

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-05

Дипломний проект

**здобувача освіти денної форми навчання
РП.05.14.000.ДП**

***МОКРИЦЬКОГО
ПАВЛА ВОЛОДИМИРОВИЧА***

**м. Одеса
2022 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **121 «Інженерія програмного забезпечення»**

Освітня програма: «**Розробка програмного забезпечення**»

Група: **4РП-05**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка чат-боту месенджера Telegram для виконання операцій з цифровою валютою (реалізація алгоритму обробки)

Проектний матеріал складається з пояснювальної записки на 55 сторінках та графічного (презентаційного) матеріалу на 25 аркушах (слайдах).

Дипломник _____ (Мокрицького П.В.)

Керівник _____ (Кіреєв І.А.)

Консультанти:

з економічної частини _____ (Копайгородська Т.Г.)

з охорони праці _____ (Чорновол Н.І.)

з дотримання вимог ЄСКД _____ (Петрашова В.І.)

старший консультант _____ (Скорнякова О.В.)

До захисту допущений

Голова циклової комісії _____ (Скорнякова О.В.)

Завідувач відділення _____ (Суліма Ю.Ю.)

Захист « » _____ 2022 р. Протокол ДКК № _____

Оцінка ДКК _____

Секретар ДКК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та Ш
Спеціальність 121 "Інженерія програмного забезпечення"

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР _____

Беркань І.В.

" _____ " _____ 2022 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Мокрицького Павло Володимировича
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка чат-боту месенджера Telegram для виконання операцій з цифровою валютою (реалізація алгоритму обробки)

затверджена наказом по коледжу від " _____ " _____ 2022 р. № _____

2. Термін здачі закінченого проекту (роботи) _____

3. Вихідні данні до проекту (роботи) _____

1. Використання номеру гаманця криптовалюти Ефір;

2. Використання технології блокчейн;

3. Мова програмування Python, інтегроване середовище розробки PyChart;

4. Реалізувати програмний модуль розробленої системи: відстеження транзакцій, балансу гаманця мережі криптовалюти Ethereum, курсу криптовалют біржі, зміна мови інтерфейсу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
Опис предметної області. Загальні відомості про криптовалюту. Переваги криптовалют.
Огляд мережі криптовалюти Ethereum. Види взаємодії із мережею криптовалюти Ethereum.
Розробка алгоритму обробки даних. Розробка інтерфейсу програми. Проектування структур даних програми. Проектування структури модулів програми. Тестування розробленої системи. Економічна частина. Охорона праці

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Вартість криптовалюти у часі; Схема взаємодії користувача та розробленого проекту;
Порівняння аналогів; Алгоритм прийому нових блоків; Алгоритм обробки блока; Алгоритм реєстрації адреси; Загальна структура проекту; Структура користувацького інтерфейсу;
Склад користувацький інтерфейсу; Інструкція для користувача інтерфейсу; Меню боту;
Зміна мови інтерфейсу; Найцінніші гаманці Ефіру; Баланс гаманця; Транзакція у боті;
Перевірка роботи курсу криптовалют.

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1. Технологічний розділ	Кірсев І.А.		
2. Екон. частина	Копайгородська Т.Г.		
3. Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		

7. Дата видачі завдання _____

Керівник *Кірсев І.А.* _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1.	Вступ. Постановка мети та задач проектування	17.05.2022	
2.	Аналіз предметної галузі	18.05.2022	
3.	Огляд функціональних елементів для реалізації моделі	19.05.2022	
4.	Основні вимоги до сервісу для взаємодії з біржею	20.05.2022	
5.	Вибір технічних та програмних засобів розробки	23.05.2022	
6.	Вибір технологій взаємодії з мережею криптовалюти	24.05.2022	
7.	Опис способів реалізації інтерфейсу користувача	26.05.2022	
8.	Опис алгоритму прийому блоків даних	30.05.2022	
9.	Опис алгоритму обробки даних	31.05.2022	
10.	Опис алгоритму роботи інтерфейсу	1.06.2022	
11.	Розробка інтерфейсу програми	3.06.2022	
12.	Тестування розробленої системи	6.06.2022	
13.	Аналіз результатів, підготовка презентації	8.06.2022	
14.	Розробка питань з економіки та охорони праці	10.06.2022	
15.	Підготовка проекту до захисту	14.06.2022	

Дипломник _____
(підпис)

Керівник _____
(підпис)

ЗМІСТ

Вступ.....	7
1 Технологічний розділ.....	7
1.1 Формування функціоналу та аналіз технологій.....	7
1.2 Вибір технологій та інструментів для реалізації моделі.....	10
1.2.1 Технологія Ethereum.....	11
1.2.2 Криптовалюта Ефір.....	13
1.3 Визначення задач проектування.....	14
1.3.1 Мінімізація порогу входу.....	15
1.3.2 Створення простий, інтуїтивно зрозумілий для звичайної людини інтерфейс.....	16
1.3.3 Широкий спектр можливостей у плані функціоналу програми.....	14
1.3.4 Короткий огляд альтернативних мереж.....	15
1.4 Основні етапи проектування.....	16
1.4.1 Вибір технологій взаємодії з мережею криптовалюти.....	15
1.4.2 Огляд способів реалізації інтерфейсу користувача.....	15
1.4.3 Вибір мови програмування та середовища розробки.....	15
1.5 Розробка алгоритму обробки даних.....	24
1.5.1 Прийом блоків даних.....	14
1.6 Розробка інтерфейсу програми.....	43
1.6.1 Проектування структур даних програми.....	14
1.6.2 Проектування структури модулів програми.....	15
1.7 Тестування розробленої системи.....	50
1.7.1 Інструкція для користувача інтерфейсу.....	50
1.7.2 Зміна мови інтерфейсу.....	53
1.7.3 Перегляд балансу гаманця.....	55
1.7.4 Перегляд останньої транзакції.....	67
2 Економічна частина.....	57
2.1 Резюме.....	45
2.2 Визначення трудоміткості розробки програмного забезпечення.....	34

					РП 05.14 000.00 ДП ПЗ	Арх.
Зл	Арх.	№ докум.	Проект	Дата		5

2.3 Розрахунок ціни програмного продукту	34
3. Охорона праці.....	47
3.1 Вступ.....	47
3.2 Аналіз умов праці програміста	47
3.3 Розробка заходів з охорони праці.....	48
3.3.1 Вимоги до виробничих приміщень.....	48
3.3.2 Освітлення.....	49
3.3.3 Параметри мікроклімату.....	50
3.3.4 Шум і вібрація.....	51
3.3.5 Електромагнітні і іонізуючі випромінювання.....	51
3.3.6 Ергономічна вимоги до робочого місця.....	52
3.4 Пожежна безпека.....	53
Висновки	53
Перелік використаних джерел.....	54
ДОДАТОК А. Фрагмент листингу програмного модуля обробки транзакцій, балансу гарантія та хендлерів.....	56

						РП 05.14 000.00 ДП ПЗ	Арх.
Зл	Арх	№ розр	Проває	Дата			б

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Формування функціоналу та аналіз технологій

Розроблюваний проект призначений для взаємодії з криптовалютними біржами за допомогою Телеграм-боту, в залежності від потреб користувача, починаючи з курсів валют, закінчуючи балансом, списком транзакцій, тощо. Використовувати програму зможе будь-яка людина з встановленим Телеграмом та активованим у ньому акаунтом.

Telegram — це служба обміну повідомленнями, яка дуже схожа на WhatsApp, але з додатковими перевагами шифрування даних, безпеки та конфіденційності. Telegram використовує, що якщо новий контакт підписується на бізнес-канал, компанія не отримує доступу до жодної з контактних або особистих даних користувачів.

Користувачі Telegram можуть шукати назви компаній і підписатися на компаній через область відкриття. Ця домовленість працює таким чином, що клієнти можуть зв'язатися з компаніями, якщо вони хочуть, але компанії не можуть зв'язатися з клієнтами без їхнього дозволу.

Телеграмний бот — це особливий тип користувачів, який є не людиною, а комп'ютерною програмою, яка може обслуговувати компанії чи бренди з багатьма функціями, такими як надсилання інформації, нагадування, відтворення мелодій, заповнення тощо. Бот може опублікувати повідомлення в групі або каналі.

Звичайні користувачі можуть стежити за будь-яким телеграм-ботом, за яким вони хотіли б бути в курсі. З усіх чотирьох однорангових типів бот Telegram має багато функцій, які дійсно корисні для бізнесу. Telegram надає API для створення ботів для соціальних взаємодій, продуктивності, ігор та послуг електронної комерції на платформі.

Боти Telegram також можуть надавати підтримку клієнтам або збирати потенційних клієнтів, підключавши їх до CRM, системи продажу квитків або платформи обміну повідомленнями. Давайте розглянемо причини, чому боти Telegram надзвичайно корисні для бізнесу:

										Арт.
										8
Зл	Арт.	№ докум.	Провак	Дата	РП 05.14 000.00 ДП ПЗ					

процес і взаємодії з інтерфейсом програми також має бути застосовано модуль для зберігання користувацьких даних. Розроблювана система буде складатися з трьох функціональних елементів:

1. Інтерфейс користувача;
2. Інтерфейс локальної бази даних;
3. Інтерфейс роботи з мережею криптовалют.

Взаємодія елементів та інтерфейсів розроблюваної системи представлена на рисунку 1.1.

Взаємодія користувача та сервісу, починається з привітання та пропозиції ввести адресу мережі криптовалют для відстежування з боку програми.

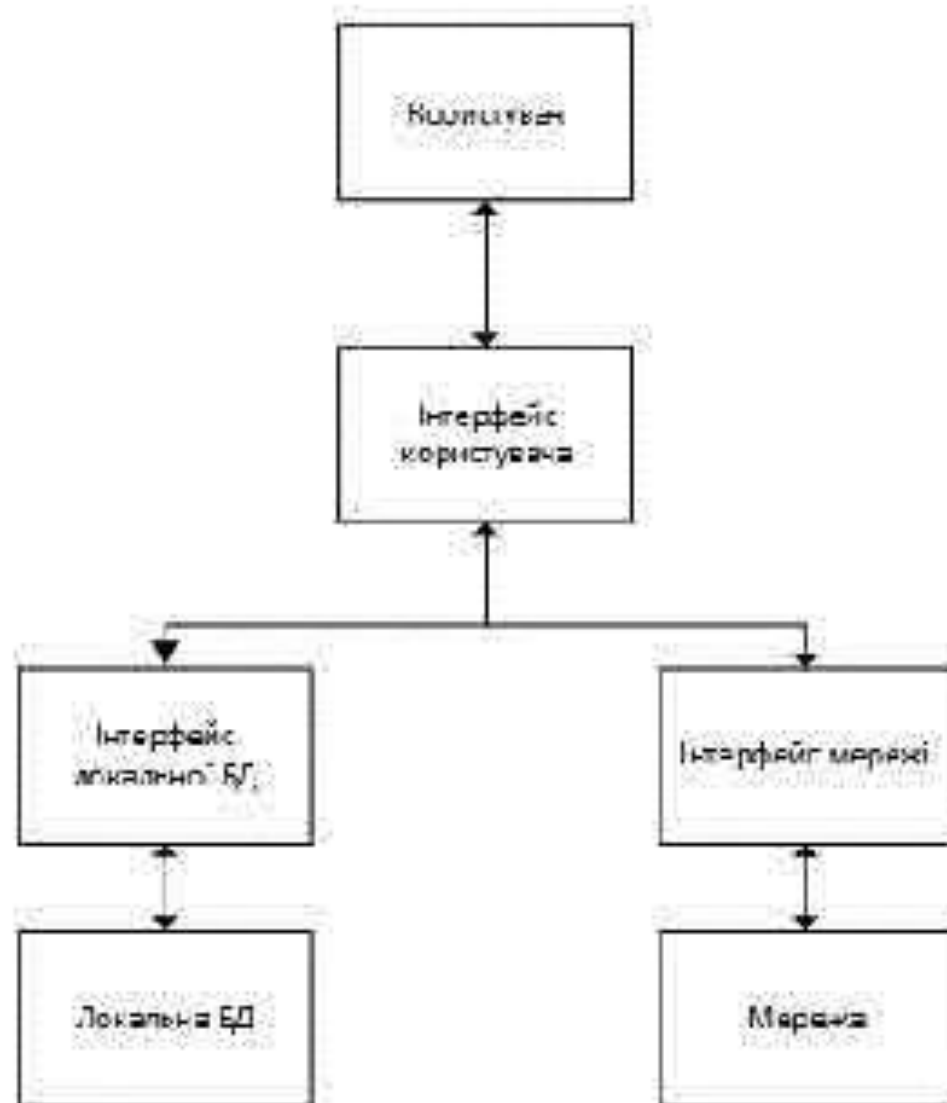


Рисунок 1.1. Взаємодія елементів та інтерфейсів розроблюваної системи

Зл	Арк	№ докум	Проект	Дата

РП 05.14 000.00 ДП ПЗ

Арк

12



Рисунок 13 Алгоритм прийому блоків даних

Зм	Лр	№ розд	Підпис	Дата

РП 05.14 000.00 ДП ПЗ

Лр

25

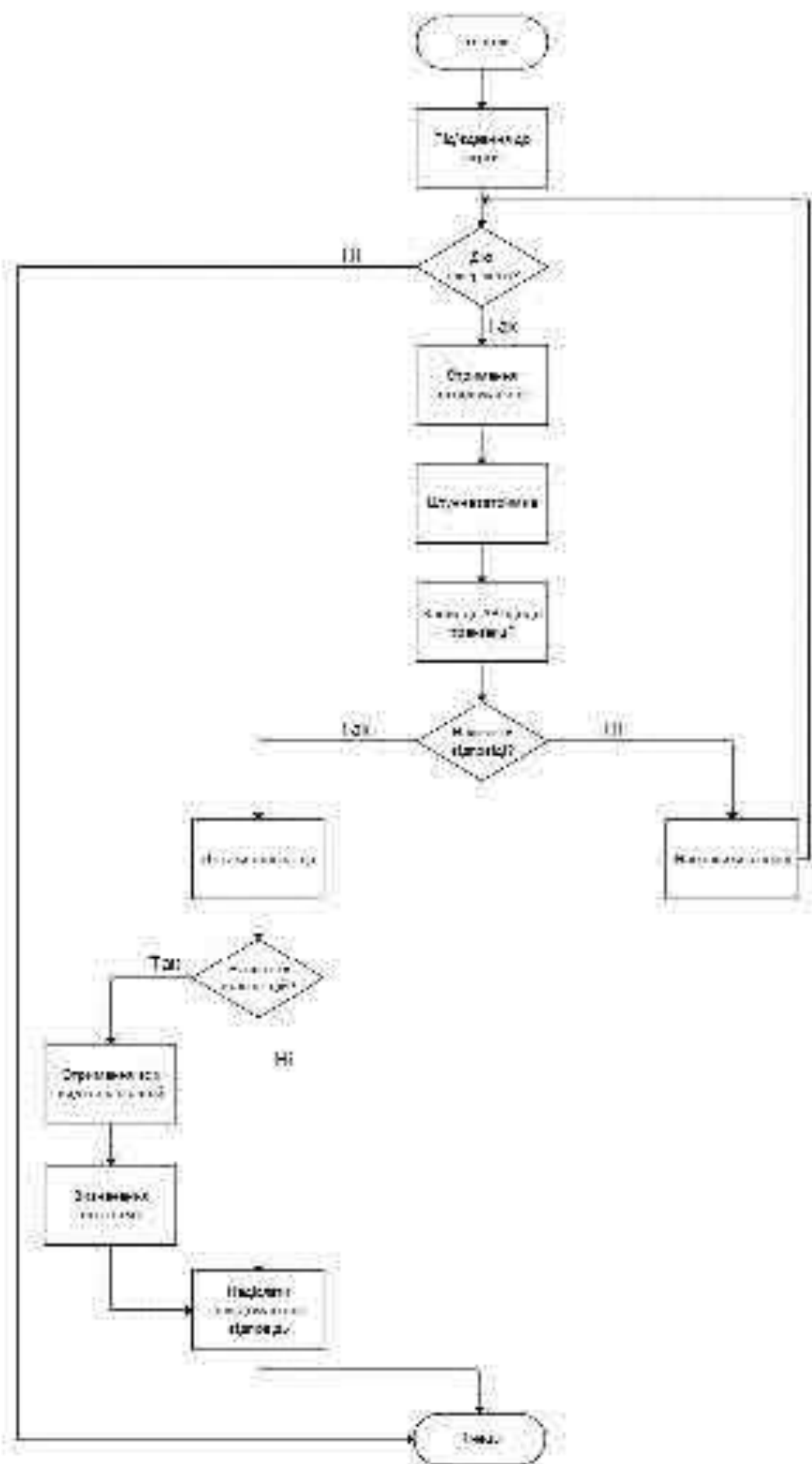


Рисунок 1.4 Алгоритм обробки

Зл	Арк	№ докум	Примк	Дата

РП 05.14 000.00 ДП ПЗ

Арк

27

1.6 Розробка інтерфейсу програми

Інтерфейс програми збудований на технології чат-ботів у месенджері Telegram. Цей підхід характеризується як подібність до діалогу між реальною людиною/користувачем та програмним додатком за чітко визначеним алгоритмом, або, простіше кажучи, сценарієм. Фактично, цей сценарій є повноцінним автоматом, оскільки він має власні умови та правила, за якими відбуваються переходи у інтерфейсі. Також використовується внутрішня пам'ять для збереження проміжних даних.

Інтерфейс програми має бути інтуїтивно зрозумілий користувачу для вдалої взаємодії з користувачем, у випадку цього проекту для взаємодії зрозумілий сценарій реєстрації адреси цільової мережі криптовалют. Під час діалогу додаток має отримати власне адресу для відстежування для відображення інформації, що необхідно в процесі експлуатації.

Даний алгоритм зображено на рисунку 1.5.

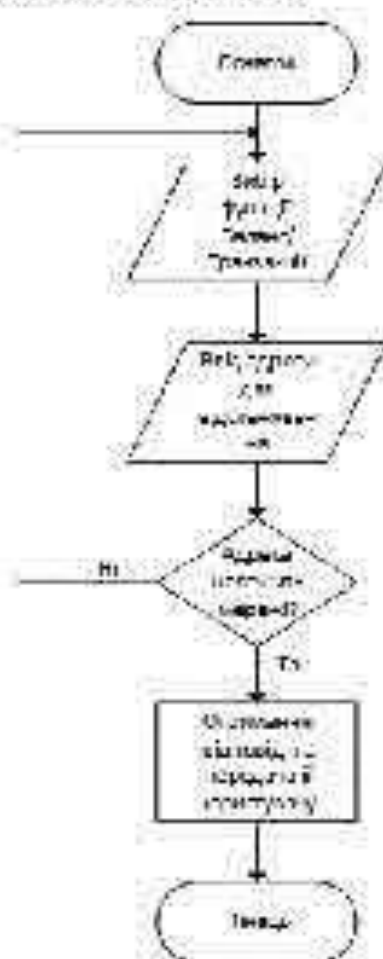


Рисунок 1.5 Алгоритм роботи інтерфейсу

Зл	Арх	№ докум	Проект	Дата

РП 05.14 000.00 ДП ПЗ

Унікальний ідентифікатор – число, що присвоюється всім користувачам месенджеру при реєстрації. Він дозволяє відрізнити певного користувача серед десятків мільйонів інших. Саме за допомогою нього адресується, наприклад, повідомлення.

1.6.2 Структурна структура модулів програми

Загальна структура програми зображена на рисунку 1.6.



Рисунок 1.6 Основні файли програми

Розробка має наступні виконуваних файли:

app.py – файл для запуску боту

loader.py – файл з основними параметрами для роботи боту, такі як токен

requirements.txt - вимоги до програмного забезпечення для нормальної роботи програми

На рисунку 1.7 відображена структура користувацького інтерфейсу.

Користувацький інтерфейс складається з трьох основних частин:

1. handlers;
2. markup;
3. user.

Директорія під назвою handlers та її дочірні директорії містять файли, що обробляють користувацькі запити. Вони формують відповіді на повідомлення чи натискання кнопок. Це головна частина інтерфейсу, до її файлів можуть імпортуватися інші перелічені вище частини.

Директорія markup містить допоміжні функції для формування користувацьких меню - а саме, кнопок, за допомогою яких виконується взаємодія користувача з програмою.

Файли у директорії state містять дані з переліком станів, яких може набувати користувач при роботі з програмою. Це необхідно для коректної роботи алгоритмів програми.

										Арс
Зл	Арс	№ раху	Прова	Дата	РП 05.14 000.00 ДП ПЗ					зі



Рисунок 1.7 Файлы проекта



Рисунок 1.8 Файлы статус

					РП 05.14 000.00 ДП ПЗ	Апр.
Эл.	Апр.	№ докум.	Проект	Дата		31

1.7 Тестування розробленої системи

1.7.1 Інструкція для користувача інтерфейсу

Як і у більшості випадків, необхідно надіслати команду “/start”. Це стандартна команда для всіх віртуальних «спікерозмовників» у месенджері Telegram. Отримавши вище зазначену команду, бот відповідає вітальним повідомленням.

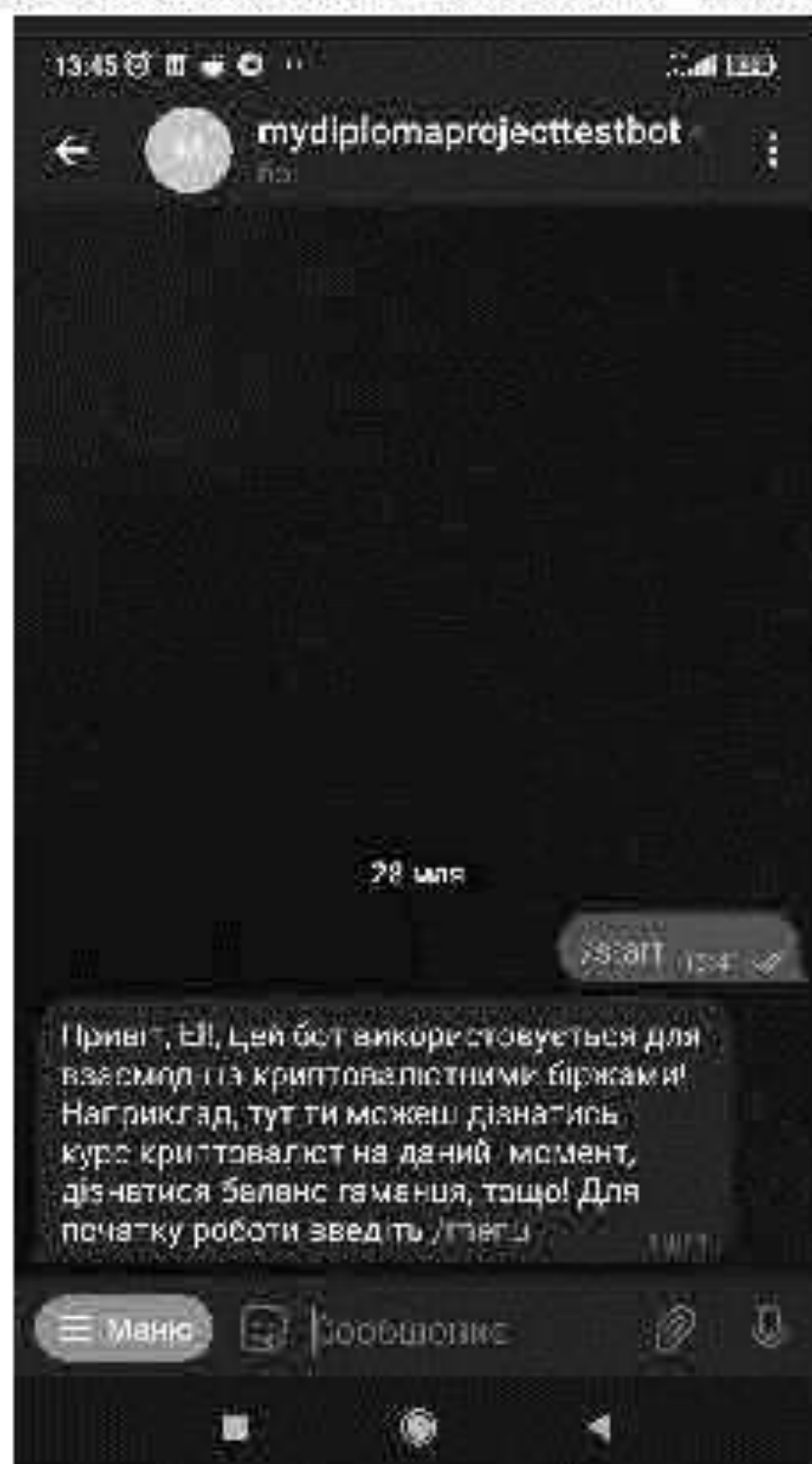


Рисунок 1.9 Привітання

						РП 05.14 000.00 ДП ПЗ	Ар.
Зл	Ар.	№ раху	Прова	Дата			32

Для початку роботи натискаємо на підсвічену команду *Меню*, та отримуємо у своє розпорядженні меню боту



Рисунок 1.10 Меню

Зл	Агр	№ раху	Підпис	Дата

РП 05.14 000.00 ДП ПЗ

Агр

33

Перевіримо у бота



Рисунок 1.14 Баланс гаманця

І як бачимо, всі данні збігаються, тільки у боті число є більш округленим.

						РП 05.14 000.00 ДП ПЗ	Арх
Зна	Арх	№ докум	Провак	Дата			36

1.7.4 Перегляд останньої транзакції

Щоб переглянути транзакції будь-якого гаманця, що вас цікавить, потрібно натиснути на кнопку «Перевірити останню Ethereum транзакцію», після чого бот попросить вас вказати унікальний номер гаманця, після чого він знову відправить ці дані до API, а отриманим відповідь – передасть її вам.

Перевірити роботу будемо у цього же гаманця за допомогою Etherscan.



Рисунок 1.15 Транзакція Etherscan

					РП 05.14 000.00 ДП ПЗ	Арх.
Зл	Арх.	№ рахунок	Пішовис	Дата		37

Тепер переір'яємо роботу боту:



Рисунок 1.16 Транзакція у боті

Як бачимо, усі дані цілком збігаються. Відрізняється лише час тому, що я використовуву часовий пояс UTC +3.

						РП 05.14 000.00 ДП ПЗ	Ар
Зл	Ар	№ раху	Піровас	Дата			22

1.7.5 Перегляд ціни популярних криптовалют

Щоб переглянути ціну на найпопулярніші криптовалюти потрібно натиснути на кнопку «Дізнатися курс криптовалют», після чого ви отримаєте ціле меню з вибором, яка саме криптовалюта вас цікавить. Після вибору ви отримаєте ціну в доларах на даний момент.



Рисунок 1.17 Перевірка роботи курсу криптовалют

						РП 05.14 000.00 ДП ПЗ	Ар.
Зл	Ар.	№ рахунок	Прізвище	Дата			39

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл. 2.2

Таблиця 2.2

Обсяг ПП, тис. умов машинних команд	Норма часу, люд. год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається з урахування норми часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7+0,8$): $T^* = 299 \times 0,8 = 183,2$ (люд. днів).

Трудомісткість програмного продукту визначається для кожного етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності

розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{m1} = T^* \times L_1 \times K_n \quad (2.1)$$

$$T_{m2} = T^* \times L_2 \times K_n \quad (2.2)$$

$$T_{m3} = T^* \times L_1 \times K_n \times K_p \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 2.2.);

K_n – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.3.); K_p – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.4.).

Таблиця 2.2. Значення лінійних коефіцієнтів трудомісткості стадій в загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L ₁)	0,15	0,12	0,12
ТП (L ₂)	0,16	0,15	0,11
РП (L ₃)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.3. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K ₀
А	Принципово нові ПО	1,75 – 1,2
Б	ПО – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПО на вчній аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розробки вального ПО типовими програмами, %	Значення K ₁
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{\text{тз}} = T^{\text{тз}} * L_1 * K_0 = 183,2 * 0,12 * 0,7 = 15,38 \text{ (люд.доден)} \quad (2.4)$$

Трудомісткість розробки технічного проекту

$$T_{\text{тп}} = T^{\text{тп}} * L_2 * K_0 = 183,2 * 0,11 * 0,7 = 14,10 \text{ (люд.доден)} \quad (2.5)$$

встановлено мінімальну заробітну плату умисльному розмірі з 1 січня 2022 року - 6500 гривень; мінімальну погодинну тарифну ставку - 39.26 грн.

Таблиця 2.6 Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудоємність робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1 Розробка ПП	135,2	39.26	5 307,9
2 Контроль керівника	4,2	50	2 100
3. Нормоконтроль	9,45	50	472,5
Усього	-	-	$\Sigma_{30} = 7 880,4$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведено в таблицю 2.7

Таблиця 2.7 Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	70	1.5	105
Разом	-	-	-	$B_{м1} = 105$
Транспортно-заготівельні витрати (10%)				$B_{тп-з} = 0,1 \times B_{м1} = 0,1 * 105 = 10,5$
Усього				$B_{м} = B_{м1} + B_{тп-з} = 115,5$

Зл	Агр	№ рахунок	Провак	Дата

РП 05.14 000.00 ДП ПЗ

Агр

44

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формов, приведеню в таблиці 2.8.

Таблиця 2.8 Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	115,5	V_m (див. табл. 2.7)
2. Основна заробітна плата	7 880,4	Z_o (див. табл. 2.6)
3. Додаткова заробітна плата	1 182	$Z_d = 0,15 \times Z_o = 7 880,4 * 0,15$
4. Відрахування до єдиного фонду соціального внеску	1 993,7	$V_{c.s.v.} = 0,22 \times (Z_o + Z_d) = 0,22 * (7 880,4 + 1 182)$
5. Накладні витрати	2 364,1	$V_{нак.} = 0,3 \times Z_o = 0,3 * 7 880,4$
6. Повна собівартість	13 535,7	$C_{пов.} = V_m + Z_o + Z_d + V_{c.s.v.} + V_{нак.}$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$\Pi = (C_o * P) / 100 = (13 535,7 * 12) / 100 = 2 030,3 \text{ грн} \quad (2.7)$$

Де p – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$C_o = C_{пов.} + \Pi = 15 566 \text{ грн} \quad (2.8)$$

Податок на додану вартість визначаємо по наступній формулі:

$$\text{ПДВ} = 0,2 * C_o = 3 113,2 \text{ грн} \quad (2.9)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$C_p = C_o + \text{ПДВ} = 18 679,2 \text{ грн} \quad (2.10)$$

загального освітлення допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану та збільшення освітленості екрану більше ніж до 300 лк. Світильники місцевого освітлення повинні мати напівпрозорий відбивач світла з захисним кутом не меншим за 40 град. Необхідно передбачити обмеження прямого блиску від джерела природного та штучного освітлення, при цьому яскравість поверхонь, що світяться (живна, джерела штучного світла) і перебувають у полі зору, повинна бути не більшою за 200 кд/кв.м. Необхідно обмежувати відбитий блиск шляхом правильного вибору типів світильників та розміщенням робочих місць відносно джерел природного та штучного освітлення.

3.3.3 Параметри мікроклімату

Принцип нормування мікроклімату - створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем.

Нормовані параметри мікроклімату для приміщень з ВДТ та ПЕОМ:

Пора року	Категорія робіт	Температура повітря, С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
		оптимальна	оптимальна	оптимальна
Холодна	Легка 1-а	22-24	40-60	0.1
	Легка 1-б	21-23	40-60	0.1
Тепла	Легка 1-а	23-25	40-60	0.1
	Легка 1-б	22-24	40-60	0.2

Рівні іонізації повітря приміщень при роботі на ВДТ та ПЕОМ:

Рівні	Кількість іонів в 1 см. Куб повітря	
	Мінімально необхідні	400
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

Для забезпечення комфортних умов використовуються як організаційні методи (раціональна організація проведення робіт в залежності від пори року та доби, чергування праці та відпочинку), так і технічні засоби (вентиляція, кондиціонування повітря, опалювальна система).

3.3.4 Шум і вібрація

Робітники в умовах тривалого шумового впливу відчувають дратівливість, головні болі, запаморочення, зниження пам'яті, підвищену стомлюваність, зниження апетиту, болі вухах іт. д. Такі порушення в роботі ряду органів і систем тим самим організму людини можуть викликати негативні зміни в емоційному стані людини впродовж до стресових. Під впливом шуму знижується концентрація уваги, порушуються фізіологічні функції, з'являється втома у зв'язку з підвищеними енергетичними витратами і нервово-психічною напругою, погіршується мовна комутація. Усе це знижує працездатність людини та її продуктивність, якість і безпеку праці.

Для забезпечення нормованих рівнів шуму у виробничих приміщеннях та на робочих місцях застосовуються шумопоглинальні засоби, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками.

3.3.5 Електромагнітні та іонізуючі випромінювання

Потужність експозиційної дози рентгенівського випромінювання на відстані 0,05 м від екрана та корпусу відеотермінала при будь-яких положеннях регульованих пристроїв відповідно до Норм радіаційної безпеки України (НРБ У-97) затверджених постановою державного санітарного лікаря Міністерства охорони здоров'я України від 18.08.97 N 58, не повинна перевищувати $7,74 \times 10^{-5}$ ступ. - 12 А/кг, що відпо відає еквівалентній дозі 0,1 мбер/год (100 мкР/год). Вміст озону в повітрі робочої зони не повинен перевищувати 0,1 мг/куб м; вміст оксиду азоту - 5 мг/куб м; вміст пилу - 4 мг/куб м.

						РП 05.14 000.00 ДП ПЗ	А пр.
Зл	А пр.	№ докум.	Проект	Дата			30

3.4 Пожежна безпека

Пожежна безпека входить в комплекс заходів з охорони праці, і організаційна робота в цій сфері на об'єктах господарювання включає широкий спектр заходів, а саме:

- 1) створення умов для безпечної праці,
- 2) мінімізації ризику виникнення пожеж,
- 3) своєчасне і повноцінне забезпечення технічними засобами для запобігання займанню та усунення самих пожеж та їх наслідків,
- 4) контроль дотримання протипожежних вимог і норм законодавства,
- 5) розробка і впровадження регламентів по гасінню пожеж, евакуації та порядку з місць пожежі й задимлення людей і майна (матеріальних цінностей),
- 6) внутрішнє і зовнішнє навчання співробітників.

Пожежна безпека при роботі з комп'ютером передбачає обережність при обслуговуванні, ремонтних та профілактичних роботах та виконувannya всіх інструкцій працівником щодо пожежої безпеки.

Приміщення, в яких розміщено ПК, слід оснащувати переносними газозвими вогнегасниками з розрахунку один вогнегасник ВВК-1А чи ВВК-2, але не менше ніж один вогнегасник зазначеного типу на приміщення.

					РП 05.14 000.00 ДП ПЗ	А пр.
Зл	А пр.	№ докум	Підпис	Дата		32

ВИСНОВКИ

Основною дипломного проекту була розробка чат-бота для взаємодії з криптовалютною біржею та криптовалютою Ethereum. У цій роботі було розглянуто і проаналізовано переваги та недоліки, а саме, ботів в месенджері Telegram, їх спосіб створення та взаємодія з ними, з криптовалютою та біржею. А саме реалізація алгоритму обробки даних користувача при взаємодії з гаманцем цифрової валюти.

Однією з головних особливостей даної програми є швидка та проста для рядового користувача взаємодія як з криптовалютною біржею CoinGecko так і з криптовалютою Ethereum, для цього були використані спеціалізовані API, такі як Etherscan API та CoinGecko API.

За допомогою вищезазначених засобів було реалізовано бота, що заощадить ваш час при роботі з вищезазначеними ресурсами, основними функціями якого є перегляд стану криптовалютного рахунку, транзакцій або перевірка ціни на найпопулярнішу на даний момент криптовалюту.

У дипломному проекті було проведено тестування розробленої системи обробки транзакцій, балансу гаманця мережі криптовалюти Ethereum, курсу криптовалют біржі CoinGecko. Результати оцінювання чітко проаналізовано. Даний розділ надає інструкцію експлуатації для користувача програмного продукту, описує можливості та основні способи взаємодії з наданим інтерфейсом.

					РП 05.14 000.00 ДП ПЗ	Апр
Зл	Апр	№ докум	Проває	Дата		53

12. Introduction to the Google Assistant Library [Электронный ресурс] – Режим доступа: <https://developers.google.com/assistant/sdk/guides/library/python>
13. Messaging, simplified [Электронный ресурс] - Режим доступа: https://www.nexmo.com/products/sms/build?utm_source=google_search&utm_medium=paid&utm_campaign=EMEA_EURO_EN_SMS_General_B&utm_term=
14. SpeechRecognition
15. Enhance Your Presence on Search [Электронный ресурс] - Режим доступа: <https://developers.google.com/search/>
16. Telegram APIs [Электронный ресурс]-Режим доступа: <https://core.telegram.org/>
17. pyTelegramBotAPI [Электронный ресурс]-Режим доступа: <https://github.com/eternnoir/pyTelegramBotAPI>
18. Пишем telegram-бота на python с помощью библиотеки telebot [Электронный ресурс] - Режим доступа: <https://habr.com/ru/post/448310>

						РП 05.14 000.00 ДП ПЗ	Ар.
Эл	Ар.	№ доку	Проект	Дата			35

ДОДАТОК А. Фрагмент лістингу програмного модулю обробки транзакцій, балансу гаманця та хендлерів у Телеграм-боті на мові Python.

```
transactions.py
import logging

from aiogram import types
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters import Command, Text
from aiogram.types import CallbackQuery
from bitcoinlib.wallets import Wallet
from pycoingecko import CoinGeckoAPI
from requests import get
from matplotlib import pyplot as plt
import asyncio
import data
from states import BalanceCheck, TransCheck

cg = CoinGeckoAPI()
API_KEY = '91NEEET9HB38K0W/HAP46M8314WU39ZNYX15"
# https://api.etherscan.io/api
BASE_URL = "https://api.etherscan.io/api"
ETHER_VALUE = 10 ** 18

from keyboards.default import menu
from keyboards.default.menu import menu_eng, menu_ru
from keyboards.inline.callback_datas import choice_callback
from keyboards.inline.choice_buttons import choice
from loader import dp, bot
from datetime import datetime

to=""
from_addr=""
value=0
gas=0
time=""

@dp.message_handler(Text(equals=["Перевірити останню Ethereum транзакцію"]))
async def get_balance(message: types.Message):
    await message.answer(f'Введіть номер гаманця: ')
    await TransCheck.askForTrans1.set()

@dp.message_handler(Text(equals=["Check last Ethereum transaction"]))
async def get_balance(message: types.Message):
    await message.answer(f'Введіть номер гаманця: ')
    await TransCheck.askForTrans1.set()

@dp.message_handler(Text(equals=["Последняя Ethereum транзакция"]))
async def get_balance(message: types.Message):
    await message.answer(f'Введіть номер гаманця: ')

```

```
await TransCheck.askForTrans1.set()
```

```
@dp.message_handler(state=TransCheck.askForTrans1)
```

```
async def answer_q2(message: types.Message, state: FSMContext):
```

```
    answer = message.text
```

```
    await state.update_data(address=answer)
```

```
    def make_api_url(module, action, address, **kwargs):
```

```
        url = BASE_URL +
```

```
        f"?module={module}&action={action}&address={address}&apikey={API_KEY}"
```

```
        for key, value in kwargs.items():
```

```
            url += f"&{key}={value}"
```

```
    return url
```

```
    def get_transactions(address):
```

```
        transactions_url = make_api_url('account', 'txlist', address, startblock=0,  
endblock=99999999, page=1,
```

```
            offset=10000,
```

```
            sort="asc")
```

```
        response = get(transactions_url)
```

```
        data = response.json()['result']
```

```
        for tx in data:
```

```
            global to
```

```
            to = tx['to']
```

```
            from_addr = tx['from']
```

```
            value = int(tx['value']) / ETHER_VALUE
```

```
            gas = int(tx['gasUsed']) * int(tx['gasPrice']) / ETHER_VALUE
```

```
            time = datetime.fromtimestamp(int(tx['timeStamp']))
```

```
            asyncio.create_task(bot.send_message(message.from_user.id, f"-----\n To: {to}\n From: {from_addr}\n Value: {value}\n Gas Cost: {gas}\n Time: {time}", ))
```

```
        address = answer
```

```
        get_transactions(address)
```

```
    await state.finish()
```

```

menu.py
import logging

from aiogram import types
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters import Command, Text
from aiogram.types import CallbackQuery
from bitcoinlib.wallets import Wallet
from pycoingecko import CoinGeckoAPI
from requests import get
from matplotlib import pyplot as plt

import data
from states import BalanceCheck, TransCheck

cg = CoinGeckoAPI()
API_KEY = "91NEEET9HB38K0W/HAP46M8314W/U39ZNYX15"
# https://api.etherscan.io/api
BASE_URL = "https://api.etherscan.io/api"
ETHER_VALUE = 10 ** 18

from keyboards.default import menu
from keyboards.default.menu import menu_eng, menu_ru
from keyboards.inline.callback_datas import choice_callback
from keyboards.inline.choice_buttons import choice
from loader import dp, bot
from datetime import datetime

@dp.message_handler(Command("menu"))
async def show_menu(message: types.Message):
    await message.answer("Оберіть, що вас цікавить", reply_markup=menu)

@dp.message_handler(Text(equals=["Дізнатися баланс гаманця"]))
async def get_balance(message: types.Message):
    await message.answer(f"Введіть номер гаманця: ")

```

```
await BalanceCheck.askForBalance1.set()
```

```
@dp.message_handler(Text(equals=["Ethereum Wallet Balance"]))
```

```
async def get_balance(message: types.Message):
```

```
    await message.answer(f'Insert wallet number: ")
```

```
    await BalanceCheck.askForBalance1.set()
```

```
@dp.message_handler(Text(equals=["Узнать баланс Ethereum кошелька"]))
```

```
async def get_balance(message: types.Message):
```

```
    await message.answer(f'Введите номер кошелька: ")
```

```
    await BalanceCheck.askForBalance1.set()
```

```
@dp.message_handler(state=BalanceCheck.askForBalance1)
```

```
async def answer_q1(message: types.Message, state: FSMContext):
```

```
    answer = message.text
```

```
    await state.update_data(address=answer)
```

```
def make_api_url(module, action, address, **kwargs):
```

```
    url = BASE_URL +
```

```
f'?module={module}&action={action}&address={address}&apikey={API_KEY}'
```

```
    for key, value in kwargs.items():
```

```
        url += f'&{key}={value}'
```

```
    return url
```

```
def get_account_balance(address):
```

```
    get_balance_url = make_api_url('account', 'balance', address, tag='latest', x='2')
```

```
    response = get(get_balance_url)
```

```
    data = response.json()
```

```
    value = int(data['result']) / ETHER_VALUE
```

```
    return value
```

```
eth = get_account_balance(answer)
```

```
await message.answer(f'Balance: {eth} ETH')
```

```
await state.finish()
```

```
@dp.message_handler(Text(equals=["ЗМІНИТИ МОВУ🇺🇸🇸🇸"]))  
async def show_menu(message: types.Message):  
    await message.answer("Choose what do you want to do", reply_markup=menu_eng)
```

```
@dp.message_handler(Text(equals=["Change Language🇺🇸🇸🇸"]))  
async def show_menu(message: types.Message):  
    await message.answer("Выберите, что вас интересует", reply_markup=menu_ru)
```

```
@dp.message_handler(Text(equals=["Оменить язык🇺🇸🇸🇸"]))  
async def show_menu(message: types.Message):  
    await message.answer("Оберіть, що вас цікавить", reply_markup=menu)
```

```
@dp.message_handler(Text(equals=["Дзнати ся курс криптовалюти"]))  
async def get_course(message: types.Message):  
    await message.answer(text="Курси на даний момент", reply_markup=choice)
```

```
@dp.message_handler(Text(equals=["Узнати курс криптовалюти"]))  
async def get_course(message: types.Message):  
    await message.answer(text="Курси на даний момент.", reply_markup=choice)
```

```
@dp.message_handler(Text(equals=["Cryptocurrency rates"]))  
async def get_course(message: types.Message):  
    await message.answer(text="Current rates:", reply_markup=choice)
```

```
@dp.callback_query_handler(text='ethereum')
async def ethereum(call: CallbackQuery):
    price = cg.get_price(ids='ethereum', vs_currencies='usd')
    await call.answer(cache_time=60)
    await call.message.answer(f'Ethereum: 1 ETH = ${price[ethereum][usd]:.2f}')
```

```
@dp.callback_query_handler(text='bitcoin')
async def bitcoin(call: CallbackQuery):
    price = cg.get_price(ids='bitcoin', vs_currencies='usd')
    await call.answer(cache_time=60)
    await call.message.answer(f'Bitcoin: 1 BTC = ${price[bitcoin][usd]:.2f}')
```

```
@dp.callback_query_handler(text='litecoin')
async def bitcoin(call: CallbackQuery):
    price = cg.get_price(ids='litecoin', vs_currencies='usd')
    await call.answer(cache_time=60)
    await call.message.answer(f'Litecoin: 1 LTC = ${price[litecoin][usd]:.2f}')
```

```
@dp.callback_query_handler(text='tether')
async def bitcoin(call: CallbackQuery):
    price = cg.get_price(ids='tether', vs_currencies='usd')
    await call.answer(cache_time=60)
    await call.message.answer(f'Tether: 1 USDT = ${price[tether][usd]:.2f}')
```

```
@dp.callback_query_handler(text='binancecoin')
async def bitcoin(call: CallbackQuery):
    price = cg.get_price(ids='binancecoin', vs_currencies='usd')
    await call.answer(cache_time=60)
    await call.message.answer(f'BNB: 1 BNB = ${price[binancecoin][usd]:.2f}')
```

```
@dp.callback_query_handler(text='solana')
async def bitcoin(call: CallbackQuery):
```

```
price = cg.get_price(ids='solana', vs_currencies='usd')
await call.answer(cache_time=60)
await call.message.answer(f'Solana: 1 SOL = ${price[solana][usd]:.2f}')
```

```
@dp.callback_query_handler(text='cancel')
async def cancel(call: CallbackQuery):
    await call.message.edit_reply_markup()
```