлення для забезпечення високої якості та надійності продукту. Процес виправлення помилок включає в себе кілька етапів:

1. Аналіз результатів тестування:

- Ретельний аналіз звітів тестів для ідентифікації всіх виявлених помилок та аномалій.
- Класифікація помилок за їхньою серйозністю та визначенням пріоритетів виправлення.

2. Планування виправлення:

- Створення плану виправлення помилок, в якому вказані терміни та відповідальні особи за кожну виявлену помилку.
- Визначення термінів виправлення критичних помилок та розподіл пріоритетів.

3. Виправлення коду:

- Командний аналіз та виправлення програмного коду, пов'язаного з виявленими помилками.
- Застосування кращих практик розробки для уникнення подібних помилок у майбутньому.

4. Повторне тестування:

- Запуск повторних тестів для перевірки виправлених компонентів та підтвердження їхньої правильності.

- Впевнення в тому, що виправлення не вплинуло на інші частини системи та не викликало нові проблеми.

5. *Валідація та верифікація:*

- Використання методів валідації для перевірки, чи виправлені помилки відповідають вихідним вимогам та специфікаціям.
- Здійснення верифікації, щоб переконатися, що система досі відповідає всім вимогам та функціональним характеристикам.

6. *Тестування на відновлення:*

- Використання процедур відновлення для перевірки, як система реагує на виправлення та чи забезпечується її стабільність.

7. *Документування:*

- Оновлення документації проекту з вказанням виправлених помилок та внесених змін.
- Формування звіту про виправлення помилок для оцінки ефективності та якості виправлень.

8. *Успішне завершення виправлень:*

- Підтвердження успішного виправлення помилок та готовності продукту до продовження процесу розробки та тестування.

Цей процес забезпечує покращення якості програмного забезпечення та підготує його до вдалого випуску на ринок [7].

В процесі тестування нашого продукту було використано різноманітні методи та типи тестів з метою забезпечення високої якості програмного забезпечення. Основні напрямки включали в себе тестування на рівні одиниць (unit testing), інтеграційне тестування (integration testing), тестування функціоналу, та системне тестування.

Тестування на рівні одиниць дозволяє перевірити кожен окрему частину програмного коду на коректність та відповідність специфікаціям. Це дозволяє виявити та виправити будь-які потенційні помилки в ізоляції.

Інтеграційне тестування було важливою частиною процесу, оскільки воно перевіряло взаємодію між різними компонентами системи. Цей підхід

допоміг виявити можливі проблеми, які можуть виникнути при об'єднанні окремих частин.

Тестування функціоналу визначає, чи відповідає продукт вимогам та специфікаціям, перевіряючи його основні функції та можливості.

Системне тестування охопило великий масштаб продукту, перевіряючи його відповідність загальним цілям та вимогам проекту.

Результати тестування були детально задокументовані, включаючи виявлені помилки та шляхи їх виправлення. Цей підхід забезпечив високий рівень якості та надійності нашого продукту перед його випуском на ринок.

РОЗДІЛ 4. ВИПУСК ГОТОВОГО ПРОДУКТУ НА РИНОК

Випуск готового продукту на ринок є заключним та вирішальним етапом у життєвому циклі розробки проекту. Цей процес включає в себе кілька ключових елементів, що сприяють успішному введенню продукту до користувачів.

По-перше, перед випуском необхідно впевнитися, що весь функціонал продукту відповідає вимогам та стандартам якості. Проведення остаточних тестів дозволяє виявити та виправити будь-які потенційні проблеми чи помилки перед виведенням продукту на ринок.

Другим важливим аспектом є розробка ефективної стратегії маркетингу та реклами, спрямованої на привернення уваги та зацікавлення цільової аудиторії. Це може включати в себе створення рекламних матеріалів, ведення кампаній у соціальних мережах, а також співпрацю з маркетинговими партнерами.

Забезпечення ефективного обслуговування та підтримки користувачів також важливо на етапі випуску. Це може включати в себе створення інструкцій, надання чат-підтримки, відповіді на запитання та вирішення можливих проблем [10].

Нарешті, здійснення контролю за відгуками користувачів та збирання аналітики можуть допомогти удосконалити продукт у майбутньому та адаптувати його до потреб ринку.

Узагальнюючи, випуск готового продукту на ринок вимагає виваженості, стратегічного планування та уважного відношення до вимог та потреб користувачів.

4.1. Впровадження продукту

Після успішного завершення етапу розробки та тестування, настав час впровадження продукту на ринок. Цей етап включає в себе кілька ключових кроків, спрямованих на максимальне надання користувачам нового додатку та забезпечення ефективної його роботи.

1. Розгортання інфраструктури:
 - Підготовка інфраструктури для розгортання додатку, включаючи сервери, бази даних та інші потрібні ресурси.
 - Забезпечення високої доступності серверів та відповідної мережевої інфраструктури.
2. Установка додатку:
 - Розгортання програмного забезпечення на визначених серверах чи хмарних обчисленнях.
 - Виконання необхідних конфігурацій для відповідності вимогам додатку.
3. Тестування впровадження:
 - Проведення тестів на етапі впровадження для перевірки коректності розгортання та взаємодії з іншими системами.
 - Виявлення та вирішення можливих проблем, що виникають під час установки.
4. Завантаження початкових даних:
 - Встановлення механізмів завантаження початкових даних для коректної роботи додатку з самого початку.
 - Забезпечення правильної синхронізації та оновлення інформації.
5. Навчання персоналу та користувачів:
 - Проведення навчання для внутрішнього персоналу, який буде взаємодіяти з продуктом.
 - Надання користувачам інструкцій та рекомендацій з використання нового додатку.
6. Моніторинг впровадження:
 - Встановлення систем моніторингу для стеження за продуктивністю та стабільністю додатку в реальному часі.
 - Вчасне виявлення і вирішення можливих проблем під час впровадження.

7. Підтримка впровадження:

- Надання технічної та консультаційної підтримки користувачам протягом початкового періоду впровадження.
- Вирішення запитань та допомога в разі виникнення труднощів.

8. Оновлення та підтримка:

- Регулярне випускання оновлень для забезпечення безпеки, додавання нових функцій та вирішення знайдених помилок.
- Забезпечення довгострокової технічної підтримки для користувачів.

Впровадження продукту на ринок - це складний процес, що вимагає детального планування та скоординованої роботи всіх сторін. Розгортання інфраструктури, установка додатку та тестування впровадження є критичними етапами, які визначають успіх нового продукту.

Надійність і коректність впровадження визначаються не лише технічними аспектами, але й готовністю персоналу та користувачів. Навчання персоналу, моніторинг впровадження та підтримка грають важливу роль у стабільному функціонуванні системи.

Підтримка впровадження не завершується після перших етапів. Регулярні оновлення та довгострокова технічна підтримка забезпечують актуальність та ефективність продукту на протязі його життєвого циклу [38].

В цілому, вдале впровадження – це не просто завершення одного етапу, але і початок взаємодії з користувачами та подальшого розвитку продукту для задоволення їхніх потреб та очікувань.

4.2. Підтримка користувачів

Після успішного впровадження додатку на ринок виникає важливе завдання – надання якісної та ефективної сервісної підтримки користувачам. Цей аспект грає ключову роль у задоволенні потреб та очікувань користувачів, а також у створенні позитивного іміджу продукту. Організація сервісної підтримки включає в себе наступні кроки:

1. *Створення служби підтримки:*

- Формування команди кваліфікованих фахівців, здатних вирішувати технічні та функціональні питання користувачів.
- Визначення контактних каналів для звернень, таких як телефонна лінія, електронна пошта, чат або онлайн-форми.

2. *Навчання персоналу:*

- Проведення тренінгів та навчання для членів служби підтримки щодо особливостей та функціоналу додатку.
- Забезпечення доступу до актуальної документації та бази знань.

3. *Запитання та відповіді (FAQ):*

- Створення та постійне оновлення сторінки часто запитуваних питань (FAQ) для надання користувачам швидкої інформації.
- Активна промоція та направлення користувачів на розділ FAQ для самостійного вирішення питань.

4. *Система відстеження звернень:*

- Впровадження системи для відстеження та аналізу звернень користувачів.
- Призначення відповідальних фахівців для оперативного вирішення питань.

5. *Аналіз та вдосконалення:*

- Проведення регулярного аналізу статистики та фідбеку від користувачів.
- Внесення змін та вдосконалень у роботу служби підтримки на основі отриманих даних.

6. *Користувальницькі форуми та спільноти:*

- Забезпечення можливості для користувачів обмінюватися досвідом та порадами у віртуальних форумах або спільнотах.
- Активне взаємодії з учасниками спільнот для вирішення їхніх питань.

7. Онлайн-консультації:

- Забезпечення можливості отримання консультацій в режимі реального часу через чат або відеоконференції.
- Зручне планування та проведення онлайн-консультацій для користувачів [13].

Надання ефективної та якісної підтримки користувачам є важливим елементом успіху продукту на ринку. Взаємодія з клієнтами, вирішення їхніх запитань та вироблення швидких рішень щодо можливих труднощів не лише сприяє стабільності та задоволенню користувачів, але й впливає на імідж бренду. Позитивний досвід взаємодії з підтримкою робить продукт більш привабливим для потенційних та існуючих клієнтів, створюючи основу для довгострокових відносин та вірних клієнтських баз.

4.3. Завершення проекту

Етап завершення проекту є важливою фазою в управлінні проектом. На цьому етапі проводиться оцінка виконання поставлених завдань та досягнення цілей проекту. Здійснюється аналіз результатів, включаючи витрати та терміни. Формується вся необхідна документація, така як звіти, аналіз ризиків та інструкції з експлуатації.

Основною метою є офіційне закриття проекту, включаючи підписання актів приймання-передачі та завершення контрактних зобов'язань. Крім того, важливим елементом є вивчення уроків з проекту, аналіз його успіхів та помилок для подальшого вдосконалення роботи. Завершення проекту також включає в себе визнання та подяку команді за їхню працю, що є ключовим для підтримання морального духу та стимулювання в подальшій діяльності.

На цьому етапі ми докладно аналізуємо досягнення та здійснюємо важливі кроки для успішного завершення проекту.

Оцінка досягнень: Аналізуємо ступінь відповідності результатів проекту вихідним цілям та завданням. Перевіряємо, чи були виконані завдання вчасно

та ефективно, а також чи було досягнуто запланованих результатів. Важливим аспектом є порівняння планових та фактичних фінансових показників проекту.

Внесення коректив: На основі аналізу результатів визначаємо можливості для поліпшень та корекцій у виконанні подальших проектів. Оцінка уроків та досвіду, набутого під час реалізації проекту, сприяє виявленню ефективних стратегій та підходів.

Фінансовий звіт: Аналізуємо фінансову ефективність проекту та готуємо відповідний звіт. Підготовка фінансового звіту включає в себе врахування всіх витрат, видатків та бюджету проекту. Такий звіт є ключовим елементом для розуміння фінансової стійкості та продуктивності проекту.

Документація та архівація: Забезпечуємо збереження повної документації проекту та архівацію важливої інформації. Це включає закріплення за архівом всіх важливих документів, які стосуються проекту. Забезпечення легкого доступу до інформації дозволяє ефективно використовувати її у майбутньому.

Закриття команди проекту: Організація процесу формального закриття команди включає в себе подяку та визнання зусиль членів команди. Визначаємо подальші завдання для членів команди та створюємо план для їх реалізації.

Звітність перед замовником: Готуємо звіт та результати для представлення замовнику. Висвітлюємо виконані роботи, досягнення та отримані результати. Забезпечуємо відповіді на питання та уточнення від замовника.

Оцінка якості: Визначаємо відповідність результатів проекту встановленим стандартам розробки. Проводимо оцінку якості як віддільних елементів, так і загальної продукції проекту.

Анкетування та інтерв'ю: Проводимо анкетування та інтерв'ю для визначення задоволеності членів команди та інших учасників. Аналізуємо отримані відгуки та враховуємо їх у майбутніх проектах.

Підготовка до підтримки та розвитку: Розробляємо рекомендації для підтримки та майбутнього розвитку. Визначаємо потреби у підтримці та можливості майбутнього розширення продукту. Розроблюємо документацію та інструкції для команди підтримки [24].

4.4. Експлуатація та технічне обслуговування додатку

Етап експлуатації та технічного обслуговування включає в себе період після впровадження продукту чи системи, коли вони використовуються в реальних умовах. Під час цього етапу проводяться дії з забезпечення стабільності та надійності продукту, вирішення проблем, які можуть виникнути під час експлуатації, а також удосконалення та розвиток продукту для відповіді на змінні потреби користувачів і ринку. Технічне обслуговування включає в себе регулярні перевірки, моніторинг продуктивності, вирішення інцидентів та випуск оновлень для покращення функціоналу та безпеки продукту, а саме:

1. Регулярні перевірки:

- Проведення періодичних технічних оглядів для визначення стану компонентів та обладнання.
- Аналіз та оцінка стану інфраструктури для запобігання можливим поломкам.

2. Вирішення інцидентів:

- Швидка і ефективна реакція на інциденти та непередбачені ситуації, що можуть виникнути під час експлуатації.
- Розробка та виконання планів аварійного відновлення для забезпечення найменшого впливу на користувачів.

3. Випуск оновлень:

- Розробка та випуск регулярних оновлень для усунення помилок, вдосконалення функціоналу та забезпечення безпеки.
- Систематичне тестування нових версій перед їхнім випуском для запобігання можливим проблемам.

4. *Забезпечення безпеки:*

- Здійснення проактивних заходів для захисту продукту від потенційних загроз безпеки.
- Постійне вдосконалення заходів забезпечення та реагування на нові виклики у сфері кібербезпеки.

5. *Оптимізація ресурсів:*

- Аналіз використання ресурсів для оптимізації ефективності та витрат.
- Вдосконалення параметрів налаштування системи для забезпечення оптимального функціонування.

6. *Документація та звітність:*

- Підтримка актуальної документації, яка включає в себе інструкції з обслуговування та вирішення проблем.
- Формування звітів щодо роботи системи, інцидентів та заходів, які були вжиті.

Ці дії спрямовані на забезпечення стабільності, надійності та безпеки продукту, а також на відповідь на змінні потреби користувачів і ринку через його постійне удосконалення та розвиток [55].

4.4.1. Моніторинг ефективності

На етапі експлуатації відводимо особливу увагу постійному моніторингу роботи системи з метою виявлення можливих проблем та забезпечення її ефективності.

Моніторинг ефективності: Забезпечуємо постійний контроль за функціонуванням системи, використовуючи різноманітні інструменти та техніки моніторингу. Слідкуємо за продуктивністю, завантаженням ресурсів та взаємодією з користувачами для вчасного виявлення можливих проблем. Проводимо аналіз зібраних даних, щоб виявити та виправити потенційні проблеми та оцінити ефективність системи для оптимального функціонування. Крім того, моніторинг є важливою частиною процесу підтримки, дозволяючи оперативно реагувати на зміни в середовищі або

потреби користувачів, а також вдосконалювати систему відповідно до нових вимог та технологічних стандартів.

Виявлення можливих проблем: Аналізуємо дані моніторингу з метою оперативного виявлення можливих проблем чи відхилень від стандартів. Розробляємо стратегії для своєчасного реагування на виниклі ситуації та усунення неполадок. Цей процес включає в себе не лише виявлення проблем, але й аналіз їхніх кореневих причин для вдосконалення стійкості та надійності системи. Крім того, розробляються плани та процедури відновлення, які дозволяють ефективно та оперативно відновити роботу системи в найкоротший термін після виникнення проблем. Цей підхід спрямований на максимальне зменшення впливу негативних ситуацій на продуктивність та функціональність інформаційної системи.

Реагування на проблеми: В разі виявлення проблем проводимо оперативні заходи для їхнього усунення. Використовуємо системи автоматизованого сповіщення та забезпечуємо швидку реакцію технічного персоналу на сигнали системи. Наш підхід включає в себе створення необхідних заходів для стабілізації системи та відновлення нормального функціонування. Крім того, розробляються плани ескалації, що визначають послідовність кроків для реагування на проблеми різного рівня складності. Використання автоматизованих засобів сповіщення та систем моніторингу дозволяє нам оперативно реагувати на критичні ситуації та забезпечує найвищий рівень доступності та надійності інформаційної системи.

Оптимізація роботи системи: Впроваджуємо заходи для оптимізації роботи системи, забезпечуючи швидкий відгук на виявлені проблеми та впроваджуючи поліпшення для підвищення ефективності. Наш підхід орієнтований на постійне вдосконалення процесів та функціональності системи. Використовуючи аналітичні дані та звіти з моніторингу, розробляємо стратегії оптимізації, які враховують особливості роботи системи та вимоги користувачів. Проводимо регулярні аудити для виявлення можливостей для вдосконалення архітектури системи та оптимізації ресурсів. Цей процес

спрямований на забезпечення стабільності, високої продуктивності та задоволення потреб користувачів у найбільш ефективний спосіб.

Планові обслуговувальні роботи: Визначаємо графік планових обслуговувальних робіт для підтримки інтегритету системи та попередження можливих проблем у майбутньому. Ретельно документуємо проведені заходи та їхні результати, щоб мати повний огляд стану системи та історію проведених операцій. Під час обслуговування системи впроваджуємо необхідні оновлення та патчі для забезпечення безпеки та стабільності. Також розглядаємо можливості вдосконалення системи, щоб врахувати зміни в бізнес-вимогах та технологічному оточенні. Цей цикл планового обслуговування допомагає забезпечити тривале та ефективне функціонування системи протягом її життєвого циклу [41].

4.4.2. Вдосконалення та оновлення

На етапі вдосконалення та оновлення системи акцентуємо увагу на постійному покращенні та адаптації продукту до змінних умов та вимог користувачів.

Вдосконалення продукту: Аналізуємо звіти про ефективність системи та здійснюємо ретельний огляд відгуків користувачів. На основі цих даних визначаємо напрямки вдосконалення, спрямовані на оптимізацію функціональності та вдосконалення користувацького досвіду. Шляхом постійної взаємодії з користувачами та врахування їхніх потреб, ми активно впроваджуємо інноваційні рішення та нові функціональні можливості для забезпечення високоякісного продукту, який відповідає сучасним стандартам та очікуванням користувачів.

Оновлення системи: Вносимо зміни та оновлення до програмного забезпечення, спрямовані на виправлення виявлених помилок, підвищення безпеки, а також розширення або покращення існуючого функціоналу. Цей процес включає в себе тестування оновлень перед їхнім впровадженням, щоб гарантувати стабільність та сумісність з іншими компонентами системи. Крім того, ми надаємо користувачам інформацію про нові можливості та

поліпшення, забезпечуючи плавний перехід до оновленого продукту та максимальну задоволеність від його використання.

Адаптація до змін вимог користувачів: Вивчаємо зміни в потребах та вимогах користувачів, а також враховуємо ринкові тенденції. Здійснюємо адаптацію продукту для відповіді на змінні умови та підтримки нових функціональних вимог. Шляхом неперервного моніторингу реакції користувачів та залучення їхнього відгуку ми впроваджуємо необхідні зміни та інновації, щоб наш продукт був завжди відповідальний на високий стандарт якості та відповідав сучасним вимогам ринку.

Контроль якості оновлень: Забезпечуємо високий стандарт якості під час впровадження оновлень, проводимо тестування для перевірки стабільності та сумісності. Ретельно моніторимо реакцію користувачів на внесені зміни. Зібрані відгуки та додаткові висновки від користування системою допомагають удосконалити процеси впровадження оновлень та надати користувачам найкращий досвід використання оновленого продукту.

Комерційна ефективність: Враховуємо економічний аспект оновлень, забезпечуючи їхню комерційну ефективність та визначаючи можливість отримання прибуткових результатів від внесених змін. Аналізуємо вартість впровадження оновлень порівняно з очікуваними користями, щоб максимізувати віддачу від інвестицій та забезпечити бізнес-цілі та стратегічні плани компанії.

Стратегія впровадження: Розробляємо стратегію впровадження оновлень, враховуючи інтереси користувачів та максимізуючи ефективність процесу оновлення. Попередньо інформуємо користувачів про планові зміни та їхні переваги, щоб забезпечити їхню готовність до нововведень та максимально сприяти позитивному прийняттю оновленого продукту [9].

ВИСНОВКИ

Дослідження, проведене у рамках даної кваліфікаційної роботи, дозволило вивчити та оцінити характеристики проекту створення додатку для торгівлі криптовалютою на платформі Android. Робота включала в себе аналіз ринку криптовалют, розробку та реалізацію мобільного додатку з використанням технологій Android, а також оцінку його функціональності та ефективності.

Одним із ключових висновків є те, що ринок криптовалют є динамічним та зростаючим. Інтерес і попит на інструменти для торгівлі криптовалютою значно зросли в останні роки, що створює перспективи для успішної реалізації подібного додатку.

Дослідження також надало важливий внесок у розуміння динаміки ринку криптовалют та технологічних аспектів розробки фінансових додатків на платформі Android. Отримані знання можуть бути використані як цінний ресурс для прийняття обґрунтованих рішень у сфері інформаційних технологій та фінансового сектору.

Розроблений додаток для торгівлі криптовалютою на платформі Android демонструє високий рівень технічної реалізації та інноваційних функцій. Використані технології дозволяють забезпечити надійність та безпеку операцій, що є важливим аспектом у сфері фінансів та торгівлі.

Оцінка функціональності додатку підтвердила його здатність забезпечувати зручний та ефективний інтерфейс для користувачів. Урахування особливостей ринку криптовалют та потреб його учасників дозволило визначити ключові функції додатку, які сприятимуть підвищенню його конкурентоспроможності.

Наступним важливим аспектом є питання безпеки. Враховуючи чутливість фінансових операцій та використання криптографії у сфері криптовалют, додаток був ретельно протестований на вразливості та має високий рівень захисту даних користувачів.

У процесі дослідження також було виявлено кілька перспективних напрямків для подальшого розвитку проекту. Зокрема, можливість інтеграції з іншими фінансовими інструментами, розширення функціональності для аналізу ринку, а також підтримка нових криптовалют та токенів.

Одним із ключових внесків роботи у властивості системи управління проектом є визначення оптимальних етапів планування, розробки та впровадження проекту. Розроблена стратегія впровадження оновлень та управління процесом розробки додатку може служити ефективним шаблоном для подібних ініціатив у майбутньому.

У цілому, результати дослідження свідчать про високий потенціал та перспективи подальшого розвитку додатку для торгівлі криптовалютою на платформі Android. Розроблена стратегія впровадження та управління проектом може служити моделлю для подібних інноваційних ініціатив, сприяючи підвищенню комерційної ефективності та задоволенню потреб користувачів.

СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ





















1. Азбука управління проєктами. Планування / О.В.Єгорченков, Н.Ю.Єгорченкова, Є.Ю.Катаєва. — Київ, 2016. — 226 стр
2. Стаття, скорочена версія вебінару з приводу Управління проєктами [Електронний ресурс] – 2021 – Режим доступу до ресурсу: <https://www.manage.gov.in/studymaterial/PM.pdf>
3. MVVM на Android з компонентами архітектури та бібліотека Kotlin [Електронний ресурс] – 2020 – Режим доступу до ресурсу: <https://medium.com/nuances-of-programming/mvvm-на-android-скомпонентами-архитектуры-библиотека-koin-e2e77b77950e>
4. Antonopoulos, A. M. (2014). "Mastering Bitcoin: Unlocking Digital Cryptocurrencies." O'Reilly Media.
5. "Проектне управління в інформаційних технологіях" - Гребенніков Д.В. - Київ, 2017.
6. "Ефективний проєкт: від ідеї до реалізації" - Коваль О.В., Мельник О.П. - Львів, 2016.
7. "Інформаційні технології в управлінні проєктами" - Мельничук О.М. - Київ, 2015.
8. "Android Programming: Pushing the Limits" - Mednieks, Zigurd, et al. - Wiley, 2013.
9. "Android Application Development All-in-One For Dummies" - Burd, Barry. - For Dummies, 2019.
10. Морозов В.В., Чумаченко І.В., Доценко Н.В., Чередніченко А.М. Управління проєктами: планування проєктних дій: Підручник, – К.: Університет економіки та права «КРОК», 2014.
11. Tapscott, D., & Tapscott, A. (2016). "Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world." Penguin.

12. "Проектний менеджмент в сучасних умовах" - Сидоренко В.П. - Харків, 2018.
13. "Управління проектами в галузі програмної інженерії" - Забігайло І.М., Головатий О.В. - Київ, 2019.
14. "Стратегічне управління проектами" - Яковлев Є.В. - Львів, 2017.
15. "Моделі та методи управління ризиками проектів" - Григор'єва І.В. - Київ, 2014.
16. "ІТ-проекти: організація та управління" - Коваленко І.І. - Харків, 2016.
17. Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction." Princeton University Press.
18. Swan, M. (2015). "Blockchain: blueprint for a new economy." O'Reilly Media.
19. "Android Programming: Pushing the Limits" - Mednieks, Zigurd, et al. - Wiley, 2013.
20. "Android Application Development All-in-One For Dummies" - Burd, Barry. - For Dummies, 2019.
21. Морозов В.В. Основи закупівель товарів, робіт та послуг а проектах: Навчальний посібник. – К.: Таксон, 2003.
22. Casey, M. J., & Vigna, P. (2018). "The Truth Machine: The Blockchain and the Future of Everything." St. Martin's Press.
23. Mougayar, W. (2016). "The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology." John Wiley & Sons.
24. "Android Developer Documentation." developer.android.com.
25. "Проектна діяльність в умовах нестабільності" - Нестеров О.М. - Київ, 2018.
26. "Android Programming: Pushing the Limits" - Mednieks, Zigurd, et al. - Wiley, 2013.
27. "Актуальні аспекти управління проектами в інформаційній сфері" - Дмитрієва І.О., Макарова Н.В. - Харків, 2017.

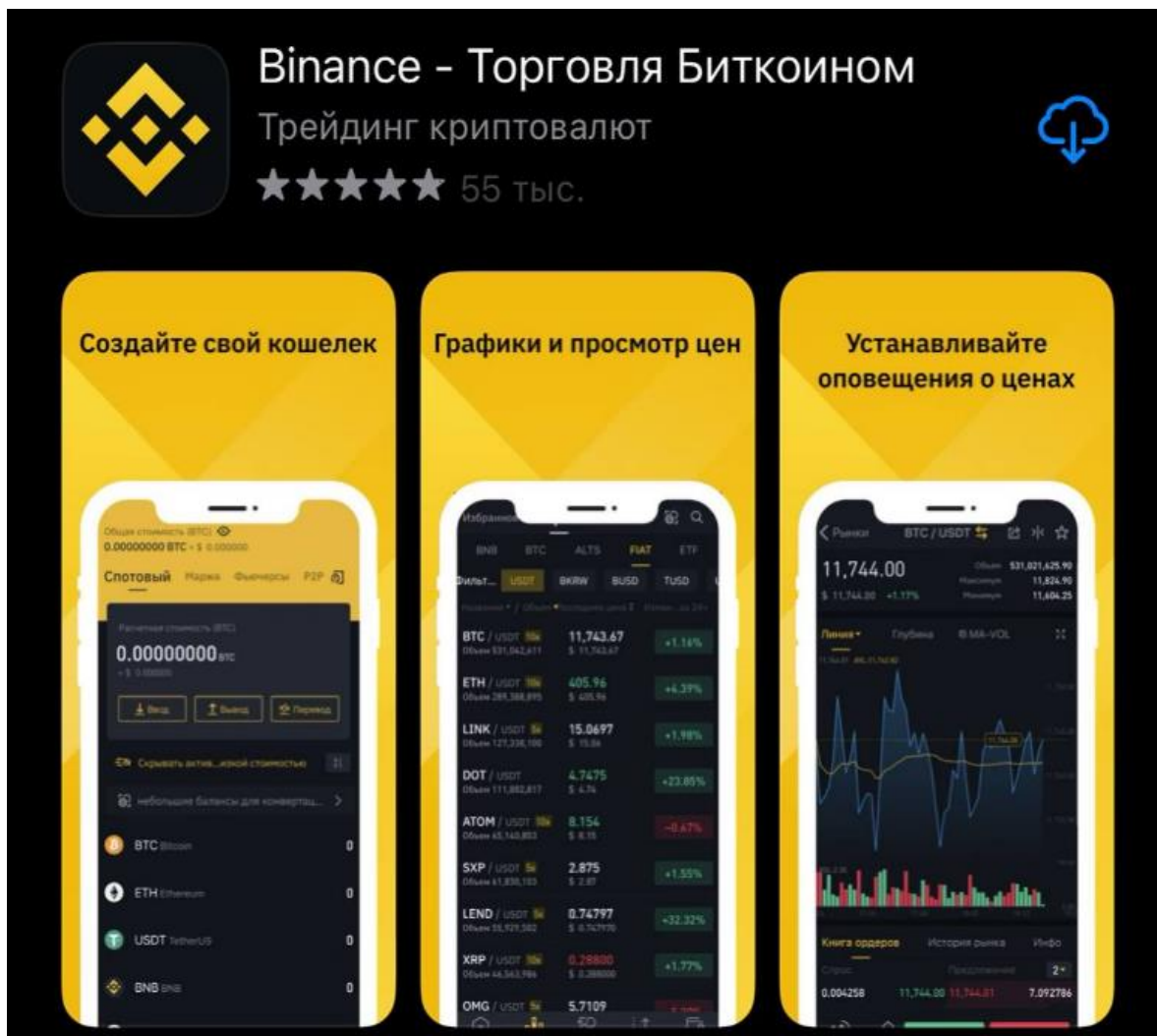
28. "Проектний підхід у розробці програмного забезпечення" - Шевченко В.М. - Київ, 2015.
29. "Управління розвитком IT-проектів" - Поляков Д.О. - Львів, 2019.
30. "Сучасні тенденції управління проектами в індустрії 4.0" - Білий В.І., Сидоров В.О. - Київ, 2020.
31. "Інноваційні підходи в управлінні проектами" - Шевченко А.С., Гордієнко О.І. - Львів, 2018.
32. Rosic, A. (2017). "Cryptocurrency: How Bitcoin and Digital Money are Challenging the Global Economic Order." Ameer Rosic.
33. Buterin, V. (2013). "Ethereum: A Next-Generation Cryptocurrency and Decentralized Application Platform." ethereum.org.
34. Морозов В.В., Чередніченко А.М., Шпильова Т.Ш. Формування, управління та розвиток команди проекту (поведінкової компетенції): Навчальний посібник. – К.: Таксон, 2009.
35. "Ефективний контроль у проектах з розробки ПЗ" - Галушко І.М. - Харків, 2016.
36. "Моделі взаємодії команд в IT-проектах" - Семенов Д.В., Кулик В.С. - Київ, 2017.
37. "Розробка та впровадження інноваційних проектів в сфері IT" - Трофимчук О.О., Колесніков В.В. - Львів, 2019.
38. "CoinMarketCap - Cryptocurrency Market Capitalizations." coinmarketcap.com.
39. "Android-програмування для професіоналів" - Девайтер, Кіт, і Джозеф Лоуренс. - Київ, 2014.
40. "Розробка Android-застосунків для початківців" - Доусон, Джейсон. - Київ, 2015.
41. "Binance: Cryptocurrency Exchange." binance.com.
42. Морозов В.В., Кузнецов Є.А. Прийняття проектних рішень в управлінні проектами: Навчальний посібник. – К.: Університет економіки та права «КРОК», 2011.

43. "Android Programming: Pushing the Limits" - Mednieks, Zigurd, et al. - Wiley, 2013.
44. "Android Application Development All-in-One For Dummies" - Burd, Barry. - For Dummies, 2019.
45. "Стратегії управління розвитком IT-проектів" - Петренко С.О., Михальченко Д.М. - Київ, 2018.
46. "Kraken: Buy, Sell, and Margin Trade Bitcoin (BTC) and Ethereum (ETH)." kraken.com.
47. "Blockchain.info: The World's Most Popular Way to Buy, Hold, and Use Crypto." blockchain.com.
48. "Стратегії управління розвитком IT-проектів" - Петренко С.О., Михальченко Д.М. - Київ, 2018.
49. "CoinDesk - Leader in Crypto and Blockchain News." coindesk.com.
50. "The Bitcoin Whitepaper." bitcoin.org.
51. Морозов В.В., Данченко О.Б. Інформаційні системи та технології в управлінні проектами: Навчальний посібник. – К.: Університет економіки та права «КРОК», 2011.
52. "Інструменти проектного менеджменту для реалізації інновацій в IT" - Денисов В.І., Черніченко О.О. - Харків, 2017.
53. "Оптимізація процесів управління проектами в галузі IT-консалтингу" - Горбач В.П., Кравченко Л.А. - Київ, 2015.
54. Морозов В.В., Хандрік О.В., Коломієць А.С. Управління проектами розвитку IT організацій: Навчальний посібник. –К.:ВПЦ «Київський університет», 2020.
55. "Android Programming: Pushing the Limits" - Mednieks, Zigurd, et al. - Wiley, 2013.
56. "Android Application Development All-in-One For Dummies" - Burd, Barry. - For Dummies, 2019.

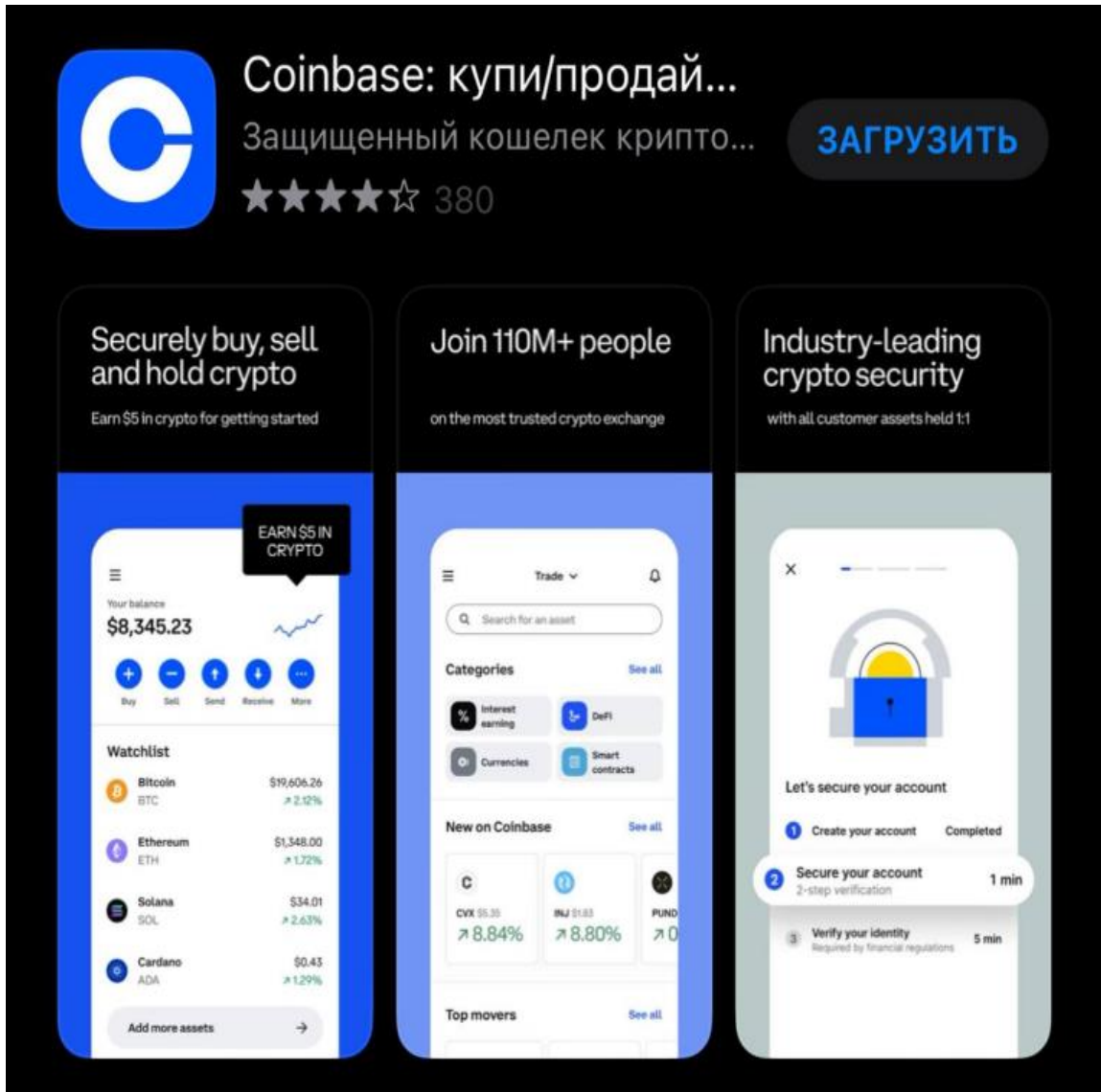
ДОДАТОК А. Рис. 1.1. Рейтинг криптовалют станом на 2023р.

#	МОНЕТА	ЦІНА	КАПІТАЛІЗАЦІЯ	ВИПУЩЕНО	ОБ'ЄМ ЗА 24Ч	ПОПРАВКА 24 ГОДИНИ	ГРАФІК ЦІНИ (30Д)
1	 Bitcoin BTC	41 717.6\$	816 053 004 571\$	19 561 362 BTC	37 885 126 998\$	+2.55% +1 063.80\$	
2	 Ethereum ETH	2 227.82\$	267 851 686 736\$	120 230 295 ETH	15 757 370 421\$	+0.67% +14.93 \$	
3	 Прив'язати USDT	0.999755 \$	89 792 109 359\$	89 814 116 890 доларів США	73 750 270 434\$	-0.06% -	
4	 Binance Coin BNB	231.345 \$	35 094 598 185\$	151 698 076 БНБ	2 140 911 254\$	+0.88% +2.04\$	
5	 XRP XRP	0.619037\$	33 358 999 156\$	53 888 571 585 XRP	1 954 973 148\$	-1.65% -0.01 \$	
6	 Солана СОНЦЕ	60.1698\$	25 533 151 357\$	424 351 910 SOL	2 207 021 361\$	-5.84% -3.51 \$	
7	 Монета доларів ... USDC	0.999763 \$	24 368 314 462\$	24 374 081 974 доларів США	7 211 062 729\$	-0.02% -	
8	 Кардано Є	0.403795 \$	14 259 030 893\$	35 312 508 598 ADA	540 586 166\$	-0.27% -	
9	 Dogecoin ДОЖ	0.090083 \$	12 801 028 831\$	142 103 066 383 DOGE	1 407 234 606\$	+4.16% -	
10	 Трон TRX	0.103408\$	9 152 521 659\$	88 509 215 013 TRX	328 235 560\$	-0.23% -	

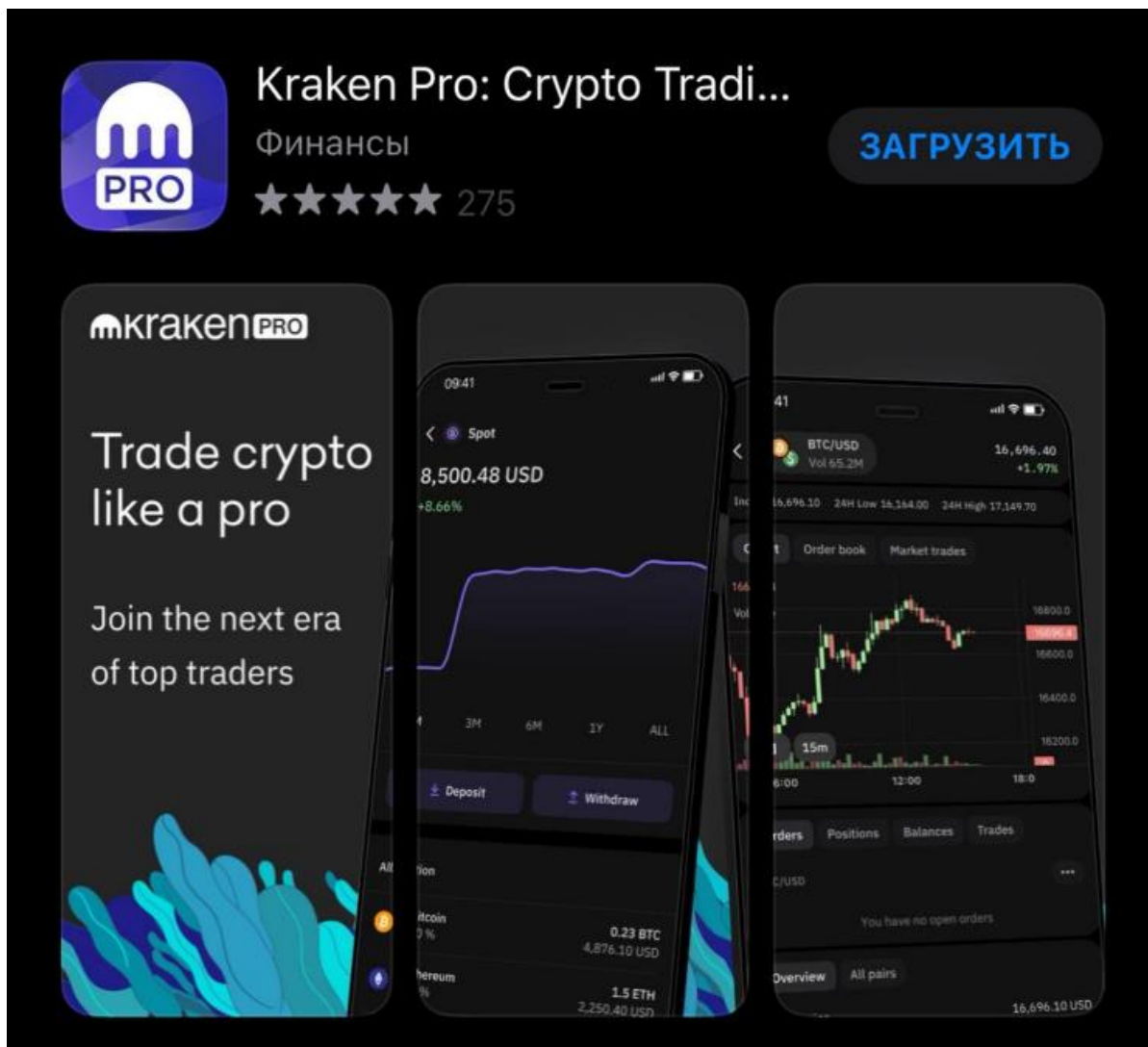
ДОДАТОК Б. Рис. 1.2. Мобільний додаток «Binance»



ДОДАТОК В. Рис. 1.3. Мобільний додаток «Coinbase»



ДОДАТОК Г. Рис. 1.4. Мобільний додаток «Kraken»



ДОДАТОК Д. Таблиця 2.1. Паспорт проекту

№	Пункт	Опис
1	2	3
1.	Назва проекту	Дослідження характеристик проекту створення додатку для торгівлі криптовалютою на платформі Android
2.	Мета проекту	Метою проекту є розробка та реалізація андроїд додатку, який забезпечує користувачам зручну та безпечну платформу для торгівлі криптовалютою на пристроях з операційною системою Android. Додаток повинен надавати широкий функціонал для торгівлі, забезпечувати надійність, швидкість та конфіденційність операцій, а також забезпечувати користувачів необхідною інформацією про курси криптовалют та тренди ринку.
3.	Основні задачі проекту	<ul style="list-style-type: none"> – Аналіз вимог та визначення функціональності додатку. – Розробка інтерфейсу користувача та реалізація основних функцій. – Інтеграція із зовнішніми API для отримання актуальних даних. – Розробка механізмів безпеки та захисту даних. – Тестування та налагодження додатку. – Впровадження та підтримка додатку.

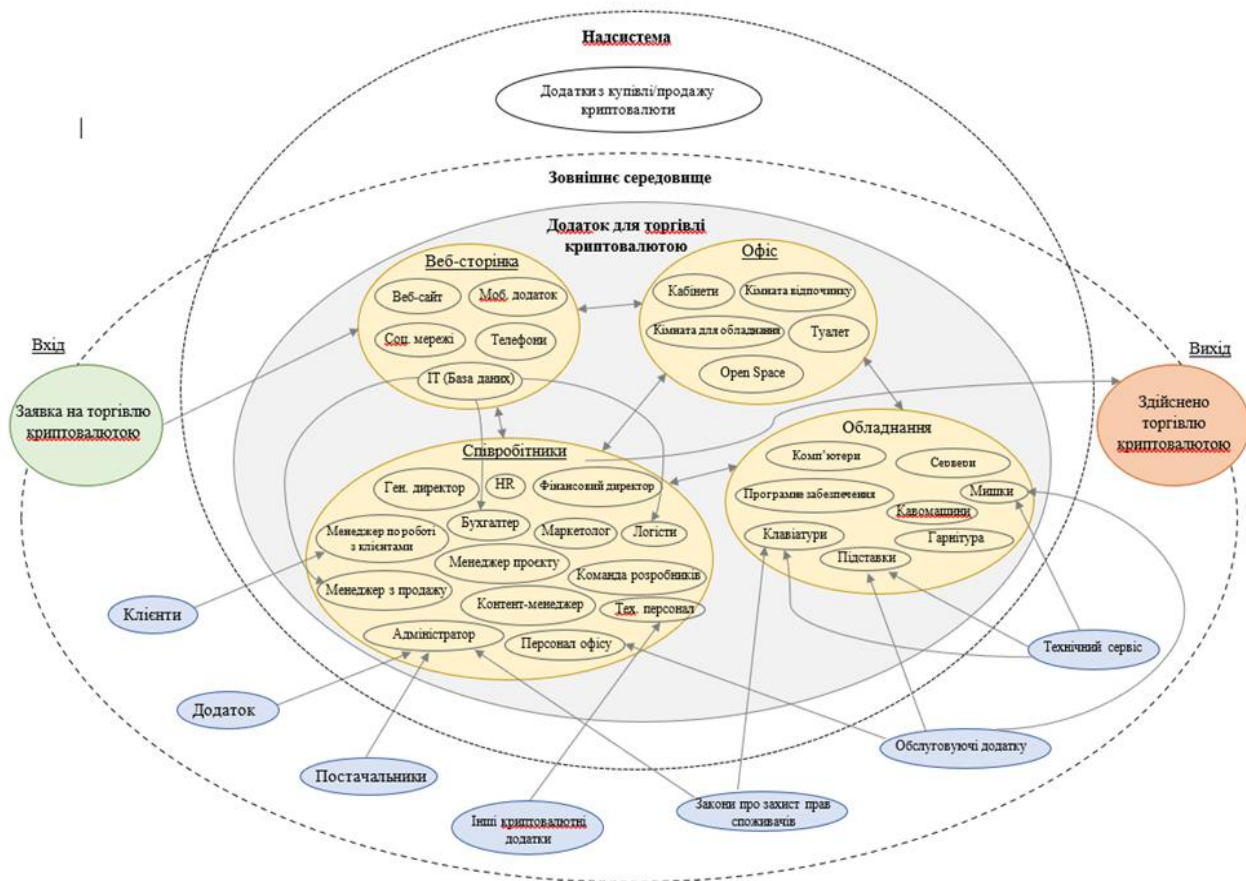
Продовження табл. 2.1.

1	2	3
4.	Очікувані результати	<ul style="list-style-type: none"> – Функціональний та стабільний андроїд додаток для торгівлі криптовалютою на платформі Android. – Забезпечення безпеки та конфіденційності даних користувачів. – Зручний та інтуїтивно зрозумілий інтерфейс користувача. – Інтеграція із зовнішніми сервісами та API для отримання актуальних даних. – Підтримка різних платіжних методів та криптовалютних бірж. – Повна документація проекту, включаючи детальний опис функціональності та технічних аспектів.
5.	Обмеження проекту	<ul style="list-style-type: none"> – Бюджет проекту обмежений певною сумою. – Потрібно забезпечити сумісність додатка з мобільною платформою Android, що може обмежити вибір технологій або підходів до розробки. <p>Проект має бути завершений протягом 12 місяців.</p>
6.	Ключові зацікавлені сторони	<ul style="list-style-type: none"> – Замовники – Кінцеві користувачі – Криптовалютні біржі – Команда проекту додатку – Фінансові установи, аналітики

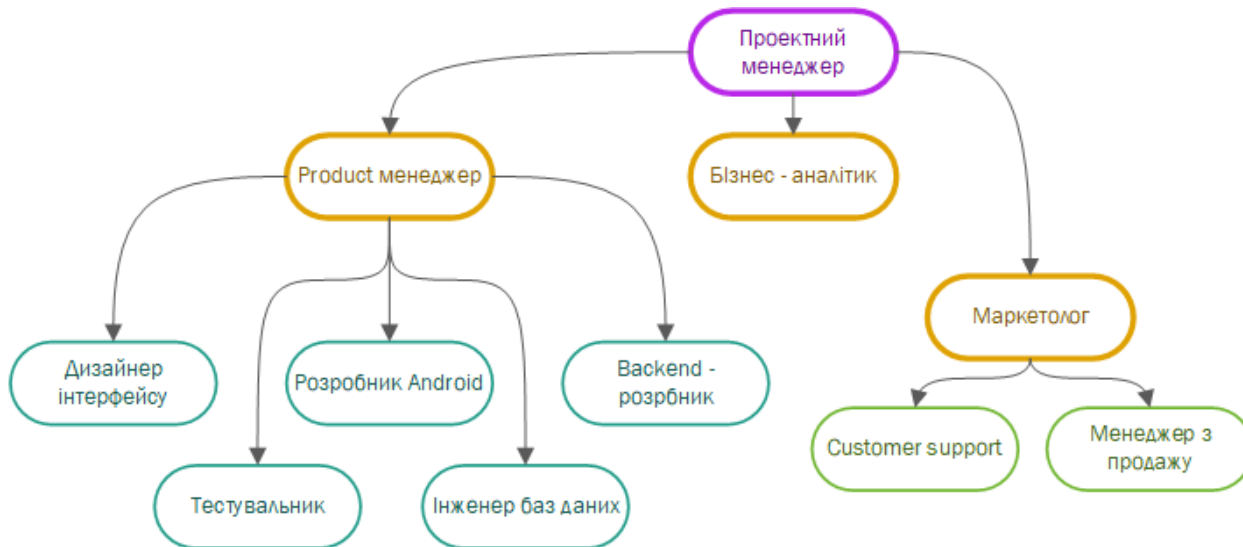
Продовження табл. 2.1.

1	2	3
7.	Основні етапи проекту	<ul style="list-style-type: none"> – Необхідно провести аналіз вимог та розробити концептуальну модель протягом 1-2 місяців. – Досягнення дизайну інтерфейсу та структури додатку потребує 2-3 місяців. – Розробка алгоритмів та програмного забезпечення займе від 3 до 6 місяців. – Проведення тестування додатку та внесення коректив займає період від 6 до 8 місяців. – Запуск додатку планується на протязі 8-12 місяців.
8.	Ризики	<ul style="list-style-type: none"> – Технічні ризики – Фінансові ризики – Ризики конкуренції – Правові ризики – Виклики з підтримки та обслуговування
9	Основні метрики успіху	<ul style="list-style-type: none"> – Кількість завантажень додатку – Кількість активних користувачів – Обсяг торгових операцій – Рівень задоволеності користувачів – Час використання додатку – Конверсія використання в операції – Задоволеність замовників

ДОДАТОК Е. Рис. 2.1. Концептуальна модель мобільного додатку



ДОДАТОК Ж. Рис. 2.2. Організаційна структура команди проекту



ДОДАТОК II. Таблиця 2.2. Бюджет проекту

Посада	Ціна (\$/міс)	Тривалість (міс)	Загальні витрати (\$)
1	2	3	4
Проектний менеджер	\$5000	12	60000\$
Бізнес-аналітик	\$4000	12	48000\$
Дизайнер інтерфейсу	\$3500	6	21000\$
Розробник Android	\$4500	12	54000\$
Бекенд-розробник	\$4500	12	54000\$
Інженер з баз даних	\$4000	12	48000\$
Тестувальник	\$3500	10	35000\$
Продакт менеджер	5000\$	12	60000\$
Маркетолог	4000\$	8	32000\$
Додаткові витрати			
Оренда офісного простору	2000\$	12	24000\$
Обладнання та програмне забезпечення	1050\$	12	12600\$
Реклама і маркетинг	800\$	12	9600\$

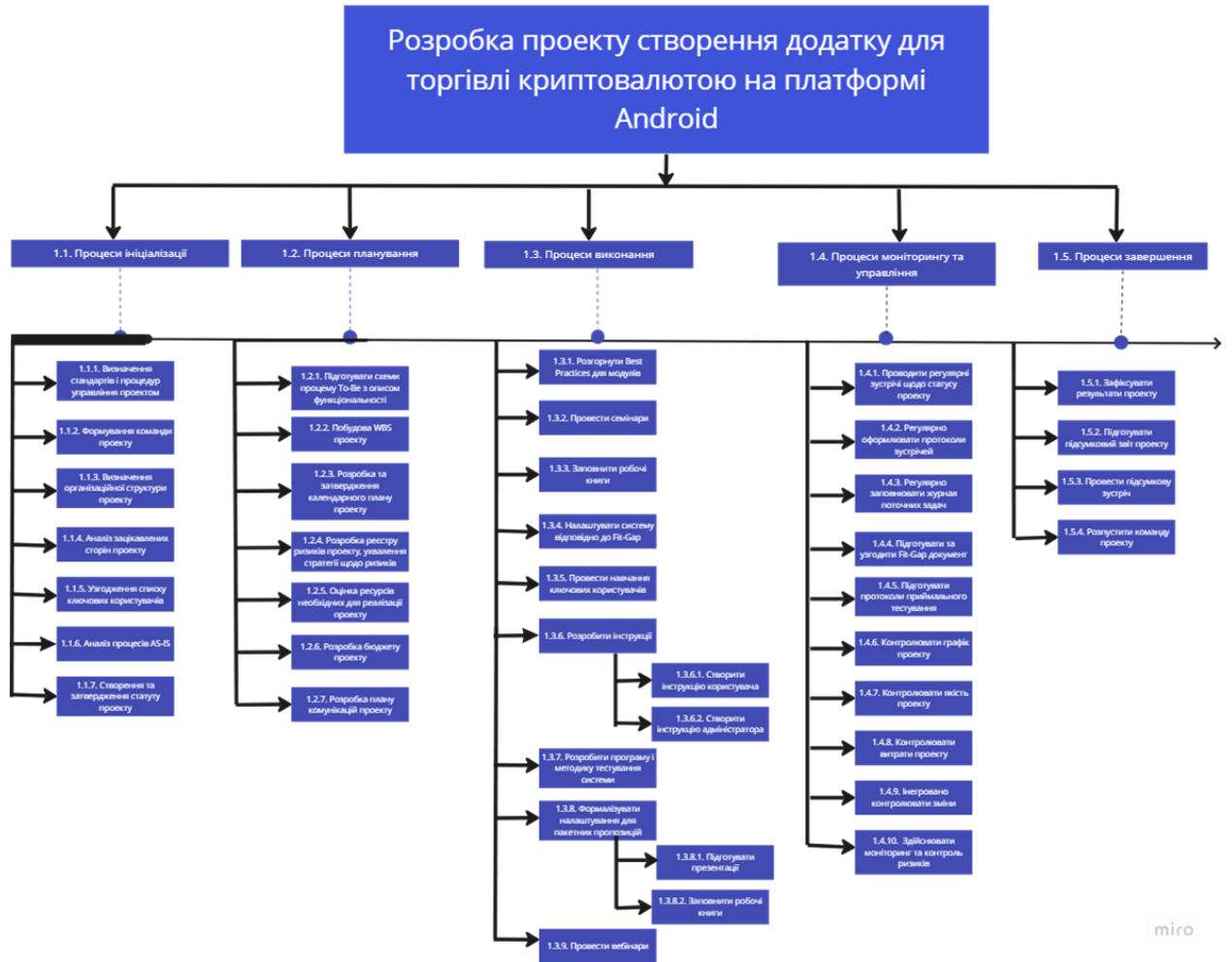
Продовження табл. 2.2.

1	2	3	4
Логістика та транспорт	500\$	12	6000\$
Витрати на управління ризиками	5000\$	12	60000\$
Правова підтримка	1000\$	12	12000\$
Інновації та дослідження	1000\$	10	10000\$
Загальні витрати			557000\$

ДОДАТОК К. Рис. 2.3. Діаграма вартості трудових ресурсів



ДОДАТОК Л. Рис. 2.4. WBS проекту по процесам



miro

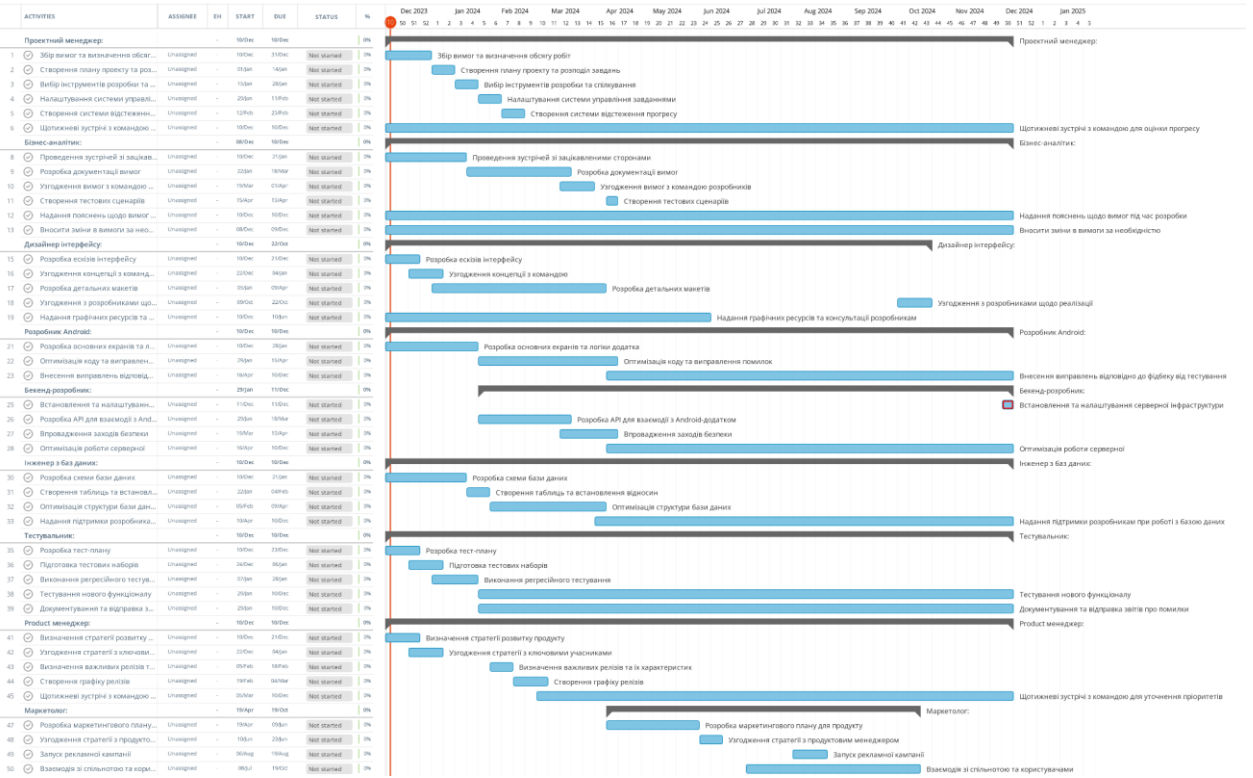
ДОДАТОК Н. Рис. 2.6. WBS проекту по продуктам



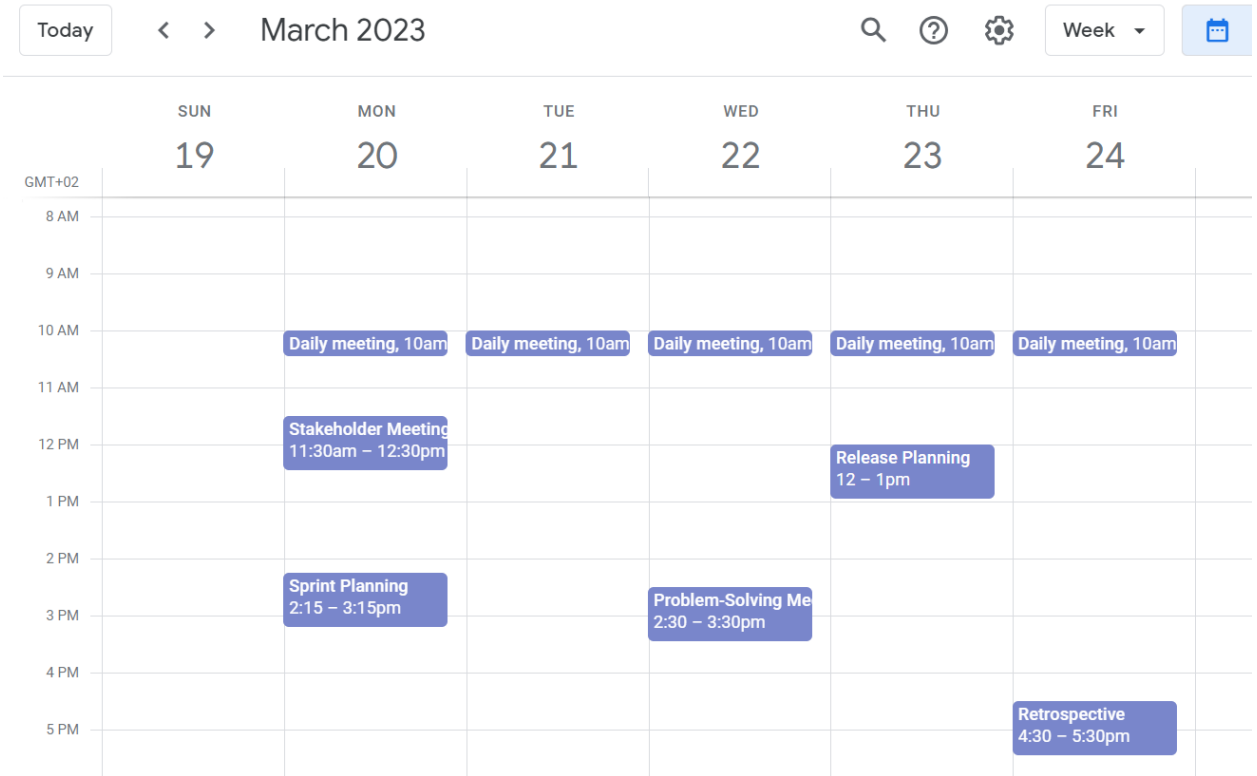
miro

ДОДАТОК П. Рис. 2.7. Діаграма Ганта

Android crypto app
Проектний менеджмент 10 лютого 2023



ДОДАТОК Р. Рис. 2.8. Приклад графіку проведення мітингів



ДОДАТОК С. Рис. 2.9. Ідентифікація ризиків проекту

№	Тип ризику	Ризикова подія	Сила впливу	Керованість
1	Програмні ризики	Виявлення критичного програмного багу	Висока	Середня
2		Неправильна інтеграція з API	Висока	Середня
3		Зміни в вимогах користувача	Середня	Низька
4		Затримка у випуску нової версії ОС	Низька	Низька
5	Апаратні ризики	Виход з ладу апаратного забезпечення	Висока	Висока
6		Недоступність необхідного обладнання	Висока	Середня
7		Втрата даних через дефект жорсткого диску	Середня	Середня
8		Проблеми зі стабільністю живлення	Висока	Низька
9	Внутрішні ризики проекту (команда)	Відсутність ключового члена команди	Висока	Середня
10		Порушення графіка роботи команди	Середня	Середня
11		Конфлікти в команді	Висока	Висока
12		Недостатня кваліфікація персоналу	Середня	Висока
13	Зовнішні (оточення)	Зміни в законодавстві про криптовалюту	Низька	Низька
14		Конкуренція на ринку криптовалют	Середня	Низька
15		Зміни у валютному ринку	Висока	Низька
16		Втрата доступу до платформи Google Play	Висока	Низька
17	Форс мажори	Природні катастрофи	Висока	Низька
18		Політичні кризи	Висока	Низька
19		Техногенні аварії	Висока	Низька
20		Збої в роботі інфраструктури	Середня	Низька
21	Ризики кібербезпеки	Вторгнення хакерів	Висока	Низька
22		Витік конфіденційної інформації	Висока	Низька
23		Спроби впливу на роботу додатку зловмисників	Висока	Низька
24		Недостатній захист даних користувачів	Середня	Низька

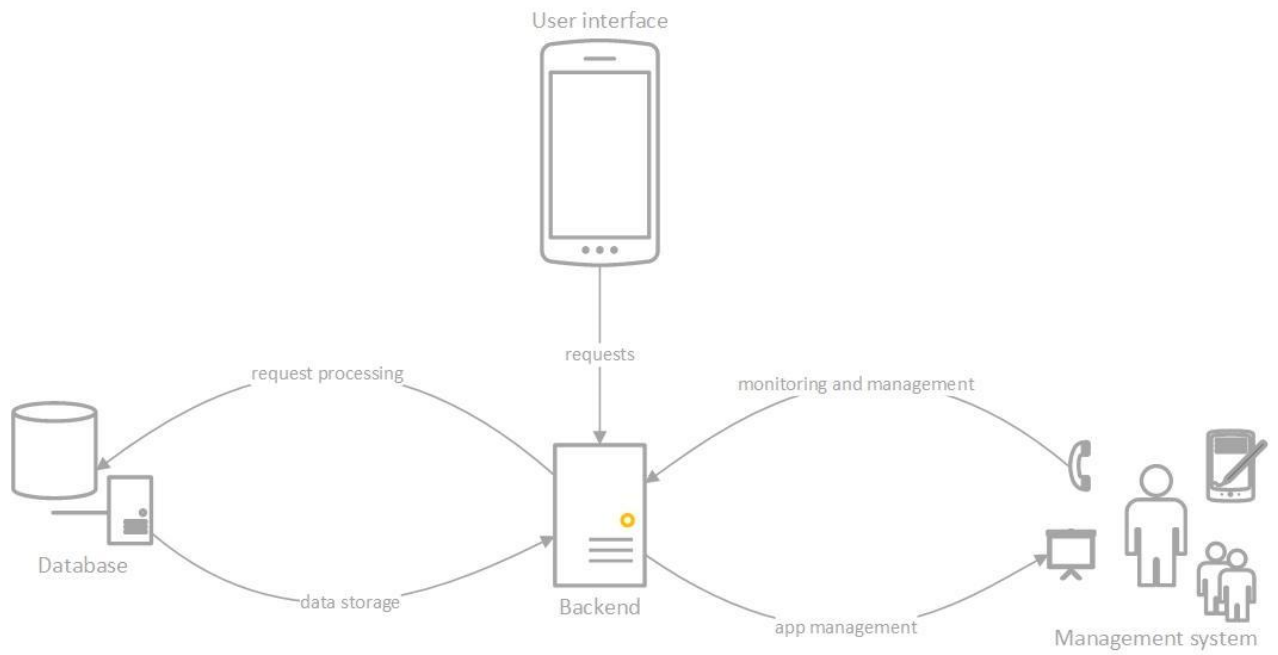
ДОДАТОК Т. Рис. 2.10. Оцінювання ризиків проекту

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовірність		Частота (за проект)		Важливість ризиків (компл. показник)
		Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.
1	Ризик подорожчання ліцензійного ПЗ	СВ	6	ВС	8	СН	4	СН	4	32
2	Ризик втрати ключового члена команди	СВ	7	ВС	8	СН	3	СН	3	24
3	Ризик зміни вимог до додатку	СС	5	ВС	7	СН	2	СН	2	14
4	Ризик недостатньої кваліфікації персоналу	СН	4	СС	6	СН	3	СС	3	18
5	Ризик виходу з ладу сервера	СН	8	СС	7	СС	2	СС	2	14
6	Ризик змін у законодавстві про криптовалюту	СС	6	СВ	7	СН	1	СН	1	7
7	Ризик конфліктів в команді	СВ	7	ВС	8	СВ	2	СВ	2	16
8	Ризик змін на фінансовому ринку	СС	6	СВ	7	СВ	1	СВ	1	7
9	Ризик неправильної інтеграції з API	СВ	7	СС	6	СС	3	СС	3	18
10	Ризик природної катастрофи	ВВ	9	СН	5	СН	1	СН	1	5
11	Ризик збоїв в роботі інфраструктури	СС	8	СС	7	СС	2	СС	2	14
12	Ризик вторгнення хакерів	СВ	8	ВВ	9	СС	3	СС	3	27
13	Ризик витоку конфіденційної інформації	СС	7	ВВ	8	СВ	2	СС	2	16
14	Ризик спроб впливу на роботу додатку зломисників	ВВ	8	ВВ	9	СВ	3	СС	3	27
15	Ризик політичної кризи	СВ	6	СН	5	ВС	1	СН	1	5
16	Ризик змін у валютному ринку	СС	7	СС	6	СС	2	СС	2	12
17	Ризик втрати доступу до платформи Google Play	СВ	7	ВВ	8	СС	3	СВ	3	24
18	Ризик техногенної аварії	СС	8	СС	7	СН	3	СС	3	21
19	Ризик недоступності необхідного обладнання	СН	5	СН	6	СН	1	СН	1	6
20	Ризик виходу з ладу апаратного забезпечення	СН	7	СН	8	СН	2	СН	2	16

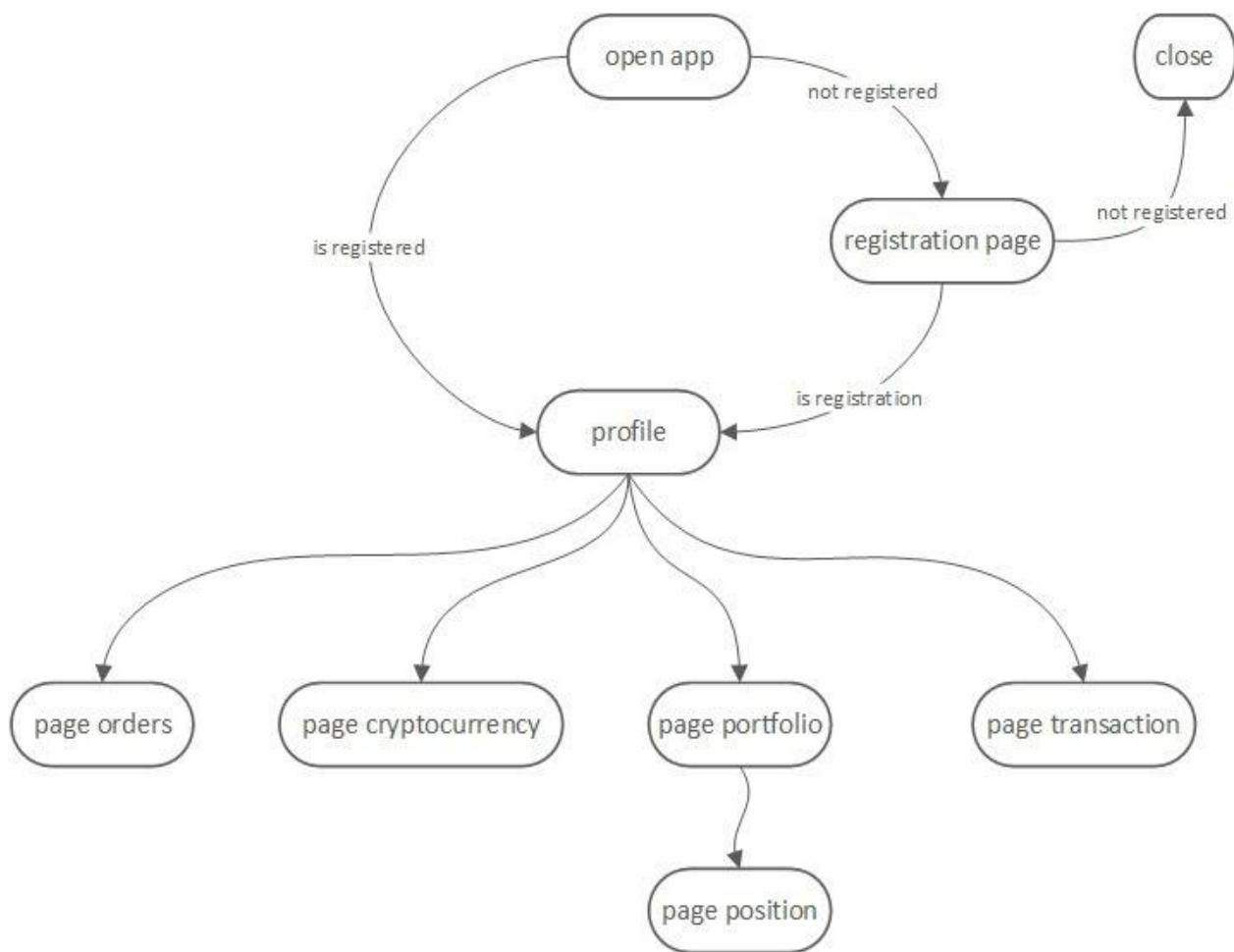
ДОДАТОК У. Рис. 2.11. Розробка протиризикових заходів (ПРЗ) проекту

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		профілактика		при симптомі	при проблемі
1	Ризик подорожчання ліцензійного ПЗ	Перевірка альтернативного програмного забезпечення та ліцензій на ринку	Збір інформації про доступні альтернативи	Розгляд можливості використання безкоштовного ПЗ	Спілкування з постачальником ліцензійного ПЗ
2	Ризик втрати ключового члена команди	Пошук і найм резервного персоналу	Моніторинг роботи членів команди	Проведення тренінгів та розвиток кадрів	Регулярні зустрічі з ключовим членом команди
3	Ризик зміни вимог до додатку	Постійний моніторинг вимог та оновлення документації з вимог	Аналіз вимог та змін до вимог	Регулярне оновлення документації з вимог	Заплановане оновлення функціональної специфікації
4	Ризик недостатньої кваліфікації персоналу	Планування та проведення навчання та розвитку персоналу	Визначення слабких місць у кваліфікації персоналу	Надання навчальних матеріалів і засобів для самостійного навчання	Регулярні оцінки кваліфікації персоналу

ДОДАТОК Ф. Рис. 3.1. Схема структури ПЗ додатку



ДОДАТОК Х. Рис. 3.2. Блок-схема алгоритму взаємодії користувача з додатком



ДОДАТОК Ц. Рис. 3.3. Вхід/реєстрація користувача

The image displays two mobile application screens for user authentication, set against a dark blue background.

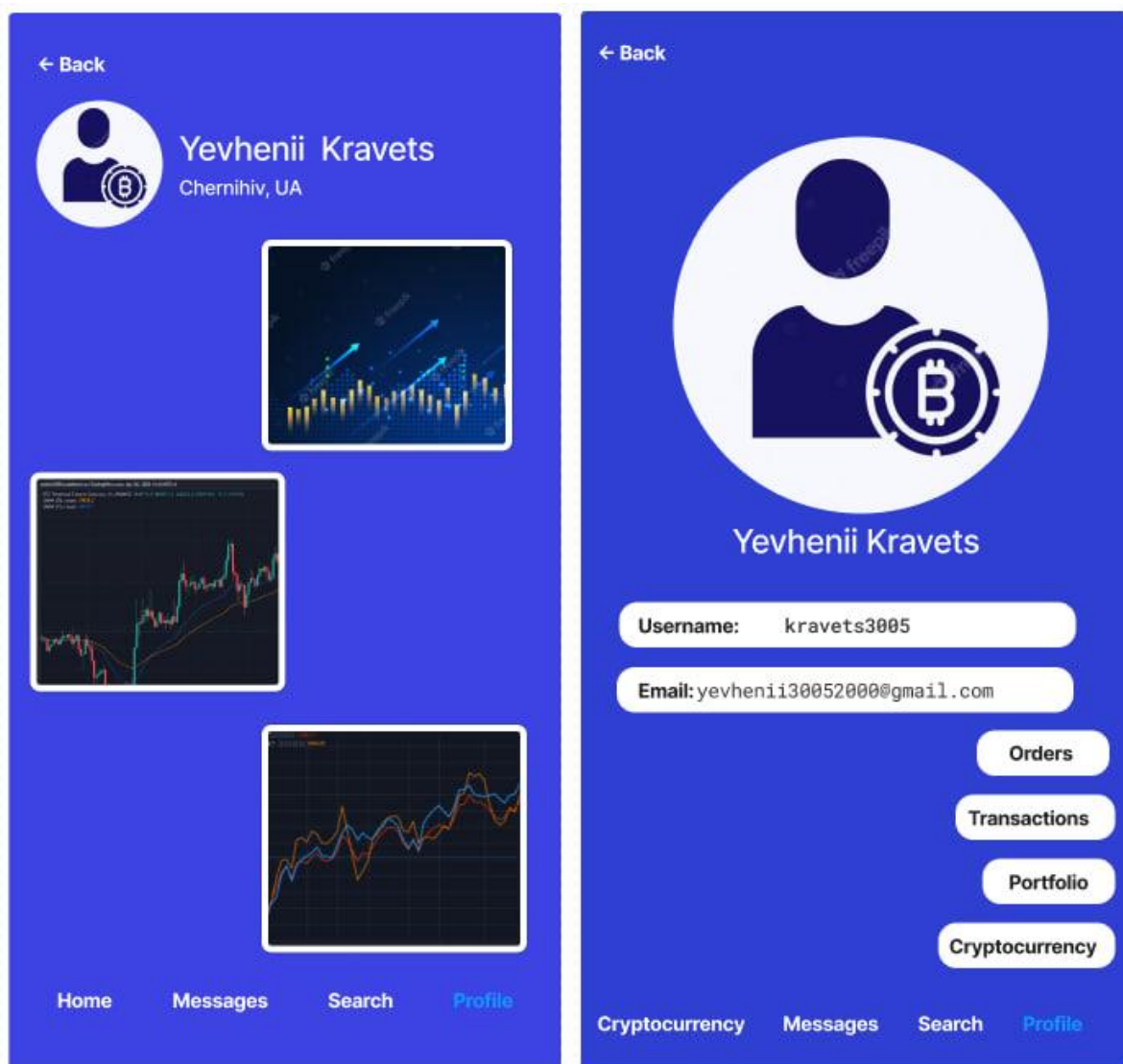
Left Screen (Login):

- Header: A circular icon containing a white silhouette of a person and a Bitcoin symbol.
- Title: "Log In" in white text.
- Form: Two white input fields. The first is labeled "Email" and contains "*****". The second is labeled "Password" and contains "*****".
- Action: A blue button labeled "Sign in".
- Links: "Forgot password" and "Register" in light blue text at the bottom.

Right Screen (Registration):

- Header: A back arrow and the text "Back" in white.
- Title: "Register" in white text.
- Form: Three white input fields. The first is labeled "username", the second "Email", and the third "Password".
- Form: A fourth white input field labeled "Re enter password".
- Image: A square image showing a Bitcoin coin on a server rack with glowing blue lines.
- Action: A blue button labeled "Sign up" at the bottom.

ДОДАТОК Ш. Рис. 3.4. Профіль користувача



ДОДАТОК Щ. Таблиця 3.1. Сутності бази даних

Сутність	Атрибути
1	2
Користувач (User)	<ul style="list-style-type: none"> – Ідентифікатор користувача (User ID) – Ім'я користувача (Username) – Дата реєстрації (Registration Date) – Баланс рахунку (Account Balance)
Криптовалюта (Cryptocurrency)	<ul style="list-style-type: none"> – Ідентифікатор криптовалюти (Cryptocurrency ID) – Назва (Name) – Скорочена назва (Symbol) – Поточна ціна (Current Price) – Кількість одиниць в обігу (Circulating Supply)
Замовлення (Order)	<ul style="list-style-type: none"> – Ідентифікатор замовлення (Order ID) – Ідентифікатор користувача (User ID) – Ідентифікатор криптовалюти (Cryptocurrency ID) – Тип замовлення (Order Type) – Кількість одиниць (Quantity) – Ціна замовлення (Order Price) – Дата замовлення (Order Date)

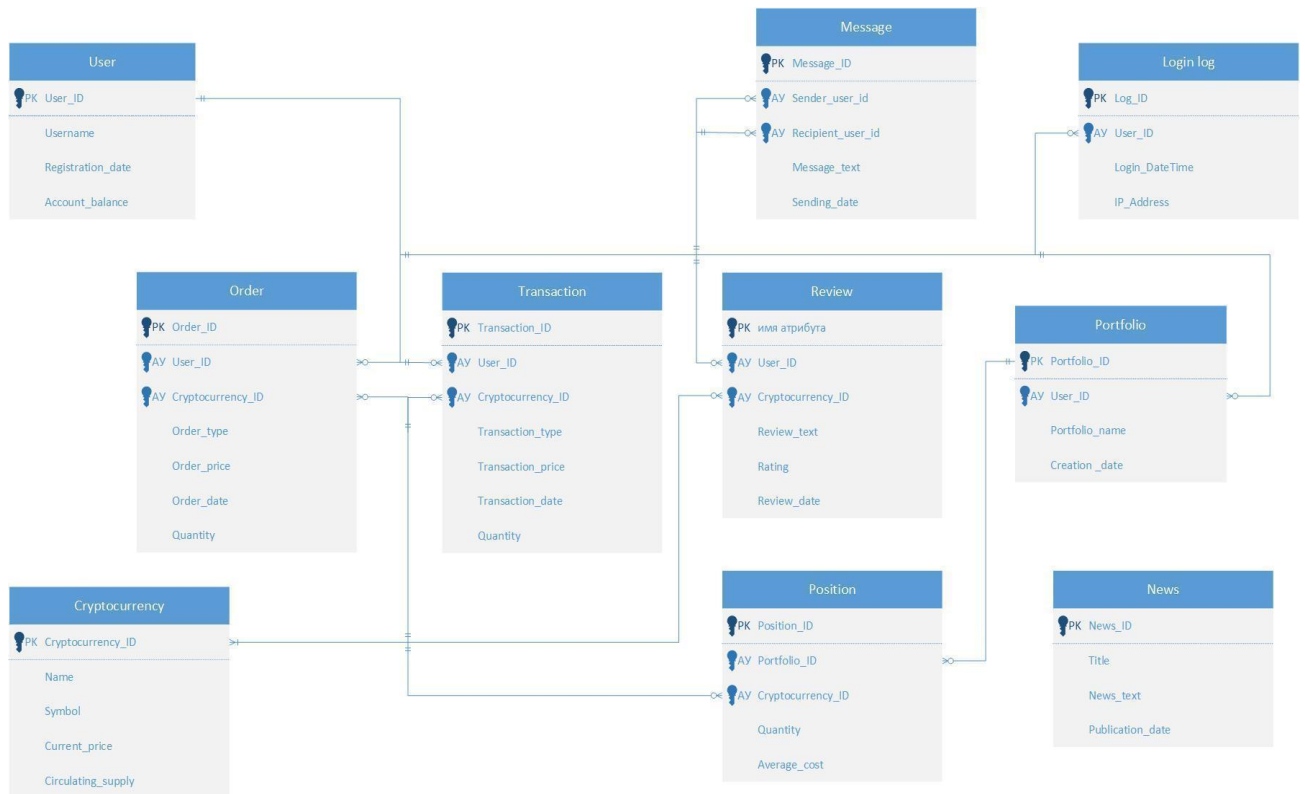
Продовження табл. 3.1.

1	2
Портфель (Portfolio)	<ul style="list-style-type: none"> – Ідентифікатор портфеля (Portfolio ID) – Ідентифікатор користувача (User ID) – Назва портфеля (Portfolio Name) – Дата створення (Creation Date)
Позиція (Position)	<ul style="list-style-type: none"> – Ідентифікатор позиції (Position ID) – Ідентифікатор портфеля (Portfolio ID) – Ідентифікатор криптовалюти (Cryptocurrency ID) – Кількість одиниць (Quantity) – Середня вартість (Average Cost)
Повідомлення (Message)	<ul style="list-style-type: none"> – Ідентифікатор повідомлення (Message ID) – Ідентифікатор користувача відправника (Sender User ID) – Ідентифікатор користувача отримувача (Recipient User ID) – Текст повідомлення (Message Text) – Дата надсилання (Sending Date)

Продовження табл. 3.1.

1	2
Транзакція (Transaction)	<ul style="list-style-type: none"> – Ідентифікатор транзакції (Transaction ID) – Ідентифікатор користувача (User ID) – Ідентифікатор криптовалюти (Cryptocurrency ID) – Тип транзакції (Transaction Type) – Кількість одиниць (Quantity) – Ціна транзакції (Transaction Price) – Дата транзакції (Transaction Date)
Журнал входу (Login Log)	<ul style="list-style-type: none"> – Ідентифікатор запису журналу (Log ID) – Ідентифікатор користувача (User ID) – Дата та час входу (Login DateTime) – IP-адреса (IP Address)
Новина (News)	<ul style="list-style-type: none"> – Ідентифікатор новини (News ID) – Заголовок (Title) – Текст новини (News Text) – Дата публікації (Publication Date)

ДОДАТОК Ю. Рис. 3.5. Діаграма типу сутність-зв'язок концептуальної моделі бази даних



ДОДАТОК Я. Таблиця 3.2. Опис таблиць бази даних

Таблиця	Опис
1	2
"Користувач" (User)	Містить основну інформацію про користувачів, таку як їх унікальний ідентифікатор (user_id), ім'я користувача (username), електронну пошту (email), пароль (password) та дату реєстрації (registration_date).
"Позиція" (Position)	Містить інформацію про позиції в портфелях користувачів, таку як їх унікальний ідентифікатор позиції (position_id), ідентифікатор портфеля, до якого належить позиція (portfolio_id), ідентифікатор криптовалюти, яка знаходиться в позиції (cryptocurrency_id) та кількість криптовалюти в позиції (quantity).
"Криптовалюта" (Cryptocurrency)	Містить основну інформацію про криптовалюти, таку як їх унікальний ідентифікатор (cryptocurrency_id), назву (name), символ (symbol) та ринкову ціну (market_price).

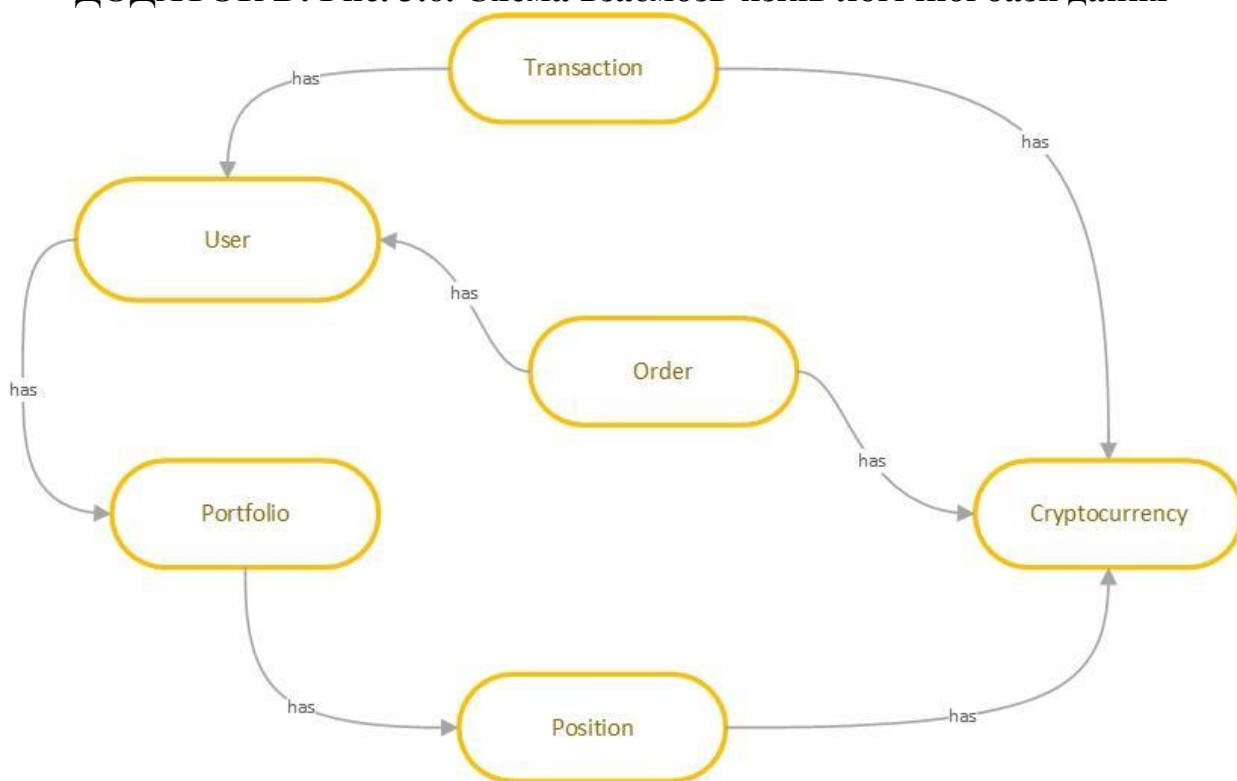
Продовження табл. 3.2

1	2
"Замовлення" (Order)	Містить інформацію про замовлення користувачів на купівлю або продаж криптовалют, таку як унікальний ідентифікатор замовлення (order_id), ідентифікатор користувача, який розмістив замовлення (user_id), ідентифікатор криптовалюти, на яку оформлено замовлення (cryptocurrency_id), тип замовлення, що може бути "купівля" або "продаж" (order_type), кількість криптовалют в замовленні (quantity), ціна за одиницю криптовалют в замовленні (price) та дата замовлення (order_date).
"Портфель" (Portfolio)	Містить інформацію про портфелі користувачів, таку як їх унікальний ідентифікатор портфеля (portfolio_id) та ідентифікатор користувача, до якого належить портфель (user_id).

Продовження табл. 3.2

1	2
"Транзакція" (Transaction)	Містить інформацію про транзакції з криптовалютами, які здійснюються користувачами, таку як унікальний ідентифікатор транзакції (transaction_id), ідентифікатор користувача, який здійснив транзакцію (user_id), ідентифікатор криптовалюти, якою здійснено транзакцію (cryptocurrency_id), тип транзакції, що може бути "купівля" або "продаж" (transaction_type), кількість криптовалюти в транзакції (quantity), ціна за одиницю криптовалюти в транзакції (price) та дата транзакції (transaction_date).

ДОДАТОК D. Рис. 3.6. Схема взаємозв'язків логічної бази даних



ДОДАТОК F. Частина коду контроллеру для Http-запитів

[HttpGet]

```
public IActionResult Range(DateTime startDate, DateTime endDate)
{
    if(endDate < startDate)
        return Json(new { success = false, error = "Start date is less than end
date" });
    var tasks = _service.GetInRange(startDate, endDate);
    return Ok(tasks);
}
```

[HttpGet]

```
public IActionResult Details(Guid id)
{
    try
    {
        var task = _service.Details(id);

        if (task == null)
            return Json(new { success = false });
        else
            return Ok(task);
    }
    catch (Exception e)
    {
        _logger.LogInformation($"Operation error: details [{e}]");
    }
    return Json(new { success = false });
}
```

ДОДАТОК G. Unit-тест для методу контроллеру

```
[Theory]
[InlineData("12")]
[InlineData("")]
[InlineData(null)]
[InlineData("123456789012345678901234567890 invalid")]
public void TaskController_Create_WithInvalidTitle_ReturnsError(string
taskTitle) {
    // Arrange
    var model = new TaskRequestModel() {
        Title = taskTitle,
        Description = "desc",
        Date = DateTime.Now,
        Color = "#ffffff"
    };
    var logger = new
Mock<Microsoft.Extensions.Logging.ILogger<TaskController>>();
    var context = InitHelpers.GetDbContext();
    var taskService = new Mock<IServiceItem<Task>>();
    var validator = new TaskRequestModelValidator();
    var service = new TaskController(logger.Object, context.Object,
taskService.Object, validator);
    //Act
    var result = service.Create(model);
    //Assert
    Assert.NotNull(result);
    var validationResult = validator.Validate(model);
    Assert.NotNull(validationResult);
    Assert.False(validationResult.IsValid);
}
```