

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра автоматизації та інтелектуальних інформаційних технологій

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Удосконалення системи обміну валют Exchanger шляхом  
імплементації боту для нотифікації системних помилок»

Виконав: студент 2 курсу, групи ІАКІТ-22м  
спеціальності 151 – Автоматизація та  
комп'ютерно-інтегровані технології

(шифр і назва спеціальності)

Ілля ЯРОВИЙ

(ПІБ студента)

Керівник: к.т.н., доцент кафедри АІТ  
Володимир ГАРМАШ

(науковий ступінь, вчене звання / посада ПІБ керівника)

«04» грудня 2023 р.

Опонент: к.т.н., доцент кафедри КН  
Руслан БЕЛЗЕЦЬКИЙ

(науковий ступінь, вчене звання / посада ПІБ опонента)

«07» грудня 2023


Допущено до захисту  
Завідувач кафедри АІТ  
Олег БІСІКАЛО  
(науковий ступінь, вчене звання)

«12» грудня 2023 року

Вінниця ВНТУ – 2023 року

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра автоматизації та інтелектуальних інформаційних технологій  
Рівень вищої освіти 2-й (магістерський)  
Галузь знань – 15 «Автоматизація та приладобудування»  
Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»  
Освітня програма – Інтелектуальні комп'ютерні системи

### ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ  
д.т.н., проф. Олег БІСІКАЛО  
  
«21» вересня 2023р.


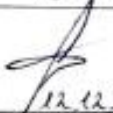
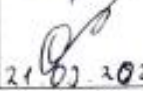

### ЗАВДАННЯ

#### НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Яровому Іллі Костянтиновичу  
(ПІБ автора повністю)

1. Тема роботи: Удосконалення системи обміну валют Exchanger шляхом імплементації боту для нотифікації системних помилок  
Керівник роботи: к.т.н., доцент каф. АІТ Гармаш В.В.  
Затвердженні наказом ВНТУ від «18» вересня 2023 року № 247.
2. Строк подання роботи студентом: до «12» грудня 2023 року.
3. Вихідні дані до роботи: API обмінника валют Exchanger, API системи технічної підтримки, стек технологій для розробки продукту; месенджер Telegram та Telegram API
4. Зміст текстової частини: Вступ; Дослідження предметної галузі; Вибір технологічних засобів для розробки боту; Розробка програмного забезпечення; Тестування програмного забезпечення; Економічний розділ; Висновки; Список використаних джерел.
5. Перелік ілюстративного (або графічного) матеріалу: Блок-схема telegram-бота; UML-діаграма класів; UML-діаграма прецедентів; UML-діаграма діяльності.

6. Консультанти розділів роботи

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1 – 4	доцент кафедри АІТ, к.т.н., Володимир ГАРМАШ	 21.05.2023	 12.12.2023
5	доцент кафедри ЕПВМ, к.е.н., Володимир КОЗЛОВСЬКИЙ.	 21.05.2023	 12.12.2023

7. Дата видачі завдання: «21» вересня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної галузі	02.10.2023	викон
2	Обґрунтування вибору технологічного стеку	09.10.2023	викон
3	Розробка програмного забезпечення	10.11.2023	викон
4	Тестування розробленого програмного забезпечення	17.11.2023	викон
5	Підготовка економічної частини	24.11.2023	викон
6	Оформлення пояснювальної записки, графічного матеріалу і презентації	04.12.2023	викон
7	Апробація результатів дослідження	08.12.2023	викон
8	Захист роботи	з 12.12.2023 по 20.12.2023	

Студент

  
(підпис)

Ілля ЯРОВИЙ

Керівник роботи

  
(підпис)

Володимир ГАРМАШ

## АНОТАЦІЯ

УДК 004.512.2

Яровий І. К. Удосконалення системи обміну валют Exchanger шляхом імплементації боту для нотифікації системних помилок. Магістерська кваліфікаційна робота зі спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології, освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2023. 83 с.

На укр. мові. Бібліогр.: 24 назв; рис.: 25; табл. 12.

Магістерська кваліфікаційна робота присвячена розробці telegram-бота для нотифікації системних помилок обмінника валют Exchanger, що дозволить автоматизувати механізм виявлення та відлагодження системних помилок даного проекту.

У даній роботі детально досліджується предметна галузь, розглядаються системи аналоги та типи чат-ботів. Окрема увага приділена вибору технологічного стеку, розробці програмного забезпечення та його тестуванню.

Ключові слова: обмінник валют, сповіщення, telegram-бот, автоматизація.

## **ABSTRACT**

Yarovy I.K. Improving the Exchanger currency exchange system by implementing a bot for system error notifications. Master's thesis on specialty 151 - Automation and computer-integrated technologies; educational program - Intelligent computer systems. Vinnytsia: VNTU, 2023. 83 p.

In Ukrainian language. Bibliography: 24 titles; Fig.: 25; table 12.

The master's qualification project is dedicated to the development of a Telegram bot for notifying about system errors in the currency exchange platform "Exchanger." This bot aims to automate the detection and debugging mechanism of systemic errors within the project.

This work extensively explores the subject domain, examining analogous systems and types of chatbots. Special attention is given to selecting the technological stack, developing the software, and testing it.

Keywords: currency exchange, notification, telegram bot, automation.

## ЗМІСТ

ВСТУП.....	4
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ГАЛУЗІ .....	6
1.1. Обмінник валют Exchanger.....	8
1.2 Обмінник валют Ramp .....	10
1.3 Обмінник валют Moonpay.....	11
1.4 Огляд типів чат-ботів .....	13
1.4.1 Чат-боти на основі меню або кнопок .....	13
1.4.2 Чат-боти на основі правил .....	14
1.4.3 Чат-боти з використанням штучного інтелекту .....	14
1.4.4 Голосові чат-боти .....	16
1.4.5 Чат-боти з генеративними можливостями штучного інтелекту.....	17
1.5 Висновки до розділу.....	17
2 ВИБІР ТЕХНОЛОГІЧНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ БОТУ .....	18
2.1 Мови програмування для розробки telegram-боту .....	19
2.1.1 JavaScript.....	19
2.1.2 Python .....	20
2.1.3 Java .....	22
2.2 Бібліотеки для створення telegram-боту .....	23
2.3 Системи управління базами даних.....	24
2.3.1 Реляційні бази даних .....	25
2.3.2 Нереляційні бази даних.....	27
2.4 Висновки до розділу.....	29
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	30

	3
3.1 Обґрунтування та проектування функціоналу telegram-боту.....	30
3.2 Проектування бази даних .....	31
3.3 Опис структури проекту та основних функцій.....	36
3.4 Висновки до розділу.....	39
4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	40
4.1 Висновки до розділу.....	48
5 ЕКОНОМІЧНИЙ РОЗДІЛ.....	49
5.1 Технологічний аудит розробленого боту для нотифікації системних помилок обмінника валют «Exchanger».....	49
5.2 Розрахунок витрат на розроблення боту для нотифікації системних помилок обмінника валют «Exchanger».....	54
5.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки.....	57
5.4 Висновки до розділу.....	63
Висновки.....	64
Список використаних джерел .....	66
Додатки .....	68
Додаток А (обов'язковий) Технічне завдання.....	69
Додаток Б (обов'язковий) Графічна частина.....	72
Додаток В (обов'язковий) Лістинг програмного забезпечення .....	76
Додаток Г (обов'язковий) Протокол перевірки роботи на наявність текстових запозичень .....	80

## ВСТУП

Швидкий розвиток інформаційних технологій в останні десятиліття створює нові можливості для взаємодії між людьми та комп'ютерними системами. Різноманітні месенджери стали важливою частиною повсякденного життя людей. Популярність месенджерів визначається не лише зручністю і швидкістю комунікації, але і можливістю інтеграції різноманітних функцій, що робить їх центральним елементом сьогодення.

Чат-боти є частиною широкого спектру застосувань в інформаційних технологіях, від онлайн-підтримки до автоматизованих систем обробки запитань. Вони можуть використовувати штучний інтелект та обробляти природне мовлення, що сприяє зручній та ефективній комунікації між користувачами та комп'ютерними системами. Окрім цього, надають можливість автоматизації завдань, вдосконалення обслуговування клієнтів і створення інтерактивних інтерфейсів для різних галузей.

Магістерська кваліфікаційна робота присвячена дослідженню методів розробки чат-ботів та безпосередньо розробці telegram-боту для нотифікації системних помилок обмінника валют.

**Актуальність** даної роботи зумовлена потребою в автоматизації та контролі за процесами обміну валют для фінансового ринку криптовалют, забезпеченні достатнього рівня безпеки фінансових операцій та зменшенні ризику витрат пов'язаними зі змінами курсів валют та іншими факторами. Розробка боту для нотифікації допоможе автоматизувати моніторинг системи і реагувати на помилки без необхідності постійного контролю оператором. Це знижує вплив людського фактора та забезпечує більш ефективну роботу обмінника валют.

Exchanger – це цифровий гаманець і онлайн-сервіс швидкого включення та відключення, який дає змогу будь-якому окремому клієнту продавати свою криптовалюту за фіксовану валюту та надсилати її на будь-який банківський рахунок чи платіжну картку або купувати та зберігати криптовалюту чи



надсилати її на будь-який інший криптосервіс.

**Метою роботи** є покращення роботи обмінника валют Exchanger шляхом імплементації бота, який автоматично відстежуватиме системні помилки та збої в роботі продукту, а також надаватиме оперативні повідомлення та звіти розробникам та службі підтримки.

**Об'єктом дослідження** є процес розробки боту для нотифікації системних помилок обмінника валют Exchanger для корпоративних месенджерів.

**Предметом дослідження** – методи та засоби розробки боту для нотифікації системних помилок обмінника валют Exchanger.

**Для досягнення мети необхідно розв'язати наступні задачі:**

- дослідити предмету галузь;
- проаналізувати технології для реалізації боту для нотифікації системних помилок обмінника валют Exchanger;
- розробити програмне забезпечення для системи нотифікації про системні помилки та збої за допомогою телеграм-боту;
- провести тестування розробленого програмного продукту.

**Методи дослідження.** В процесі дослідження застосовуються аналітичні методи та методи розробки ботів для месенджерів.

**Наукова новизна одержаних результатів.** Розроблений telegram-бот вирішує конкретну проблему оперативного виявлення та вирішення системних помилок в обміннику валют, що робить його науково новаторським в контексті підвищення надійності та якості функціонування подібних фінансових систем.

**Практичною цінністю роботи** є розроблене програмне забезпечення, що дозволить підвищити контроль якості фінансових операцій та автоматизувати процес обміну крипто та фіатних валют.

## 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ГАЛУЗІ

Обмінники валют є ключовою частиною сучасного фінансового ринку та глобальної економіки. Вони забезпечують можливість конвертувати одну валюту в іншу, що важливо для міжнародної торгівлі, подорожей і багатьох інших аспектів життя. Системи обмінника валют дозволяють користувачам швидко та зручно здійснювати валютні операції. Вони знижують ризики валютних коливань і сприяють збереженню капіталу. Крім того, обмінники валют можуть бути доступні для користувачів у різних частинах світу і підтримувати різні методи оплати.

Обмінники валют також відіграють важливу роль у фінансовій стабільності, допомагаючи у вирішенні питань ліквідності та забезпечуючи сталість валютних курсів.

Бот – це спеціальна програма, яка за заданим розкладом або автоматично виконує певні дії через ті ж інтерфейси, що і звичайний інтернет-користувач. Призначена для виконання одноманітних і повторюваних процедур з максимальною швидкістю і точністю [11]. Боти можуть обробляти дані, транслювати їх, а також відправити команди іншим пристроям чи системам. В сучасному світі боти використовуються не лише для автоматизації завдань, але й у штучному інтелекті, клієнтському обслуговуванні та інших галузях.

Розглянемо платформу Telegram на базі якої розроблено програмний продукт у вигляді боту.

Telegram – це популярний додаток, який дозволяє швидко та безпечно здійснювати обмін повідомленнями та має широкий безкоштовний базовий функціонал. Окрім цього, надає API для розробки програмних продуктів та забезпечує синхронізацію повідомлень між різними з'єднаннями. Станом на 2023 рік Telegram входить до 10 найбільш використовуваних додатків у світі, маючи понад 700 мільйонів активних користувачів щомісяця. Telegram надає широкі можливості для розробки різноманітних ботів та інших автоматизованих сервісів, що робить його популярним серед розробників та користувачів. [4].

Імплементація telegram-боту для нотифікації системних помилок обмінника валют допоможе знизити витрати та покращити його ефективність, особливо в умовах великого обсягу операцій, а також підвищить рівень безпеки системи, сприяючи оперативному реагуванню на можливі проблеми через автоматизовані сповіщення.

Завдяки імплементації боту для нотифікації системних помилок обмінника валют підвищується ефективність та швидкість проведення валютних операцій, а також робота технічної підтримки, так як дана автоматизація дозволяє зробити роботу більш ефективною та допомагає скоротити час і витрати на взаємодію з клієнтами, що забезпечує покращення репутації обмінника валют та збільшення лояльності клієнтів через зручність та оперативність обслуговування.

Таким чином, впровадження telegram-боту для нотифікації системних помилок обмінника валют значно покращує комунікацію з клієнтами і допомагає в ефективному управлінні операціями обміну валют, що призводить до збільшення прибутковості і конкурентоспроможності бізнесу.

Використання криптовалют має значний вплив на обмінники валют та фінансовий сектор загалом. Криптовалюти дозволяють людям здійснювати фінансові операції без прив'язки до конкретного регіону чи банківської системи. Крім того, технологія блокчейну, на якій ґрунтуються багато криптовалютних операцій, підвищує рівень безпеки та надійності транзакцій, дозволяє проводити операції швидше, оскільки вони не обмежені банківськими годинами роботи чи географічними межами.

Варто зазначити, що впровадження криптовалют в обмінні операції ставить перед обмінниками валют ряд викликів, такі як висока волатильність курсів, потенційні загрози кібербезпеки, необхідність вирішення питань легітимності та довіри, регулювання та законодавчі аспекти, можливі технічні проблеми та питання конфіденційності, що вимагає від обмінників розробки стратегій управління ризиками, впровадження ефективних кіберзахистних заходів, дотримання законів та нормативів, покращення технічної стійкості та конфіденційності.

Розглянемо основні відмінності системи обмінника валют Exchanger від його найбільш відомих аналогів – Ramp та MoonPay. В процесі порівняння системи обмінника валют Exchanger зі своїми конкурентами, важливо розглянути різноманітні аспекти, які визначають їхню конкурентну перевагу чи унікальність. У вільному доступі інформація про аналоги розроблюваного телеграм-боту відсутня.

### **1.1. Обмінник валют Exchanger**

Exchanger – це цифровий гаманець і онлайн-сервіс швидкого включення/відключення, який дає змогу будь-якому окремому клієнту продавати свою криптовалюту за фіксовану валюту та надсилати її на будь-який банківський рахунок чи платіжну картку, або купувати та зберігати криптовалюту чи надсилати її на будь-який інший криптосервіс. На даний момент система знаходиться на стадії активної розробки та тестування.

Продукт має повноцінну функціональність гаманця на додаток до сервісу підвищення рівня обслуговування, який забезпечує найкращі можливі динамічні коефіцієнти конверсії з широким вибором підтримуваних методів оплати, що дозволяє не лише купувати чи продавати криптовалюту, але також зберігати та конвертувати внутрішні баланси, робити поповнення та виплати. Усе це можна легко адаптувати до будь-якої юрисдикції, що дозволяє здійснювати перевірку користувача відповідно до будь-яких вимог.

Основний функціонал та особливості обмінника валют Exchanger:

- вхід та реєстрація користувача;
- верифікація облікового запису користувача;
- верифікація способу оплати користувача;
- обмін валюти;
- конфігурація BestChange як джерело курсу для обміну;
- поповнення балансу;
- виплата з балансу;

- переказ іншому користувачу системи;
- конвертування коштів на внутрішньому балансі;
- перегляд і керування балансами валют;
- історія транзакцій;
- керування криптоадресами;
- керування списками ризиків;
- партнерська програма;
- керування обліковими записами через API.

На рисунку 1.1 представлено головну сторінку проекту Exchanger з функціоналом для проведення обмінної валютної операції.

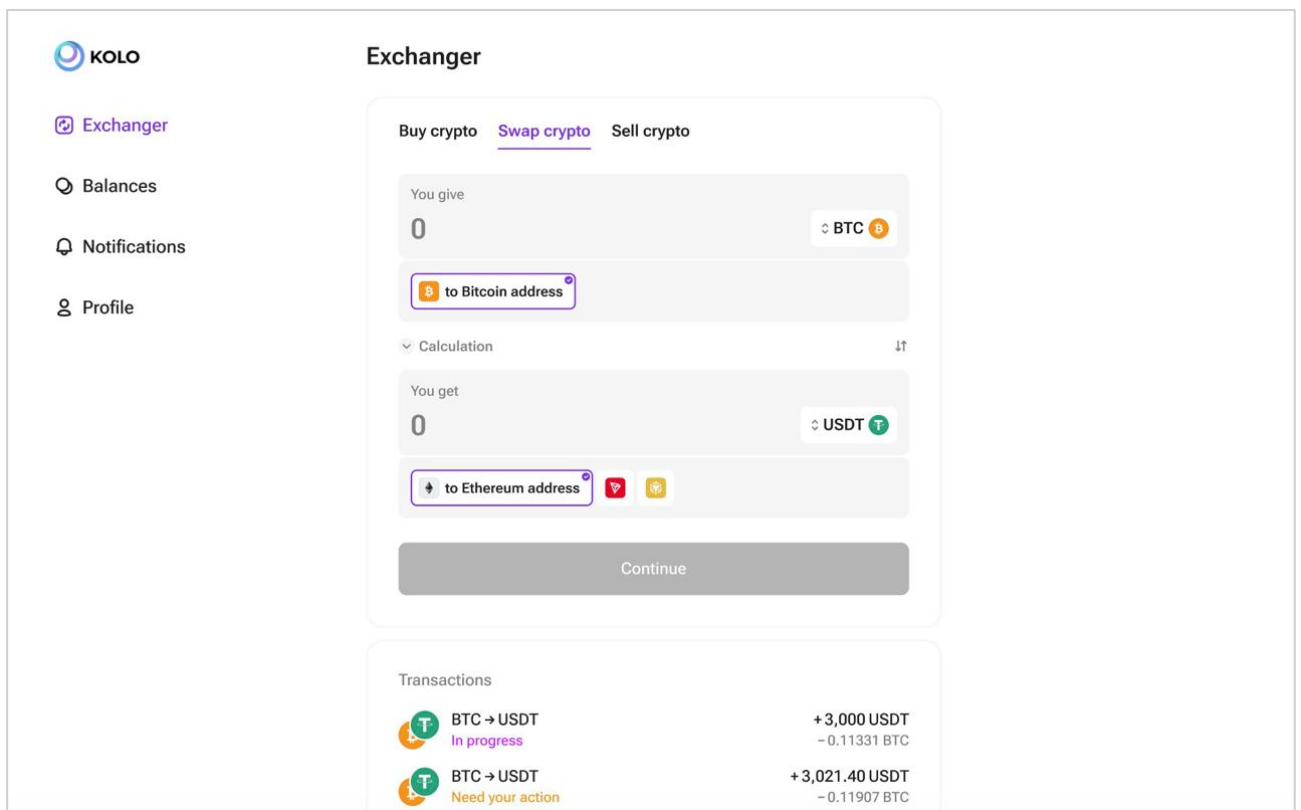


Рисунок 1.1 – Обмінник валют Exchanger

Особливістю даного обмінника фіатних та криптовалют є можливість підключення до одного профілю декількох користувачів, що корисно для компаній та фінансових організацій.

Однією із ключових переваг даної системи є високий рівень безпеки та конфіденційність користувачів. Їхні особиті дані, такі як номери карт, CVV коди та реквізити IBAN не зберігаються у системі, що забезпечується використанням спеціального мікросервісу з хешуванням під час проведення валютної операції та негайним видаленням захешованих даних після її завершення.

## 1.2 Обмінник валют Ramp

Ramp - це глобальна фінтех-компанія, яка розробляє рішення, що з'єднують криптоекономіку з сучасною фінансовою інфраструктурою забезпечуючи спрощений та плавний досвід конвертації між криптовалютами та фіатними валютами [4]. Основною метою компанії є забезпечення доступності web3 для всіх користувачів та створення процесу купівлі та продажу криптовалюти простим та безпечним.

Одним із продуктів компанії є обмінник криптовалют. Розглянемо переваги продукту:

- Підтримка великої кількості платіжних методів, включаючи дебетові та кредитні картки, банківські перекази, Apple Pay, Google Pay, з можливістю додавання інших методів оплати у майбутньому.
- Співпраця з регуляторами та дотримання стандартів, що забезпечує високий рівень безпеки проведення валютних операцій;
- Інтуїтивно зрозумілий та привабливий інтерфейс користувача;
- Наявність API для інтеграції обмінника валют з різноманітними проектами.

Недоліками даного сервісу є:

- Високий рівень комісії за проведення операції, що може призвести до затримок у прийнятті рішень та зменшення конкурентоспроможності;
- Географічна обмеженість використання обмінника.

На рисунку 1.2 представлено приклад інтерфейсу користувача обмінника, який відзначається не лише естетичною привабливістю, але й простотою використання, що робить його досить пристосованим для широкого кола користувачів.

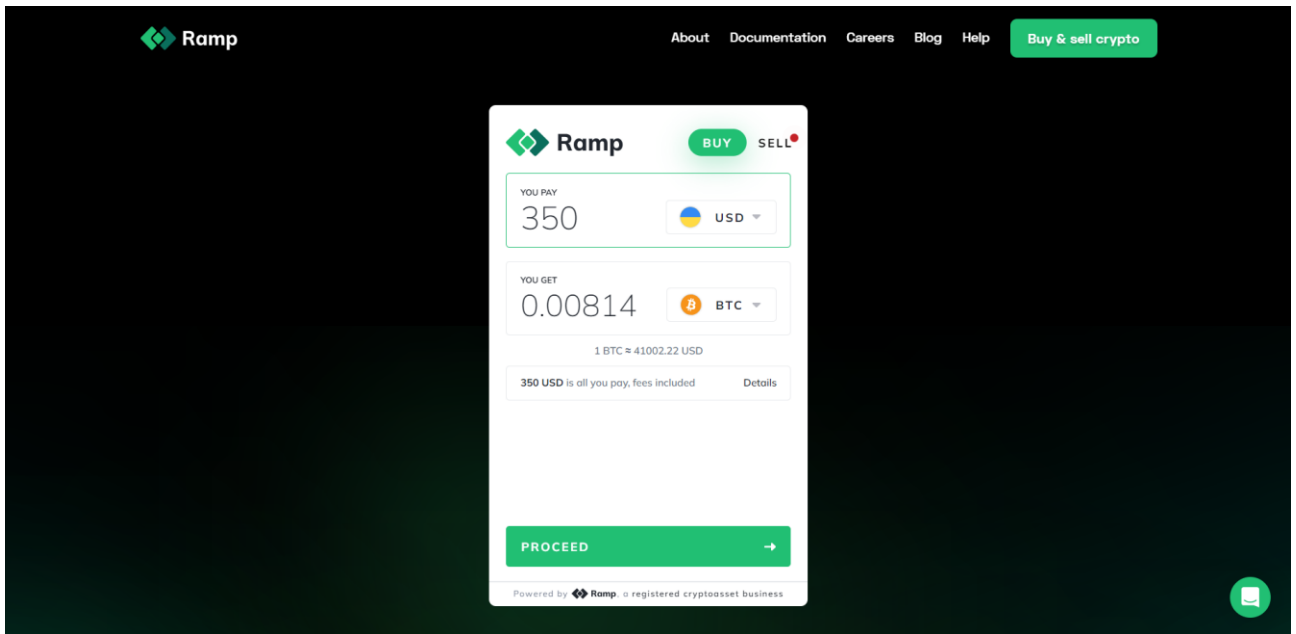


Рисунок 1.2 – Обмінник криптовалют Ramp

У даному обміннику валют відсутній telegram-бот для нотифікації технічної підтримки про системні збої під час проведення валютної операції, однак для власних проєктів є можливість його розробки з використанням наданим компанією Ramp API.

### 1.3 Обмінник валют Moonpay

MoonPay - це платформа для обміну криптовалют, яка надає можливість користувачам придбати криптовалютні активи шляхом використання традиційних фіатних методів оплати, таких як кредитні картки та банківські перекази [5].

До переваг даного обмінника можна віднести:

- підтримка різних методів оплати, включаючи кредитні та дебетові картки, банківські перекази та інші фіатні способи оплати;
- підтримка клієнтів і можливість працювати з різними мовами та валютами в залежності від регіону користувача;
- інтуїтивно зрозумілий та простий у використанні інтерфейс;

Недоліками Moonpay є:

- взаємодія з третіми сторонами, такими як банки чи платіжні партнери, що створює ризики та затримки в обробці операцій через залежність від цих сторін;
- обмеження на максимальні та мінімальні обсяги обміну;
- географічні обмеження, включаючи відсутність змоги проводити валютні операції на території України.

На рисунку 1.3 представлено користувацький інтерфейс обмінника валют Moonpay.

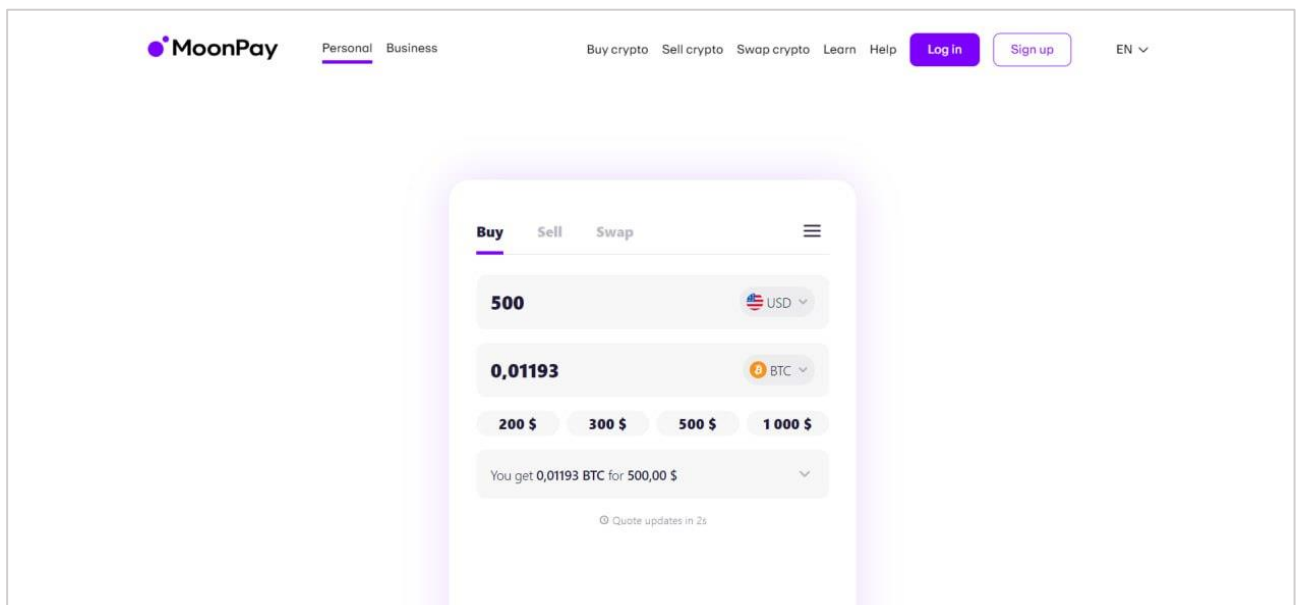


Рисунок 1.3 – Обмінник валют Moonpay

Інформація про наявність telegram-бота для нотифікації про системні помилки обмінника валют Moonpay у відкритому доступі відсутня, як і інтеграція з власним проектом за наданим API.



## 1.4 Огляд типів чат-ботів

Програмні агенти, або боти, стали невід'ємною складовою нашого цифрового повсякдення, активно функціонуючи в різних областях та виконуючи різноманітні завдання. Виявивши свою ефективність не лише в автоматизації рутинних операцій, вони також внесли суттєвий вклад у перетворення способу, яким ми взаємодіємо з технологіями, граючи ключову роль у численних аспектах нашого сучасного життя.

Розглянемо основні типи ботів, які використовуються для месенджерів [9].

### 1.4.1 Чат-боти на основі меню або кнопок

Чат-боти, які базуються на меню чи кнопках, є найпростішим видом чатботів, де користувачі можуть взаємодіяти з ними, натискаючи кнопку із задалегідь скриптованого меню, що найкраще відображає їхні потреби. Залежно від того, на що натиснув користувач, простий чатбот може запропонувати ще один набір варіантів для вибору, поки не буде досягнуто найбільш відповідного, конкретного варіанту. В основному дані чатботи працюють як дерево рішень.

Незважаючи на те, що чатботи на основі меню чи кнопок пропонують простий функціонал є корисними для відповіді на повторювані, прямолінійні запитання користувачів, вони можуть зазнавати труднощів при більш витончених запитаннях через обмеженість задалегідь визначеними варіантами відповідей.

Основні недоліки даного виду чат-ботів:

- тривалий процес розуміння потреб користувача, особливо якщо користувач повинен пройти кілька ітерацій натискання кнопок меню.
- відсутність можливості відповіді на питання користувача, яке не включене у перелік питань меню;
- Можливий негативний досвід користувача через обмеженість функціоналу чат-боту.

### **1.4.2 Чат-боти на основі правил**

Даний вид чат-ботів є розвитком функціоналу простого дерева вибору чат-бота. У типі бота на основі правил використовується умовна логіка "якщо/то" для створення потоків автоматизованого спілкування. Ці боти на основі правил діють як інтерактивні FAQ, де архітектор діалогів програмує заздалегідь визначені комбінації питань і відповідей, щоб чат-бот міг розуміти введення користувача і надавати точні відповіді.

Діючи на основі простого виявлення ключових слів, чат-боти на основі правил досить легкі у навчанні і працюють добре, коли задаються заздалегідь визначені питання. Однак, подібно жорстким чат-ботам на основі меню, вони неспроможні відповісти на складні запитання. Дані чат-боти мають труднощі з відповіддю на питання, які не були передбачені дизайнером розмов, оскільки їх вивід залежить від наперед написаного контенту, програмованого розробниками бота.

Оскільки неможливо передбачити і попередньо запрограмувати чат-бот для всіх видів запитань користувачів, часто він не може зрозуміти запит користувача. У такому випадку, бот пропускає важливі деталі та просить користувача повторити інформацію, яка вже була надана. Це призводить до неприємного досвіду користувача.

### **1.4.3 Чат-боти з використанням штучного інтелекту**

У той час як чат-бот на основі правил підтримує лише заздалегідь визначені питання та варіанти відповідей, чат-боти з використанням штучного інтелекту можуть розуміти запитання користувача, незалежно від того, як вони сформульовані. Використання штучного інтелекту та здатності розуміння природної мови (Natural Language Understanding, NLU) чат-бот може швидко виявляти всю необхідну контекстуальну інформацію, надану користувачем, що забезпечує безперервну та наближену до природньої розмови відповідь на

питання. У разі виявлення більше однієї дії, яка може задовольнити запит користувача, чат-бо може задати додаткові уточнюючі питання. Крім того, показати список можливих дій, із якого користувач може вибрати опцію, яка відповідає його потребам.

Алгоритми машинного навчання, що лежать в основі чат-ботів з використанням штучного інтелекту, дозволяють їм самостійно навчатися та розвивати все більш інтелектуальну базу знань на основі питань та відповідей, що ґрунтуються на взаємодії з користувачами. За допомогою глибокого навчання, чим довше чат-бот працює, тим краще він може розуміти мету користувача і надавати більш детальні та точні відповіді.

Чат-боти з використанням штучного інтелекту можуть пам'ятати розмови з користувачами та враховувати цей контекст у своїх взаємодіях. У поєднанні з можливостями автоматизації, такими як роботизована автоматизація процесів (Robotic Process Automation, RPA), користувачі можуть виконувати завдання через досвід чат-бота. Наприклад, при замовленні піци чат-бот ресторану може визнати лояльного клієнта, який повертається робити замовлення, вітати його за іменем, пам'ятати його "звичайне" замовлення та використовувати збережену адресу доставки та кредитну картку для завершення замовлення. Завдяки глибокій інтеграції з системами бізнесу чат-бот з використанням штучного інтелекту може отримувати інформацію з різних джерел, яка містить історію замовлень клієнта, та створювати оптимізований процес замовлення.

Час, необхідний для створення чат-бота з використанням штучного інтелекту варіюється в залежності від наступних факторів:

- технічний стек та засоби розробки;
- складність чат-бота;
- необхідність інтеграції з іншими системами, базами даних чи платформами.

Вагомою перевагою даного виду чат-ботів є можливість переходу до спілкування з технічною підтримкою непомітно для користувача, якщо подальша відповідь на питання ботом неможлива.

#### 1.4.4 Голосові чат-боти

Даний тип чат-ботів представляє собою інтерактивний інструмент для розмов, який використовує штучний інтелект та розуміння природної мови (NLU) для інтерпретації намірів та значень у командах, висловлених усно користувачем.

Голосові чат-боти з використанням штучного інтелекту можуть надавати ті ж самі вдосконалені можливості, що й звичайні чат-боти на основі штучного інтелекту, однак використовують голосові канали і технологій перетворення тексту в мовленнєві сигнали та навпаки.

Голосові боти розуміють природну мову та можуть ініціювати або брати участь у двосторонньому спілкуванні з користувачами. За допомогою команд користувачів голосові боти використовують модулі для прослуховування, розуміння та навчання протягом свого використання.

Розглянемо технології та процеси необхідні для розробки інтелектуального голосового чат-боту:

- Перетворення мовлення на текст (Speech to Text). Технологія визнання мовлення та його переклад, відома як "speech-to-text" (STT), перетворює природне мовлення на текст, розпізнаючи різні акценти та мови.
- Текст в мовлення (Text to Speech). За допомогою двигуна тексту в мовлення (TTS), голосовий бот перекладає текст в синтезоване мовлення.
- Двигун розуміння природної мови (NLU). Дана технологія допомагає ботам розуміти питання, слова та речення для забезпечення можливості користувачу вільно висловлюватися та отримувати миттєві відповіді, не обмежуючись лише "так" чи "ні".
- Визначення мови (Language detection). Мова та діалект користувача розпізнаються, і голосовий бот автоматично переходить до цієї мови.
- Машинне навчання (Machine Learning). Голосовий бот вивчає питання та дані користувача, додаючи ці знання до бази знань голосового бота.

### **1.4.5 Чат-боти з генеративними можливостями штучного інтелекту**

Наступне покоління чат-ботів, обладнаних генеративними можливостями штучного інтелекту, володіє значно розширеними функціональними можливостями завдяки їх здатності розуміти загальну мову, адаптуватися до індивідуального стилю спілкування користувача та використовувати емпатію при формулюванні відповідей на їх запитання. У порівнянні з чат-ботами, що використовують розмовний штучний інтелект, які можуть обробляти питання чи коментарі користувача та генерувати відповіді, схожі на людські, чат-боти із генеративним штучним інтелектом виходять за межі, генеруючи новий контент як результат.

Цей новий контент може включати в себе текст високої якості, зображення та звук, створені на основі обширних мовних моделей, на яких вони були навчені. Інтерфейси чат-ботів з генеративним штучним інтелектом демонструватимуть здатність впізнавання, узагальнення, перекладу, передбачення та творення контенту у відповідь на запитання користувача, уникнувши при цьому необхідності активної взаємодії з людиною.

## **1.5 Висновки до розділу**

В першому розділі роботи було розглянуто сутність чат-ботів, проведено аналіз обмінників валют, зокрема Exchanger та його конкурентів Ramp і Моопрау. Під час аналізу вивчалися основні характеристики цих платформ, такі як принцип їх роботи, доступність валютних пар, комісії та обмінні курси. Окрема увага була приділена інтерфейсам користувача та наявності нотифікаційних чат-ботів. Також розглянуто основні типи чат-ботів, їхні особливості та можливості.

Завдяки telegram-боту боту, можливо оперативно виявляти системні помилки та передавати отримані данні службі підтримки, що сприяє підвищенню надійності та ефективності роботи обмінника валют Exchanger.

## 2 ВИБІР ТЕХНОЛОГІЧНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ БОТУ

Розробка чат-ботів для месенжеру Telegram здобула популярність завдяки своїй простоті та ефективності. Це стало можливим завдяки доступності відкритого API та широким можливостям налаштування та інтеграції, які надають розробникам відмінні засоби для створення чат-ботів різних типів.

Даний розділ присвячено огляду технологічних засобів для розробки чат-ботів, а також обґрунтуванню їхнього вибору для розробки telegram-боту для нотифікації системних збоїв та помилок.

Telegram - це хмарний месенджер з безшовною синхронізацією, який забезпечує доступ до листувань користувача з декількох пристроїв одночасно, включаючи планшети та комп'ютери, а також дозволяє обмінюватися необмеженою кількістю фотографій, відео та файлів різноманітних форматів розміром до 2 ГБ кожен [24].

Платформа надає можливість обміну текстовими та мультимедійними повідомленнями, а також підтримує групові чати та канали.

Розглянемо переваги платформи Telegram для розробки чат-боту:

- наявність доступного API з детальною документацією для розробки різних типів чат-ботів;
- легкість досягнення цільової аудиторії та забезпечення швидкої та ефективної комунікації з користувачами завдяки високій популярності месенджера;
- високий рівень конфіденційності та захисту даних завдяки використанню унікальної комбінації протоколу MTProto та наскрізного шифрування;
- можливість інтеграції із зовнішніми службами (Gmail bot, GitHub bot, Youtube bot та інші).

Станом на 2023 рік Telegram посідає третє місце серед найбільш популярних месенджерів у світі та займає перше місце в Україні [24].

## 2.1 Мови програмування для розробки telegram-боту

Для розробки telegram-боту необхідно обрати мову програмування залежно від особливостей розроблюваного продукту та його вимог. Розглянемо найбільш розповсюдженні мови програмування, які дозволяють реалізувати telegram-бот.

### 2.1.1 JavaScript

JavaScript є високорівневою, інтерпретованою мовою програмування, що використовується для розробки веб-застосунків, серверного програмного забезпечення, а також може використовуватися для розробки ботів для різних месенджерів, у тому числі і для Telegram.

Розробка telegram-боту з використанням JavaScript можлива на базі платформи Node.js.

Розглянемо переваги та недоліки даної мови програмування для розробки чат-ботів. Перевагами JavaScript є:

- полегшена інтеграція бота з веб-серверами, базами даних та іншими веб-технологіями.
- наявність великої кількості бібліотек та модулів, що сприяє швидкому розвитку та впровадженню функціоналу бота.
- можливість використання асинхронного програмування, що дозволяє ефективно обробляти багато запитів одночасно.

До недоліків JavaScript для розробки telegram-боту можна віднести:

- менша ефективність для важких обчислень або завдань, які вимагають великої обробки даних.
- слабка типізованість мови, що може призвести до помилок, які важко виявити на етапі компіляції.
- обмежений вибір серверного середовища для виконання JavaScript.

## 2.1.2 Python

Python - це інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Вбудовані високорівневі структури даних, разом із динамічною типізацією та динамічним з'єднанням є однією з головних причин вибору даної мови програмування для розробки програм та використання як скриптової або з'єднуючої мови для інтеграції існуючих компонентів. Python підтримує концепції модулів та пакетів, сприяючи модульності програм [1].

За даними дослідження ТЮВЕ станом на листопад 2023 року Python очолює рейтинг найбільш використовуваних мов програмування [6]. На рисунку 2.1 наведено десятку лідерів даного дослідження.











Nov 2023	Nov 2022	Change	Programming Language	Ratings	Change
1	1		 Python	14.16%	-3.02%
2	2		 C	11.77%	-3.31%
3	4	▲	 C++	10.36%	-0.39%
4	3	▼	 Java	8.35%	-3.63%
5	5		 C#	7.65%	+3.40%
6	7	▲	 JavaScript	3.21%	+0.47%
7	10	▲	 PHP	2.30%	+0.61%
8	6	▼	 Visual Basic	2.10%	-2.01%
9	9		 SQL	1.88%	+0.07%
10	8	▼	 Assembly language	1.35%	-0.83%

Рисунок 2.1 – Рейтинг популярності мов програмування за даними ТЮВЕ Index for November 2023

Основні переваги мови програмування Python для розробки telegram-боту для нотифікації системних помилок:

- зрозумілий синтаксис, що забезпечує високу швидкість розробки продукту;



- наявність бібліотек для взаємодії з Telegram API та реалізації функцій чат-бота;
  - Python є інтерпретованою мовою програмування, що припиняє роботу програми у разі виявлення помилки, забезпечуючи зручність відлагодження коду та пошук слабких місць системи;
  - можливість інтеграції з іншими системами та мовами програмування.
- До недоліків даної мови програмування можна віднести наступні:
- обмежена підтримка паралельного програмування порівняно з іншими мовами;
  - залежність від сторонніх модулів та бібліотек.

У таблиці 2.1 наведено порівняльну характеристику мови програмування Python та JavaScript, яка розглянута у попередньому пункті.

Таблиця 2.1 – Порівняльна характеристика Python та JavaScript

<b>Критерій</b>	<b>Python</b>	<b>JavaScript</b>
Область застосування	Розробки чат-ботів та веб-застосунків, наукові дослідження, машинне навчання, аналіз даних.	Розробка веб-застосунків, можливість розробки ботів з використанням платформи Node.js
Типізація даних	Динамічна типізація	Динамічна типізація
Паралельне програмування	Обмежена підтримка паралельного програмування через Global Interpreter Lock (GIL), що може впливати на продуктивність системи.	Виконання асинхронного коду код завдяки концепції Event Loop (циклу подій)
Синтаксис	Інтуєтивно зрозумілий, що полегшує розробку та підтримку коду	Гнучкий, вимагає більше часу на розуміння

Бібліотеки Python, які значно пришвидшують та спрощують розробку чат-ботів, представляють собою найбільшвикористовувані фрагменти коду, які розробники можуть включати у свої програми. За замовчуванням у Python є стандартна бібліотека, окрім якої доступно понад 137 тисяч бібліотек як розв'язання різних типів задач.

### 2.1.3 Java

Java – це об'єктно-орієнтована мова програмування з класовою структурою та статичною типізацією. Основною перевагою даної мови є принцип `write once, run anywhere (WORA)`, що означає можливість виконання коду на всіх платформах, які її підтримують без необхідності повторної компіляції.

Особливості мови програмування Java:

- незалежність від платформи, що забезпечується перетворенням коду у байткод компілятором з подальшим його виконанням віртуальною машиною Java (JVM);
- надійність, завдяки збірці сміття (`garbage collection`), обробці винятків та розподілу пам'яті;
- високий рівень безпеки;
- підтримка батопоточності, що дозволяє одночасне виконання двох чи більше частин програми для максимально ефективного використання центрального процесору;
- архітектура Java визначена таким чином, що зменшує накладні витрати під час виконання, а використання компілятора `Just In Time (JIT)`, де компіляція коду відбувається лише за потребою, компілюючи лише ті методи, які викликаються забезпечує максимальну швидкодію додатку.

Мова програмування Java використовується для розробки ігор, веб-додатків, мобільної розробки, у галузі інтернету речей, штучного інтелекту та машинного навчання. Наявність бібліотек для інтеграції з Telegram API дозволяє також використовувати її для розробки чат-ботів.

## 2.2 Бібліотеки для створення telegram-боту

Розробка та впровадження чат-ботів у Telegram відкриває широкі можливості для взаємодії з користувачами. Однак, без зручних інструментів процес розробки може бути довгоривалим та технічно складним. Рішенням даної проблеми є використання бібліотек для розробки telegram-ботів, які забезпечують взаємодію з Telegram API та Telegram Bot API, пришвидшують і спрощують процес розробки.

Розглянемо найбільш популярні бібліотеки та проведемо їхній порівняльний аналіз.

Бібліотека `python-telegram-bot` підтримує асинхронне виконання коду, надає зручні засоби для обробки вхідних повідомлень, команд та подій. Вагомою перевагою даної бібліотеки є можливість інтеграції боту із зовнішніми сервісами, що особливо важливо для чат-боту для нотифікації системних помилок обмінника валют Exchanger, так як використання бібліотеки дозволить встановити з'єднання з самим обмінником, системою технічної підтримки та базою даних.

Бібліотека `aiogram` є універсальним інструментом для розробки telegram-ботів, який забезпечує високу продуктивність роботи, дозволяє працювати з асинхронними запитами та має широкий функціонал. Варто зазначити, що під час роботи з даною бібліотекою можуть виникнути труднощі, так як `aiogram` вимагає глибокого розуміння принципів асинхронного програмування та детального налаштування.

Бібліотека `Telebot` надає зручний та ефективний інтерфейс для взаємодії з Telegram Bot API. Дана бібліотека дозволяє створювати модульні чат-боти та досить легко розширювати їхній функціонал, має об'ємну базу прикладів використання і дозволяє почати розробку без попереднього глибокого вивчення.

У таблиці 2.2 представлено порівняння ключових можливостей розглянутих бібліотек, на основі якої прийнято рішення використовувати `telebot` для розробки боту нотифікації системних помилок обмінника валют Exchanger.

Таблиця 2.2 – Порівняння бібліотек для розробки telegram-боту

Параметр	Telebot	aiogram	python-telegram-bot
Продуктивність	середня	висока	висока
Функціональність	базова	повна	повна
Асинхронність	підтримується	підтримується	підтримується
Повнота документації	детальна	базова	базова
Підтримка спільноти	підтримується	підтримується	підтримується
Інтеграція із зовнішніми системами	+	+	+

Так як для розробки telegram-боту для нотифікації системних помилок обмінника валют достатньо базової функціональності було обрано бібліотеку Telebot, яка повністю покриває поставлені вимоги та має детальну документацію.

### 2.3 Системи управління базами даних

Однією з ключових складових ефективної та надійної роботи чат-ботів є належне керування та збереження даних. Системи управління базами даних є основним інструментом, забезпечуючи зберігання, організацію та швидкий доступ до інформації.

Розглянемо два ключові підходи до проектування структури бази даних: реляційний та нереляційний. Кожен із цих підходів має свої особливості та переваги.

Основою реляційної бази даних є таблиці та зв'язки між ними. За допомогою SQL можна виконувати запити, які отримують дані з однієї чи декількох таблиць.

Дані у нереляційній базі даних зберігаються без жорсткої структури та чітких зв'язків між собою.

### 2.3.1 Реляційні бази даних

В сучасному інформаційному середовищі реляційні бази даних відіграють важливу роль у забезпеченні ефективного зберігання, організації та обробки даних. Реляційність ґрунтується на концепції таблиць, взаємозв'язків та структурованості даних. Кожна таблиця представляє собою сутність, а відносини між таблицями дозволяють ефективно організовувати та взаємодіяти з інформацією.

Процес нормалізації бази даних є ключовим для забезпечення цілісності та усунення надмірності в даних. Проходження через нормальні форми дозволяє досягти ефективної структуризації та уникнути аномалій під час операцій з базою даних.

Розглянемо особливості реляційних баз даних:

- для взаємодії з даними використовується мова SQL;
- строго типізована визначена заздалегідь модель даних;
- взаємозв'язки між таблицями встановлюються за допомогою ключів;
- підтримка транзакції, що забезпечують атомарність, консистентність та ізоляцію операцій.

Між таблицями можуть встановлюватися бінарні, тернарні і n-арні зв'язки. При зв'язуванні двох таблиць виділяють основну і підпорядковану таблиці, логічне зв'язування яких відбувається за допомогою ключа зв'язку.

Доступ до даних реляційної бази даних здійснюється за допомогою реляційних систем керування базами даних. Найбільш відомими є Microsoft SQL Server, MySQL та PostgreSQL.

Microsoft пропонує SQL Server у чотирьох основних редакціях, які забезпечують різні рівні включених сервісів. Повнофункціональна версія для розробників є безкоштовною, а Express Edition може використовуватися для запуску невеликих баз даних з об'ємом дискового простору до 10 ГБ. Редакція для розробників не ліцензується для виробничого використання. Додатково,

програми більшого обсягу, які вимагають підтримки на рівні виробництва, ліцензуюються як Enterprise Edition. У Standard Edition зменшений набір функцій та обмежена масштабованість за рахунок обмеження кількості використовуваних ядер центрального процесора та об'єму пам'яті.

MySQL - це система управління реляційними базами даних (RDBMS), розроблена Oracle і заснована на мові структурованих запитів SQL. MySQL є безкоштовною системою керування базами даних для некомерційного використання. Для оптимізації продуктивності MySQL використовує пул з'єднань та кешування запитів.

У таблиці 2.3 розглянуто основні відмінності Microsoft SQL Server та MySQL.

Таблиця 2.3 – основні відмінності між Microsoft SQL Server та MySQL

<b>Параметр</b>	<b>Microsoft SQL Server</b>	<b>MySQL</b>
Розробник	Microsoft	Oracle
Підтримка платформ	Менша підтримка	Більша підтримка
Масштабування	Вищий рівень масштабування	Менший рівень масштабування
Безпека	Відсутня можливість редагування бази даних під час виконання коду	Наявна можливість редагування бази даних під час виконання коду
Ліцензування	Платне програмне забезпечення	Відкритий вихідний код

PostgreSQL представляє собою систему управління реляційними базами даних із відкритим вихідним кодом, яка працює за моделлю клієнт-сервер. Вона використовує мову SQL для обробки даних і виконання запитів, доступна за допомогою різних протоколів, таких як TCP/IP і Unix-доменні сокети. Особливістю даної системи управління базами даних є підтримка як реляційних, так і нереляційних запитів.

### 2.3.2 Нереляційні бази даних

Дані у нереляційній базі даних зберігаються без чітких зв'язків та структури. Нереляційні бази даних використовують мову неструктурованих запитів NoSQL.

У таблиці 2.4 наведено порівняльну характеристику реляційних та нереляційних баз даних [10].

Таблиця 2.4 – Порівняння реляційних та нереляційних баз даних.

Параметр	Реляційні БД	Нереляційні БД
Модель даних	Реляційні моделі спеціалізуються на нормалізації даних та їх поданні у вигляді таблиць. У таблицях дані розподіляються між рядками та стовпцями, визначаються індекси та зв'язки між таблицями з строгою специфікацією, що забезпечує цілісність залежностей між даними в різних таблицях.	Моделі даних бувають різних типів (ключ-значення, документоорієнтована та заснована на графах) і мають гнучку структуру.
ACID	Забезпечують властивості ACID: консистентність, атомарність, ізольованість та надійність.	Пом'якшення вимог ACID для можливості використання гнучкої моделі даних та горизонтального масштабування
Продуктивність	Вплив на продуктивність бази даних значно залежить від характеристик дискової підсистеми. Необхідна нормалізація бази даних для забезпечення оптимальної продуктивності та оптимізація запитів.	Розмір кластера визначає кількість серверів чи вузлів, які обробляють дані, і може впливати на швидкодію та масштабованість системи.
Масштабування	Можливе завдяки збільшенню обчислювальних можливостей апаратного забезпечення.	Масштабування на основі розподіленої архітектури, що забезпечує високу продуктивність у великих масштабах.
API	Використання мови SQL для створення запитів та керування даними.	Можливість виконання пошуку за допомогою пар "ключ-значення", наборів стовпців або частково структурованих документів.

Розглянемо основні типи нереляційних баз даних: "ключ-значення", документоорієнтовану та граф.

Бази даних, що використовують пари "ключ-значення", відзначаються високою роздільністю та можливістю горизонтального масштабування. Ігрові, рекламні додатки та застосунки Інтернету речей (IoT) є типовими прикладами областей використання таких баз даних "ключ-значення". Прикладами систем керування базами даних даного типу є MemcacheDB, Redis і Berkeley DB.

Документоорієнтовані бази даних використовуються для зберігання та управління даними в форматі об'єктів або документів, часто у формі JSON, що відповідає ефективній та інтуїтивно зрозумілій моделі даних для розробників. У кодї програми дані представлені у вигляді структур, аналогічних об'єктам чи документам, що полегшує їхню взаємодію та обробку. Документні бази даних особливо ефективні у випадках, коли дані можуть бути унікальними для кожного документу та піддаються змінам з плином часу, як, наприклад, в системах каталогізації та управління користувацькими профілями. Прикладами систем управління базами даних документоорієнтованого типу є CouchDB, MarkLogic та MongoDB.

Графові бази даних відзначаються тим, що полегшують розробку та ефективний запуск додатків, які обробляють складнопов'язані дані. Такі бази даних використовують графову модель для представлення відносин між різними елементами, що сприяє зручній обробці та аналізу складних взаємозв'язків. Типові приклади використання графових баз даних включають Соціальні мережі, та графи знань. Прикладами систем управління графовими базами даних є ArangoDB, OrientDB, HyperGraphDB та Neo4j.

Кожен з розглянутих типів нереляційних баз даних має свої особливості, які роблять їх ефективними в різних сценаріях використання. Вибір конкретного типу нереляційної бази даних залежить від унікальних вимог та потреб розроблюваної системи.

Для визначення доцільності використання нереляційної бази даних для розробки telegram-ботів варто звернути увагу на наступні фактори:



- Обсяг та структура даних. Якщо чат-бот обробляє великі обсяги даних, або якщо дані мають складну структуру, то доцільно обрати нереляційну базу даних;
- Зміна структури даних. У випадку можливості зміни структури даних чат-боту, доцільним є використання нереляційної бази даних з гнучкою моделлю даних;

## **2.4 Висновки до розділу**

У другому розділі магістерської кваліфікаційної роботи було детально розглянуто мови програмування та бібліотеки для розробки telegram-ботів, зокрема для системи нотифікації системних помилок обмінника валют. Після проведення порівняльного аналізу, вибір був здійснений на користь мови програмування Python та бібліотеки Telebot.

Окрема увага була приділена вибору системи управління базами даних. Після ретельного аналізу властивостей реляційних і нереляційних баз даних, а також їхніх СУБД, вирішено використати реляційну базу даних MySQL.

Даний вибір технологій дозволяє забезпечити ефективну розробку та надійну роботу telegram-бота для нотифікації системних помилок обмінника валют Exchanger.

## 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Обґрунтування та проектування функціоналу telegram-боту

Проект telegram-боту для нотифікації системних помилок обмінника валют Exchanger спрямований на створення ефективного засобу миттєвого інформування адміністраторів та технічних спеціалістів про можливі аномалії та помилки в роботі системи. Розробка цього боту передбачає реалізацію широкого спектру функцій, спрямованих на надання оперативної та точної інформації про стан обмінника в реальному часі. Наявність такого інструменту сприятиме зниженню часу реакції на можливі проблеми, що, в свою чергу, позитивно позначиться на надійності та продуктивності Exchanger.

Для розробки проекту необхідно визначити функціональні вимоги telegram-боту для нотифікації системних помилок обмінника валют. Дані вимоги наведені у таблиці 3.1.

Таблиця 3.1 – Функціональні вимоги розроблюваного чат-боту

№	Вимоги
1	Під час виникнення помилки відповідне сповіщення надсилається у telegram-бот.
2	Після отримання сповіщення про помилку з'являється можливість створення тикету зі статусом To Do для технічної підтримки.
3	Після переведення тикету у статус Done у адміністративній панелі зовнішньої системи технічної підтримки сповіщення про закриття тикета надходить у telegram-бот.
4	Забезпечення зв'язку з обмінником валют Exchanger
5	Забезпечення зв'язку з зовнішньою системою технічної підтримки
6	Відловлення усіх помилок системи обмінника крипто та фіатних валют Exchanger.

### 3.2 Проектування бази даних

Важливим етапом розробки програмного забезпечення є проектування структури бази даних. Проектування бази даних – це процес створення структури та організації бази даних для забезпечення ефективного зберігання, обробки і взаємодії з даними. Цей процес включає в себе визначення потреб користувачів, аналіз вимог до інформації, створення моделі даних, нормалізацію, вибір відповідних типів даних та встановлення взаємозв'язків між різними таблицями чи сутностями.

Для проектування бази даних перш за все визначимо сутності та їхні атрибути. У моделі даних термін «сутність» вказує на об'єкт або концепцію, яку можна ідентифікувати і визначити в системі. В моделюванні даних, сутності є основною будівельною одиницею, яку використовують для опису даних та їх взаємодії.

Провівши аналіз вимог та потреб користувачів було визначено наступні сутності розроблюваного чат-боту:

1. Користувачі (users) – для збереження інформації про користувачів, які взаємодіють з ботом;
2. Валютні обміни (exchanges) – для відстеження інформації про валютні обміни;
3. Картка (ticket) – для відстеження інформації про тікети в системі технічної підтримки;
4. Адміністратор (manager) – для обробки помилок та створення тікетів;
5. Помилка (error) – для зберігання інформації про помилку у разі її виникнення.

Атрибути є конкретними характеристиками або властивостями сутностей, які визначаються для зберігання в інформаційній системі. Розглянемо атрибути кожної визначеної сутності бази даних.

Атрибутами користувача є його унікальний ідентифікатор, країна розташування, email, нікнейм та час реєстрації.

Атрибутами сутності валютних обмінів є наступні:

- унікальний ідентифікатор обміну;
- сума обміну;
- валюта для обміну;
- валюта, на яку здійснюється обмін;
- мітка часу, коли відбувся обмін;

Атрибутами тикету є унікальний ідентифікатор тикету, статус, опис, мітка часу створення тикету.

Атрибутами адміністратора є його унікальний ідентифікатор, ім'я, прізвище, email.

Атрибутами помилки є вид валюної операції, код помилки, опис, час виникнення.

Наступним кроком визначимо зв'язок між сутностями бази даних, що проектується. В базах даних, зв'язки визначають взаємодію між таблицями, що є важливим аспектом для правильно організованого зберігання та отримання даних.

Існує три види зв'язків сутностей у реляційних базах даних: «один до одного», «один до багатьох» та «багато до багатьох».

Вид зв'язку «один до одного» у моделі бази даних вказує на те, що кожен елемент одного набору даних пов'язаний лише з одним елементом іншого набору даних і навпаки. Це взаємозв'язок, де кожен запис у першій таблиці асоціюється з одним і тільки одним записом у другій таблиці, і навпаки.

Вид зв'язку «один до багатьох» в моделі бази даних вказує на те, що кожен запис у першій таблиці пов'язаний з одним або більше записами у другій таблиці, але кожен запис у другій таблиці пов'язаний лише з одним записом у першій таблиці.

Зв'язок «багато до багатьох» в моделі бази даних вказує на те, що кожен запис у першій таблиці може бути пов'язаний з декількома записами у другій

таблиці, і навпаки. Такий вид зв'язку забезпечується спеціальною проміжною таблицею, яка дозволяє встановлювати зв'язки між двома сутностями.

Визначимо тип зв'язку між сутностями «Користувач» та «Валютний обмін». Очевидним є те, що видом зв'язку між даними сутностями є «один до багатьох», так як один користувач може проводити багато валютних обмінів. На рисунку 3.1 представлено ER-діаграму сутностей «Користувач – Валютний обмін».

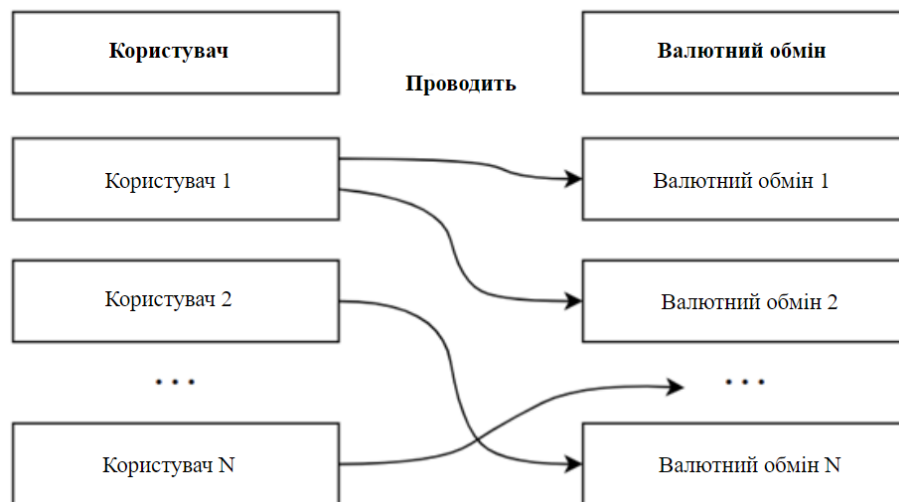


Рисунок 3.1 – ER-діаграма сутностей «Користувач – Валютний обмін»

Визначимо вид зв'язку для сутностей «Валютний обмін» та «Помилка». Так як одна операція обміну може мати одну помилку, то видом зв'язку є «один до одного». На рисунку 3.2 представлено ER-діаграму сутностей «Валютний обмін – Помилка».

Один адміністратор може обробляти багато помилок, тому видом зв'язку для сутностей «Адміністратор» та «Помилка» є «один до багатьох». На рисунку 3.3 представлено ER-діаграму сутностей «Адміністратор – Помилка».

Один адміністратор може заводити багато тікетів, тому видом зв'язку між сутностями «Адміністратор» та «Тікет» є «один до багатьох». На рисунку 3.4 представлено ER-діаграму сутностей «Адміністратор – Тікет».

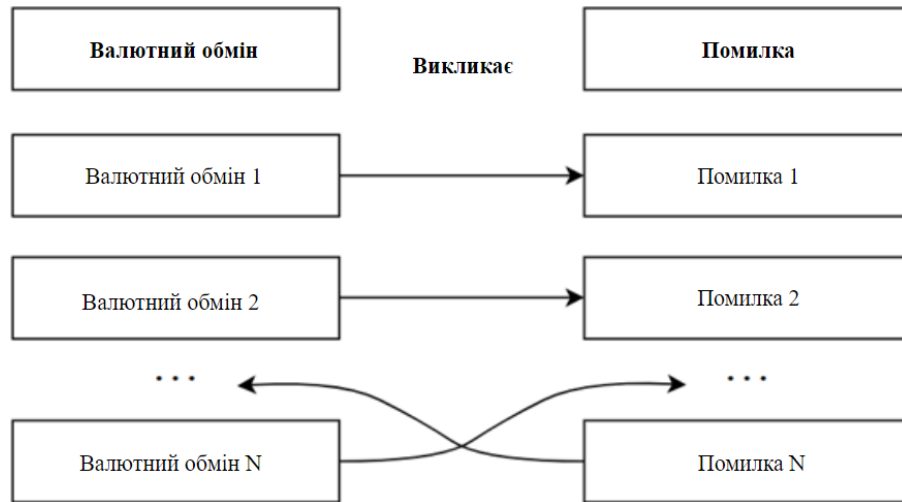


Рисунок 3.2 – ER-діаграма сутностей «Валютний обмін – Помилка»

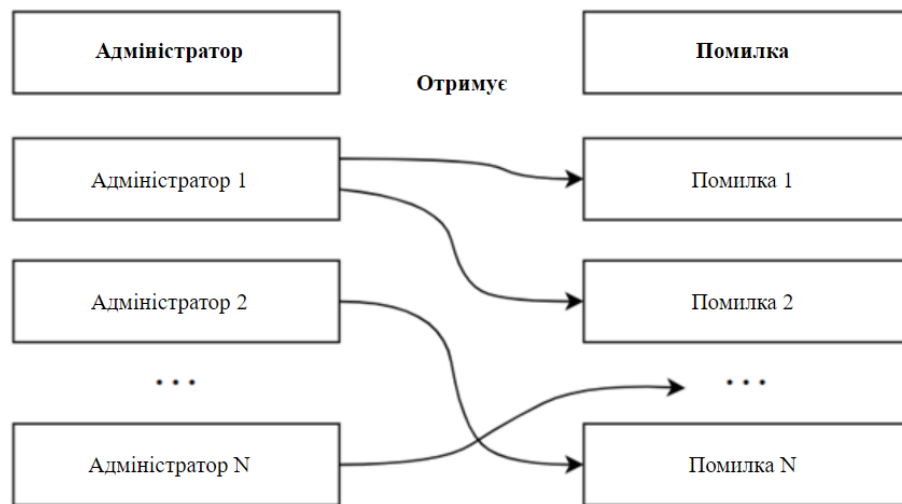


Рисунок 3.3 – ER-діаграма сутностей «Адміністратор – Помилка»

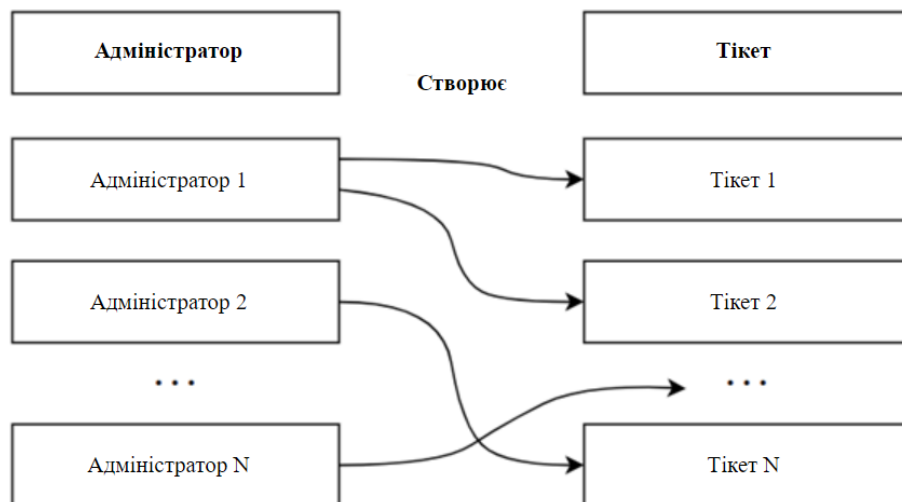


Рисунок 3.4 – ER-діаграма сутностей «Адміністратор – Тікет»

На основі створених ER-діаграм було розроблено одну результуючу, представлену на рисунку 3.5.



Рисунок 3.4 – Результуюча ER-діаграма сутностей чат-боту

На основі результуючої ER-діаграми та визначених атрибутів було спроектовано модель бази даних. На рисунку 3.5 представлено спроектовану базу даних для telegram-боту для нотифікації системних помилок обмінника валют Exchanger.

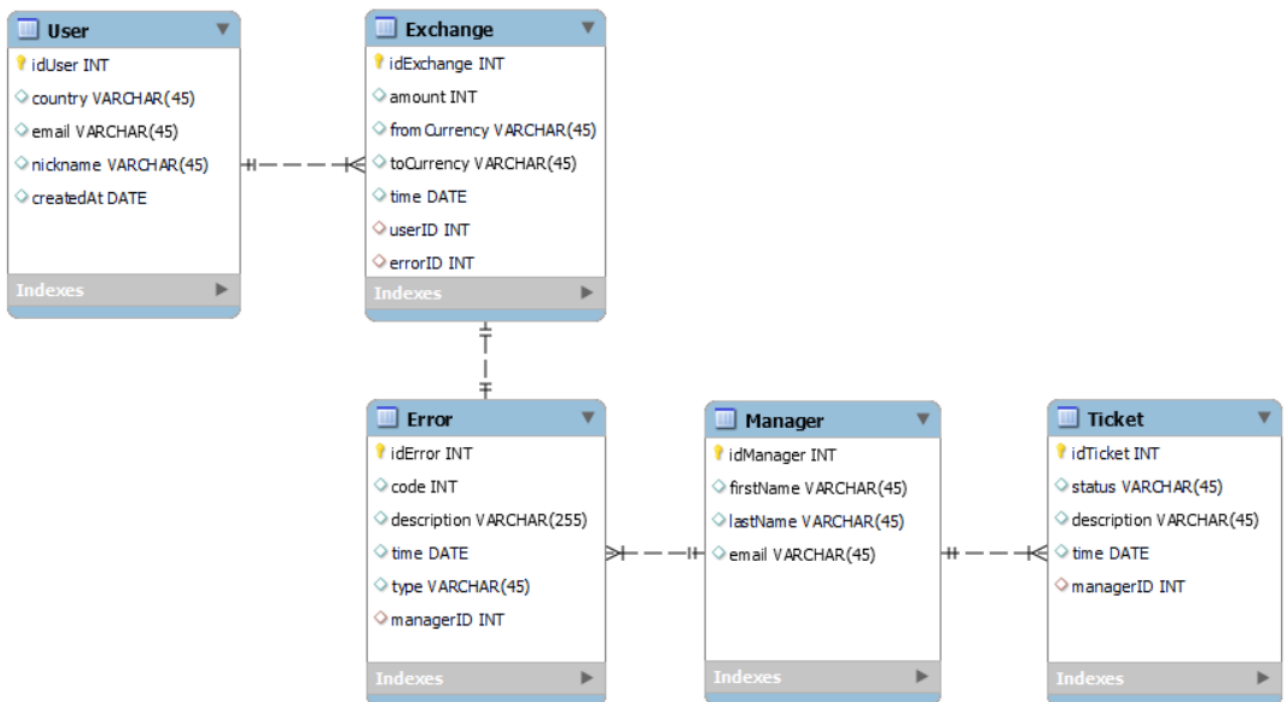


Рисунок 3.5 – Модель бази даних чат-боту

У таблиці Exchange, поля userID та errorID є зовнішніми ключами для створення зв'язку з таблицями User та Error. У таблиці Manager поле errorID є зовнішнім ключем для зв'язку з таблицею Error, а у таблиці Ticket зовнішнім ключем для зв'язку з таблицею Manager є поле managerID.

### 3.3 Опис структури проекту та основних функцій

У даному підрозділі описано структуру проекту та основні принципи взаємодії його структурних елементів. На рисунку 3.6 представлена структура telegram-боту для нотифікації системних помилок обмінника валют Exchanger.

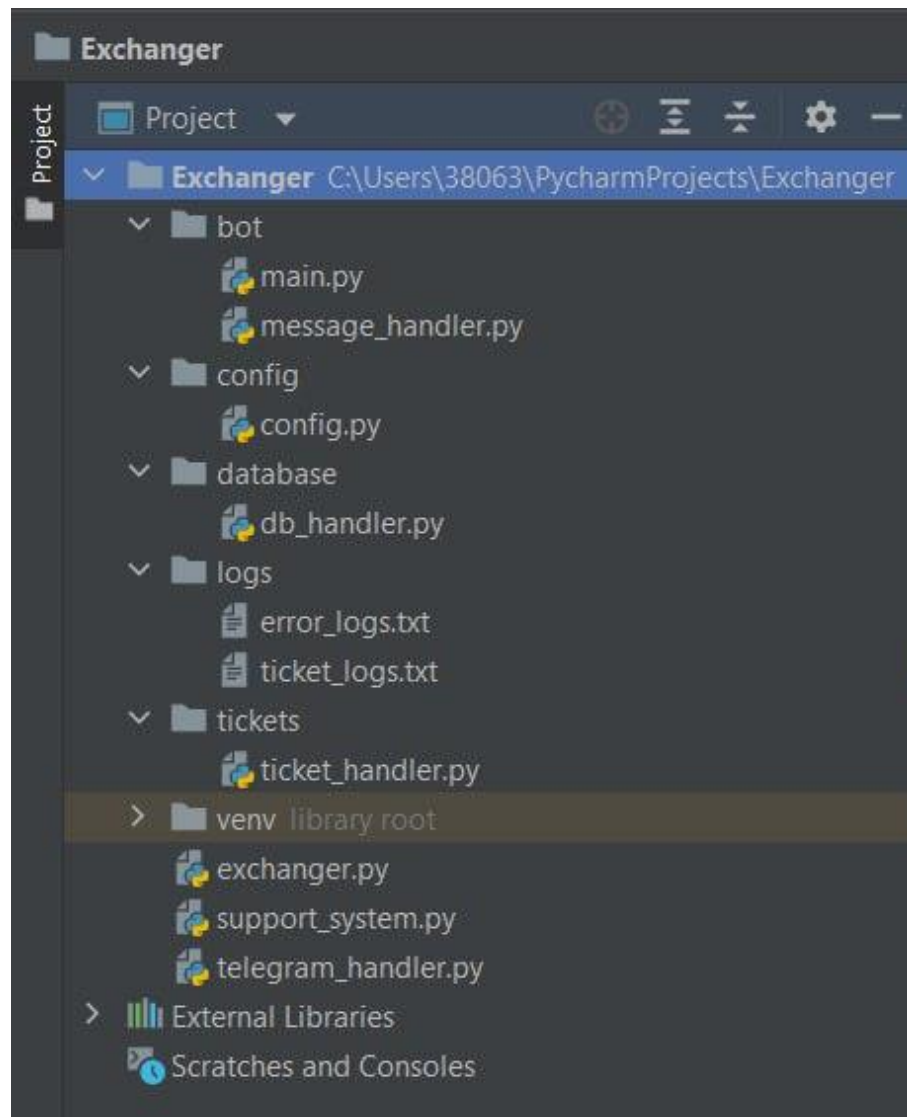


Рисунок 3.6 – Структура проекту



Розглянемо призначення та взаємодію компонентів системи. У корені папки проекту містяться директорії `bot`, `database`, `tickets`, `config` та `logs`.

Директорія `bot` містить файли, які безпосередньо пов'язані з логікою бота. Основним файлом для запуску бота є `main.py`. За логіку обробки повідомлень відповідає файл `message_handler.py`.

Директорія `database` призначена для роботи з базою даних. У файлі `database.py` встановлюється з'єднання з базою даних `mysql` де зберігається інформація про помилки та статуси тікетів.

У директорії `tickets` міститься файл `tickets_handler.py`, який призначений для обробки тікетів та зміни їх статусів, а саме створення та закриття тікетів.

Для збереження конфігураційних параметрів створено файл `config.py` у директорії `config`.

Директорія `logs` призначена для збереження логів, помилок та тікетів у файлах `error_logs.txt` та `ticket.logs.txt` відповідно.

У корені проекту містяться модулі `exchanger.py`, `telegram_handler.py` та `support_system.py`. Модуль `exchanger.py` викликає функції обробки помилок обмінника валют `Exchanger`. Модуль `telegram_handler.py` містить функції для реакції на повідомлення про отримання помилки. Модуль `support_system.py` слугує для забезпечення зв'язку із зовнішньою системою технічної підтримки та містить логіку створення тікетів.

Для встановлення з'єднання з базою даних `MySQL` було завантажено та використано бібліотеку `mysql-connector-python`. На рисунку 3.7 представлено функцію `create_db_connection` для встановлення з'єднання із базою даних.

Для виведення помилки та відправки сповіщення використано вбудовану функцію бібліотеки `Telebot` `send_message`. Приклад функції, призначеної для виведення отриманої помилки під час проведення валютної операції наведено на рисунку 3.8. Дана функція використовує функцію для створення тікету `create_ticket` при натисканні на відповідну кнопку та його відправку до зовнішньої системи підтримки, код функції `create_ticket` наведено на рисунку 3.9 та 3.10.

```

2   import mysql.connector
3   from mysql.connector import Error
4
5   def create_db_connection():
6       try:
7           connection = mysql.connector.connect(
8               host='localhost',
9               database='exchanger_stg',
10              user='test_user',
11              password='password12345'
12          )
13          if connection.is_connected():
14              print(f"Connected to MySQL database: {connection.database}")
15              return connection
16          except Error as e:
17              print(f"Error: {e}")
18              return None

```

Рисунок 3.7 – Функція підключення до бази даних

```

def handle_external_service_error(user_id, status_code, error_message):

    keyboard = types.ReplyKeyboardMarkup(row_width=1, resize_keyboard=True)
    create_ticket_button = types.KeyboardButton("Create ticket")
    keyboard.add(create_ticket_button)

    bot.send_message(
        user_id,
        f"External error.\nStatus code: {status_code}\nMessage: {error_message}",
        reply_markup=keyboard
    )

```

Рисунок 3.8 – Функція для відправки сповіщення про отриману помилку

```

support_service = 'https://adminpanel.stg/create_ticket'

def create_ticket(user_id, order_type, status_code, message, time):
    try:
        data = {
            'user_id': user_id,
            'order_type': order_type,
            'status_code': status_code,
            'message': message,
            'time': time
        }
    }

```

Рисунок 3.9 – Функція створення та відправки тикету. Частина 1

```

response = requests.post(support_servise, json=data)
response.raise_for_status()
bot.send_message(user_id, "Successful create ticket!")

except requests.exceptions.RequestException as e:
    bot.send_message(user_id, f"Can't create a ticket. Error: {str(e)}")

@bot.message_handler(func=lambda message: True)
def handle_create_ticket_button(message):
    if message.text == "Create ticket":
        user_id = message.chat.id
        order_type = "YourOrderType"
        status_code = "YourStatusCode"
        message_text = "YourMessageText"
        current_time = "YourCurrentTime"

        create_ticket(user_id, order_type, status_code, message_text, current_time)

```

Рисунок 3.9 – Функція створення та відправки тикету. Частина 2

### 3.4 Висновки до розділу

У третьому розділі дипломної розглянуто функціональні вимоги до розроблюваного telegram-бота, який призначений для нотифікації про системні помилки в обміннику валют Exchanger.

Основна увага була приділена опису процесу та етапів проектування бази даних для забезпечення ефективного функціонування чат-бота. Крім того, у даній частині роботи детально розглянута структура проекту, з описом основних модулів та компонентів. Розділ описує основні функції бота, які спрощують взаємодію з користувачем та забезпечують ефективне виконання його завдань у контексті нотифікації системних помилок під час проведення валютних операцій обмінника валют. Діаграми та блоксхеми розробленого telegram-боту наведені у додатку Б – Графічна частина.

Розроблений програмний продукт має можливості вдосконалення та перспективи подальшого розвитку, зокрема, інтеграції з іншими системами та можливості розширення функціональності для покращення користувацького досвіду та ефективності використання бота у реальних умовах.

## 4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування програмного забезпечення – це процес перевірки та валідації програмного продукту з метою визначення, чи відповідає він вимогам, функціонує коректно та задовольняє потреби користувачів.

Існують різні типи тестування програмного забезпечення, поєднання яких дозволяє досягти максимальної якості вихідного програмного продукту. Розглянемо деякі типи тестування програмного забезпечення та їхні особливості.

Функціональне тестування – тестування спрямоване на перевірку функціональності програми. Основною метою даного типу тестування є визначення відповідності вимогам системи та роботи основних функцій.

Нефункціональне тестування – це тип тестування програмного забезпечення, спрямований на оцінку характеристик, які не відносяться безпосередньо до функціональності системи. Відмінною особливістю нефункціонального тестування є перевірка продуктивності, безпеки, масштабованості та інші.

Тестування безпеки є процесом, спрямованим на ідентифікацію потенційних вразливостей в програмному забезпеченні або системі з метою захисту їх від небажаних атак, витоку чи несанкціонованого доступу. Основна мета тестування безпеки полягає в забезпеченні та підтримці відповідного рівня безпеки системи.

Позитивне тестування є методом тестування, при якому тестові випробування спрямовані на перевірку правильності роботи програмного забезпечення в умовах очікуваних та правильних вхідних даних та умов.

Негативне тестування – це метод тестування програмного забезпечення, спрямований на перевірку того, як система чи програма реагує на невірні, неправильні або неприпустимі вхідні дані. Основною метою негативного тестування є виявлення вразливостей, відсутності валідації та обробки помилок, а також визначення, як система поводить себе в умовах некоректних дій користувачів або непередбачуваних ситуацій.

Для функціонального тестування telegram-боту для нотифікації системних помилок було розроблено ряд тест-кейсів, представлених у таблиці 4.1.

Таблиця 4.1 – Тест-кейси функціонального тестування telegram-боту

№	Тест-кейс	Результат
1	Помилка валідації при вимкненій валюті поповнення	+
	<i>Передумови</i> користувач успішно зареєстрований та авторизований є доступ до бази даних (таблиця currency)	
	<i>Кроки</i> 1. Перейти в таблицю currency та вимкнути валюту поповнення, наприклад TRX 2. Відправити запит на створення поповнення	
	<i>Очікуваний результат</i> Сповіщення про наступну помилку у боті 422 Unprocessable Entity { "error": "VALIDATION_ERROR", "message": "Invalid value" }	
2	Помилка валідації при вимкненій валюті виплати	+
	<i>Передумови</i> користувач успішно зареєстрований та авторизований є доступ до бази даних (таблиця currency)	
	<i>Кроки</i> 1. Перейти в таблицю currency та вимкнути валюту виплати, наприклад TRX 2. Відправити запит на створення виплати { "external_id": "{{\${randomUUID}}}", "to_payway": "tron", "from_currency": "TRX", "to_currency": "TRX", "to_amount": "1", "requisites": { "token": "b0be02e0-db20-518e-9df9-78628748120b" } }	
	<i>Очікуваний результат</i> Сповіщення про наступну помилку у боті 422 Unprocessable Entity { "error": "VALIDATION_ERROR", "message": "Invalid value" }	

Продовження таблиці 4.1

№	Тест-кейс	Результат
3	<p>Помилка валідації при вимкненій валюті конвертації (валюта продажу)</p> <p><i>Передумови</i> користувач успішно зареєстрований та авторизований є доступ до бази даних (таблиця currency)</p> <p><i>Кроки</i> 1. Перейти в таблицю currency та вимкнути валюту конвертації (валюта продажу), наприклад TRX 2. Відправити запит на створення конвертації {   "from_currency": "TRX",   "to_currency": "UAH",   "from_amount": "10",   "external_id": "{\$randomUUID}" }</p> <p><i>Очікуваний результат</i> Сповіщення про наступну помилку у боті 404 Not Found {   "error": "NOT_FOUND",   "message": "Cannot find the resource or item: Currency" }</p>	+
4	<p>Помилка валідації при вимкненій валюті конвертації (валюта купівлі)</p> <p><i>Передумови</i> користувач успішно зареєстрований та авторизований є доступ до бази даних (таблиця currency)</p> <p><i>Кроки</i> 1. Перейти в таблицю currency та вимкнути валюту конвертації (валюта купівлі), наприклад UAH 2. Відправити запит на створення конвертації {   "from_currency": "TRX",   "to_currency": "UAH",   "from_amount": "10",   "external_id": "{\$randomUUID}" }</p> <p><i>Очікуваний результат</i> Сповіщення про наступну помилку у боті 404 Not Found {   "error": "NOT_FOUND",   "message": "Cannot find the resource or item: Currency" }</p>	+

Продовження таблиці 4.1

№	Тест-кейс	Результат
5	<p>Помилка валідації при вимкненій платіжному направленні поповнення</p> <p><i>Передумови</i> користувач успішно зареєстрований та авторизований є доступ до бази даних (таблиця payway)</p> <p><i>Кроки</i> 1. Перейти в таблицю payway та вимкнути платіжне направлення, (наприклад tron) 2. Відправити запит на створення поповнення {   "external_id": "{{ \$randomUUID }}",   "from_payway": "tron",   "from_currency": "TRX",   "to_currency": "TRX",   "from_amount": "100" }</p> <p><i>Очікуваний результат</i> Сповіднення про наступну помилку у боті 404 Not Found {   "error": "NOT_FOUND",   "message": "Cannot find the resource or item: Payway" }</p>	+
6	<p>Помилка валідації при вимкненій платіжному направленні виплати</p> <p><i>Передумови</i> користувач успішно зареєстрований та авторизований є доступ до бази даних (таблиця payway)</p> <p><i>Кроки</i> 1. Перейти в таблицю payway та вимкнути платіжне направлення, (наприклад tron) 2. Відправити запит на створення виплати {   "external_id": "{{ \$randomUUID }}",   "to_payway": "tron",   "from_currency": "TRX",   "to_currency": "TRX",   "to_amount": "1",   "requisites": { "token": "b0be02e0-db20-518e-9df9-78628748120b" } }</p> <p><i>Очікуваний результат</i> Сповіднення про наступну помилку у боті 404 Not Found { "error": "NOT_FOUND",   "message": "Cannot find the resource or item: Payway" }</p>	+

Продовження таблиці 4.1

№	Тест-кейс	Результат
7	<p>Помилка валідації при вимкненій валютній парі транзакція - конвертація</p> <p><i>Передумови</i> користувач успішно зареєстрований та авторизований є доступ до бази даних (таблиця convert_direction)</p> <p><i>Кроки</i> 1. Перейти в таблицю рауway та вимкнути валютну пару, (наприклад TRX -&gt; UAH) 2. Відправити запит на створення конвертації {   "from_currency": "TRX",   "to_currency": "UAH",   "from_amount": "10",   "external_id": "{{ \$randomUUID }}" }</p> <p><i>Очікуваний результат</i> Сповіднення про наступну помилку у боті 404 Not Found {   "error": "NOT_FOUND",   "message": "Cannot find the resource or item: Convert Direction" }</p>	+
8	<p>Перевірити створення тикету після отримання повідомлення про помилку в телеграм боті</p> <p><i>Передумови</i> користувач успішно зареєстрований та авторизований є доступ до бази даних(таблиця convert_direction) вимкнена валютна пара, (наприклад TRX -&gt; UAH) відправлений запит на створення конвертації {   "from_currency": "TRX",   "to_currency": "UAH",   "from_amount": "10",   "external_id": "{{ \$randomUUID }}" }</p> <p><i>Кроки</i> 1. Перейти в телеграм бот 2. Перевірити відображення повідомлення про помилку в транзакції 3. Натискаємо на кнопку "Create ticket" віправляється запит на створення тикету з даними помилки, з такими полями як user_id, order_type, status_code, message, time</p> <p><i>Очікуваний результат</i> Успішно створений тикет у системі адмінстрування запитів зі статусом To Do</p>	+



Продовження таблиці 4.1

№	Тест-кейс	Результат
9	Перевірити закриття тикету після отримання повідомлення про вирішення помилки в телеграм боті	+
	<i>Передумови</i> користувач успішно зареєстрований та авторизований є доступ до бази даних(таблиця convert_direction) вимкнена валютна пара, (наприклад TRX -> UAH) відправлений запит на створення конвертації { "from_currency": "TRX", "to_currency": "UAH", "from_amount": "10", "external_id": "{{ \$randomUUID }}" }	
	є створений тикет по данній помилці перевести його в статус DONE	
	<i>Кроки</i> Перейти в телеграм бот 2. Перевірити відображення повідомлення про вирішену помилку 3. Натискаємо на кнопку "Close ticket" віправляється запит на закриття тикету	
<i>Очікуваний результат</i> Тикет успішно закриття, сповіщення про закриття тикета прийшло у telegram-бот		

Отримані результати тестування підтверджують, що створений telegram-бот, який призначений для нотифікації про системні помилки в обміннику валют Exchanger, повністю відповідає поставленим вимогам.

Протягом тестування було ретельно перевірено різноманітні сценарії використання, включаючи реакцію на системні помилки, введення користувача та обробку запитів. Система продемонструвала високу ступінь коректності та точності у своїй роботі, забезпечуючи вчасну інформацію щодо будь-яких аномалій чи проблем в роботі обмінника валют.

Розглянемо приклад виконання тест-кейсу №3 – Помилка валідації при вимкненій валюті конвертації (валюта продажу). Для цього перейдемо на сторінку обмінника валют Exchanger та спробуємо провести валютну операцію з продажем валюти TRX, яка вимкнена для країни розташування користувача.

Даний крок представлено на рисунку 4.1.

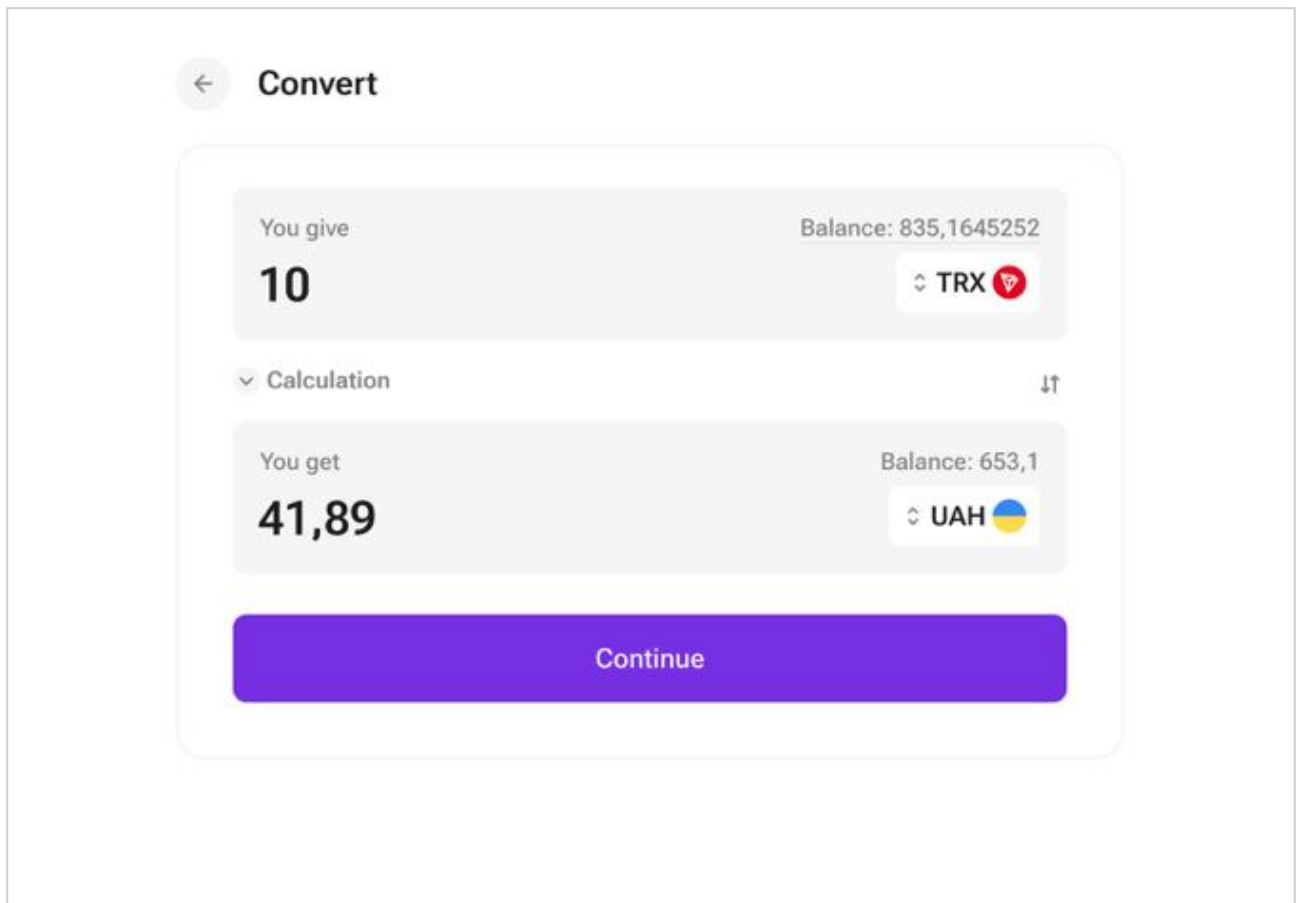


Рисунок 4.1 – Проведення валютної операції для валюти вимкненої поточній країні користувача

Після натискання на кнопку Continue у розроблений telegram-бот успішно прийшло сповіщення про помилку під час проведення валютної операції. На рисунку 4.2 представлено отримане сповіщення з можливістю відкрити тикет для помилки.

Після натискання та кнопку «Create ticket» у адміністративній панелі технічної підтримки успішно створено тикет зі статусом To Do. Приклад створеного тикета представлено на рисунку 4.3.

Після переведення тикету в Done у telegram-бот приходить сповіщення про вирішення проблеми і його успішне закриття. Приклад сповіщення про закриття тикету представлено на рисунку 4.4.

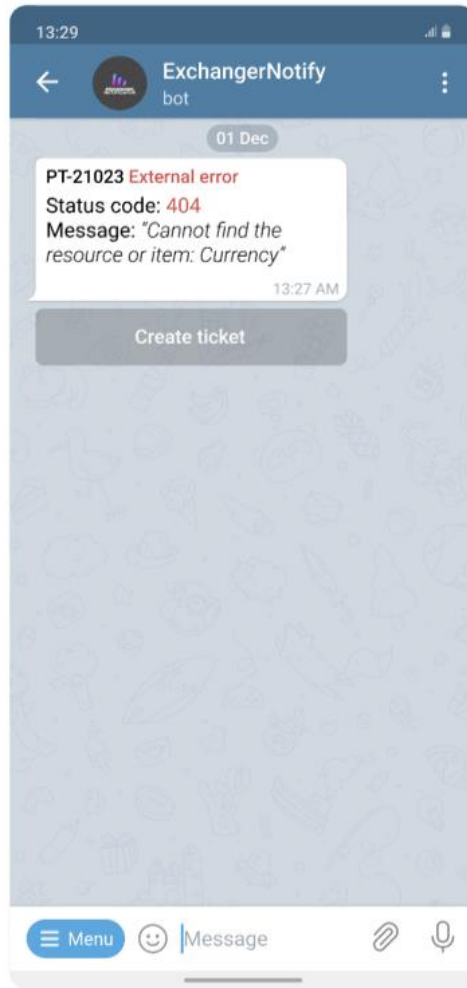


Рисунок 4.2 – Сповіднення у telegram-бот про отриману системну помилку

Tickets							AS Admin admin@anyup.co
Number	Status	Description	Date	Actions			
OP-01123	TO DO	VALIDATION_ERROR	24 Nov, 2023			Close	
MIN-85467	DONE	Cannot Find The Resource Or Item: Payway	25 Nov, 2023			Close	
RE-01231	DONE	Server Is Unable To Process The Request At This Moment	25 Nov, 2023			Close	
PY-31698	IN PROGRESS	The Requested Currency Pair Is Not Supported By The Server. Please Choose A Different Currency Pair And Try Again.	25 Nov, 2023			Close	
VR-5110	IN PROGRESS	The Format Of The Currency Exchange Request Is Invalid. Please Ensure Your Request Follows The Specified Format ...	27 Nov, 2023			Close	
HJ-10210	IN PROGRESS	Exchange Rate Data For The Specified Currencies Is Unavailable At The Moment. Please Try Again Later Or Choose A ...	29 Nov, 2023			Close	
PT-21023	TO DO	Cannot Find The Resource Or Item: Currency	01 Dec, 2023			Close	
SD-23014	DONE	Server Is Unable To Process The Request At This Moment	01 Dec, 2023			Close	
CO-45623	DONE	An Error Occurred While Converting The Currency. Please Double-Check The Provided Values And Try Again.	01 Dec, 2023			Close	

Рисунок 4.3 – Приклад створеного тикету

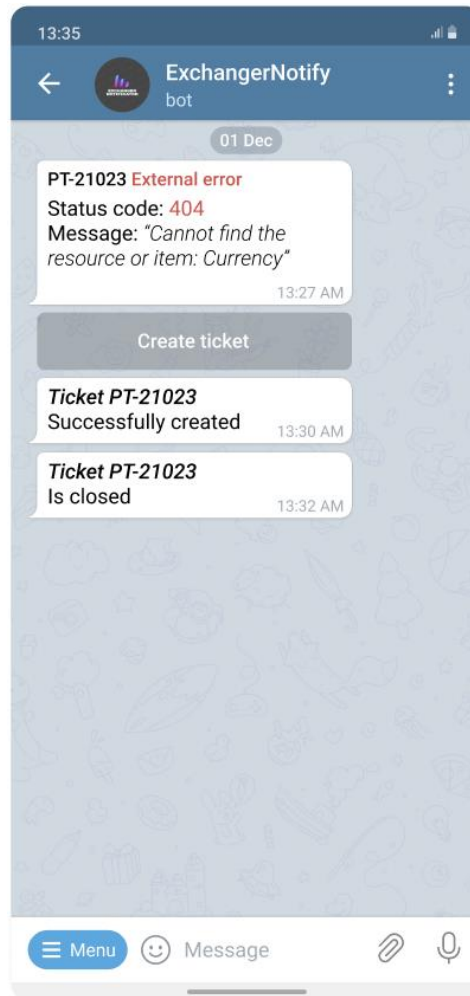


Рисунок 4.4 – Сповіщення про закриття тікета

#### 4.1 Висновки до розділу

У даному розділі було розглянуто сутність тестування та його основні типи, розроблено тест-кейси та проведено функціональне тестування telegram-боту для нотифікації системних помилок обмінника валют Exchanger.

Результати тестування підтверджують, що розроблений telegram-бот повністю відповідає поставленим вимогам. За допомогою розробленого програмного продукту відповідальний працівник технічної підтримки миттєво отримує сповіщення про системну помилку, у разі прийняття рішення про відкриття тікета для команди розробників та тестувальників працівник технічної підтримки може його відкрити безпосередньо з чат-боту. Після закриття тікету даний працівник отримує відповідне сповіщення.

## 5 ЕКОНОМІЧНИЙ РОЗДІЛ

### 5.1 Технологічний аудит розробленого боту для нотифікації системних помилок обмінника валют «Exchanger»

Як було зазначено у попередніх розділах нашої роботи, «Exchanger» – це цифровий гаманець і онлайн-сервіс швидкого включення та відключення, який дає змогу будь-якому окремому клієнту продавати свою криптовалюту за фіксовану іншу валюту та надсилати її на будь-який банківський рахунок чи платіжну картку або купувати та зберігати криптовалюту чи надсилати її на будь-який інший криптосервіс.

Здійснення такої обмінної операції має бути гарантоване забезпеченням для клієнта достатнього рівня фінансової безпеки та зменшенням ризику можливих втрат, пов'язаних зі зміною курсів валют та іншими випадковими несприятливими подіями.

Для часткового розв'язання цієї проблеми нами було розроблено бот для нотифікації системних помилок обмінника валют «Exchanger» для корпоративних месенджерів, використання якого буде автоматично відстежувати системні помилки та збої в роботі системи та відправляти оперативні повідомлення та звіти розробникам системи та службі підтримки.

Для досягнення поставленої мети під час виконання МКР було: проведено аналіз існуючих систем аналогів; зроблено ретельний огляд інструментарію для реалізації боту для нотифікації системних помилок обмінника валют «Exchanger»; розроблено програмне забезпечення системи нотифікації про системні помилки та збої за допомогою боту; проведено тестування розробленого продукту.

В результаті було розроблено програмне забезпечення, використання якого що дозволить підвищити контроль якості фінансових операцій та повністю автоматизувати процес обміну крипто та фіатних валют.

Для встановлення комерційного потенціалу розробленого нами боту для нотифікації системних помилок обмінника валют «Exchanger» проведено його технологічний аудит, для чого запросимо 3-х відомих експертів – кандидатів технічних наук, доцентів Гармаша В.В., Богач І.В. та д.т.н. професора Бісікала О.В.

Встановлення комерційного потенціалу розробленого нами боту для нотифікації системних помилок обмінника валют «Exchanger» було здійснено за критеріями, наведеними в таблиці 5.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу будь-якої розробки і їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
<b>Ринкові перспективи</b>					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
<b>Практична здійсненність</b>					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у пром. комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали відомі та використовуються у виробництві

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Запрошені експерти оцінили розроблений нами бот для нотифікації системних помилок обмінника валют «Exchanger» таким чином (табл. 5.2):

Таблиця 5.2 - Результати технологічного аудиту розробленого нами бота (за шкалою оцінювання «0»-«1»-«2»-«3»-«4»)

Критерії	Прізвище, ініціали експертів		
	Гармаш В.В.	Богач І.В.	Басікало О.В.
	Бали, що їх виставили експерти:		
1	4	4	3
2	4	3	4
3	4	3	4
4	4	4	4
5	3	4	3



Продовження таблиці 5.2

6	4	4	4
7	3	4	4
8	4	3	3
9	4	4	4
10	4	4	4
11	4	4	4
12	4	3	4
Сума балів	$СБ_1 = 46$	$СБ_2 = 44$	$СБ_3 = 45$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{46 + 44 + 45}{3} = \frac{135}{3} = 45,00$		

Встановлення комерційного потенціалу розробленого нами боту будемо здійснювати на основі рекомендацій, наведених в таблиці 5.3 [3].

Таблиця 5.3 - Рівні комерційного потенціалу будь-якої наукової розробки

Середньоарифметична сума балів $\overline{СБ}$ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Оскільки середньоарифметична сума балів, що їх виставили експерти, складає 45 балів, то це свідчить, що розроблений нами бот для нотифікації системних помилок обмінника валют «Exchanger» має рівень комерційного потенціалу, який вважається «високим».

Це пояснюється тим, що нами розроблено та імплементовано алгоритми виявлення специфічних системних помилок для обмінника валют «Exchanger», а також розроблені нами алгоритми можна доопрацювати під потреби іншого проекту без необхідності підлаштовування під обмежений набір мов розробки.

## 5.2 Розрахунок витрат на розроблення боту для нотифікації системних помилок обмінника валют «Exchanger»

На розроблення telegram-боту були зроблені певні витрати.

А). Основна заробітна плата  $Z_o$  розробників, яка визначається за формулою:

$$Z_o = \frac{M}{T_p} \cdot t \text{ грн}, \quad (5.1)$$

де  $M$  – місячний посадовий оклад розробника, грн; прийmemo, що

$M = (6700 \dots 20000)$  грн/місяць;

$T_p$  – число робочих днів в місяці; прийmemo  $T_p = 22$  дні;

$t$  – число днів роботи розробників.

Зроблені розрахунки зведемо до таблиці 5.4:

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник магістерської роботи	18600	845,45	20 годин	$\approx 2818$
2. Магістрант-студент-виконавець	2000 (беремо 6700)	304,54	75	$\approx 22841$
3. Консультант з економічної частини	17000	772,72	1,5 години	$\approx 193$ (при 6-годинному робочому дні)
Загалом				$Z_o = 25852$ грн

Б). Додаткова заробітна плата  $Z_d$  розробників розраховується як (10...12)% від величини їх основної заробітної плати, тобто:

$$Z_d = \alpha \cdot Z_o = (0,1 \dots 0,12) \cdot Z_o. \quad (5.2)$$

Прийmemo, що  $\alpha = 0,1$ . Тоді для нашого випадку отримаємо:

$$Z_d = 0,1 \times 25852 = 2585,2 \approx 2586 \text{ грн.}$$

В). Нарахування на заробітну плату НЗП<sub>зп</sub> розробників (дослідників) розраховуються за формулою:

$$\text{НЗП}_{\text{зп}} = (З_о + З_д) \cdot \frac{\beta}{100}, \quad (5.3)$$

де  $\beta$  – ставка обов'язкового єдиного внеску на державне соціальне страхування, %.  $\beta = 22\%$ . Тоді:

$$\text{НЗН}_{\text{зп}} = (25852 + 2586) \times 0,22 = 6256,36 \approx 6257 \text{ грн.}$$

Г). Амортизація основних засобів А, які використовувались під час виконання даної роботи:

$$A = \frac{Ц \cdot N_a}{100} \cdot \frac{T}{12} \text{ грн.}, \quad (5.4)$$

де Ц – загальна балансова вартість основних засобів, грн;

$N_a$  – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що  $N_a = (5...25)\%$ ;

T – термін використання основних засобів, місяці.

Зроблені розрахунки зведено в таблицю 5.5.

Таблиця 5.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
1. Комп'ютерна техніка, обладнання тощо	80000	25	3,0 (при 60% використанні)	3000
2. Приміщення університету, кафедри	40000	4	3,0 (при 15% використанні)	60
Всього				A = 3060 грн

Д). Витрати на матеріали М розраховуються за формулою:

$$M = \sum_1^n H_i \cdot Ц_i \cdot K_i - \sum_1^n B_i \cdot Ц_в \text{ грн.}, \quad (5.5)$$

де  $H_i$  – витрати матеріалу  $i$ -го найменування, кг;  $\Pi_i$  – вартість матеріалу  $i$ -го найменування;  $K_i$  – коефіцієнт транспортних витрат,  $K_i = (1,1\dots1,15)$ ;  $V_i$  – маса відходів матеріалу  $i$ -го найменування;  $\Pi_v$  – ціна відходів матеріалу  $i$ -го найменування;  $n$  – кількість видів матеріалів.

Е). Витрати на комплектуючі  $K$  розраховуються за формулою:

$$K = \sum_1^n H_i \cdot \Pi_i \cdot K_i \text{ грн}, \quad (5.6)$$

де  $H_i$  – кількість комплектуючих  $i$ -го виду, шт.;  $\Pi_i$  – ціна комплектуючих  $i$ -го виду;  $K_i$  – коефіцієнт транспортних витрат,  $K_i = (1,1\dots1,15)$ ;  $n$  – кількість видів комплектуючих.

Під час виконання роботи загальні витрати на матеріали та комплектуючі склали приблизно 5000 грн.

Ж). Витрати на силову електроенергію  $V_e$  розраховуються за формулою:

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d}, \quad (5.7)$$

де  $V$  – вартість 1 кВт-год. електроенергії, в 2023 р.  $V \approx 4,5$  грн/кВт;

$\Pi$  – установлена потужність обладнання, кВт;  $\Pi = 1,2$  кВт;

$\Phi$  – фактична кількість годин роботи обладнання, годин.

Прийmemo, що  $\Phi = 220$  годин;

$K_{\Pi}$  – коефіцієнт використання потужності;  $K_{\Pi} < 1 = 0,79$ .

$K_d$  – коефіцієнт корисної дії,  $K_d = 0,68$ .

Тоді витрати на силову електроенергію будуть дорівнювати:

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d} = \frac{4,5 \cdot 1,2 \cdot 220 \cdot 0,79}{0,68} = 1380,17 \approx 1381 \text{ грн.}$$

И). Інші витрати  $V_{\text{інш}}$  можна прийняти як (50...300)% від основної заробітної плати розробників, тобто:

$$V_{\text{інш}} = (0,5\dots3) \times 3_0. \quad (5.8)$$

Для нашого випадку отримаємо:

$$V_{\text{інш}} = 2,0 \times 25852 = 51704 \text{ грн.}$$

К). Сума всіх попередніх статей витрат складає витрати на виконання нашої роботи (безпосередньо розробником-магістрантом) – В.

$$B = 25852 + 2586 + 6257 + 3060 + 5000 + 1381 + 51704 = 95840 \text{ грн.}$$

Л). Загальні витрати на розробку бота для нотифікації системних помилок обмінника валют «Exchanger»  $V_{\text{заг}}$  розраховуються за формулою:

$$V_{\text{заг}} = \frac{B}{\beta}, \quad (5.9)$$

де  $\beta$  – коефіцієнт, який характеризує етап (стадію) виконання цієї роботи. Можна прийняти, що,  $\beta \approx 0,5$ , оскільки робота потребує подальшого доопрацювання.

$$\text{Тоді: } V_{\text{заг}} = \frac{95840}{0,5} = 191680 \text{ грн або приблизно 192 тисячі грн.}$$

Тобто прогнозовані загальні витрати на розробку бота для нотифікації системних помилок обмінника валют «Exchanger»  $V_{\text{заг}}$  становлять приблизно 192 тисяч грн.

### **5.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки**

Економічний ефект від можливої комерціалізації розробленого нами бота для нотифікації системних помилок обмінника валют «Exchanger» пояснюється її значно кращими функціональними можливостями, суттєвим зменшенням рівня ризику при виконанні обмінних операцій тощо. Тому нашу розробку можна буде реалізовувати на ринку дещо дорожче, ніж аналогічні за функціями розробки. Так, якщо річна підписка на подібні за функціями розробки у 2022 році коштувала для потенційного клієнта на ринку приблизно 4500 грн за рік, то нашу річну підписку на нашу розробку можна буде реалізовувати приблизно за 5000 грн або на 500 грн дорожче.

Аналіз місткості ринку показує, що на сьогодні в Україні кількість реальних користувачів послугами подібних ботів може складати приблизно 2000

осіб щороку. Це, насамперед, організації та інші зацікавлені особи тощо. Тому, у зв'язку зі стрімким розвитком ринку криптовалют, можна очікувати суттєве зростання попиту на нашу розробку принаймні протягом 3-х років після її впровадження.

Тобто, якщо наша розробка буде впроваджена з 1 січня 2024 року, то її результати будуть виявлятися протягом 2024-го, 2025-го та 2026-го років.

Прогноз зростання попиту на нашу розробку може становити по роках:

- а) 2024 р. – приблизно +300 шт. до базового року;
- б) 2025 р. – +600 шт. до базового року;
- в) 2026 р. – +1000 шт. до базового року.

Можливе збільшення чистого прибутку  $\Delta\Pi_i$ , що його може отримати потенційний інвестор від комерціалізації, тобто виведення нашої розробки на ринок, становитиме:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\nu}{100}\right), \quad (5.10)$$

де  $\Delta\Pi_o$  – покращення основного якісного показника від впровадження результатів розробки у цьому році. Для нашого випадку це є збільшення річної ціни підписки на нашу розробку  $\Delta\Pi_o = 5000 - 4500 = + 500$  грн;

$N$  – основний кількісний показник, який визначає обсяг діяльності у році до впровадження результатів розробки;  $N = 2000$  шт.;

$\Delta N$  – покращення основного кількісного показника від впровадження результатів нашої розробки. Таке покращення становитиме по роках, відповідно: у 2024 році – + 300 шт., у 2025 році +600 шт, та у 2026 році +1000 шт. (до базового 2023-го року);

$\Pi_o$  – основний якісний показник (тобто ціна), який визначає обсяг діяльності у році після впровадження результатів розробки, грн;  $\Pi_o = 5000$  грн;

$n$  – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; для нашого випадку  $n = 3$ ;

$\lambda$  – коефіцієнт, який враховує сплату податку на додану вартість;  
 $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати  $\rho = (0,2 \dots 0,5)$ ; візьмемо  $\rho = 0,5$ ;

$\nu$  – ставка податку на прибуток. У 2024-2026 роках  $\nu = 18\%$ .

Тоді можливе зростання чистого прибутку  $\Delta\Pi_1$  для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2024 р.) складе:

$$\Delta\Pi_1 = [500 \cdot 2000 + 5000 \cdot 300] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 854132,5 \text{ грн}$$

або приблизно 855 тисяч грн.

Можливе зростання чистого прибутку  $\Delta\Pi_2$  для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2025 р.) року складе:

$$\Delta\Pi_2 = [500 \cdot 2000 + 5000 \cdot 600] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 1366612,0 \text{ грн}$$

або приблизно 1367 тисяч грн.

Можливе зростання чистого прибутку  $\Delta\Pi_3$  для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2026 р.) року складе:

$$\Delta\Pi_3 = [500 \cdot 2000 + 5000 \cdot 1000] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 2049918 \text{ грн}$$

або приблизно 2050 тисяч грн.

Приведена вартість зростання всіх чистих прибутків від можливого впровадження нашої розробки становитиме:

$$\text{ПП} = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.11)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн;

$t$  – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку  $t = 3$  роки;

$\tau$  – ставка дисконтування. Прийmemo  $\tau = 0,10$  (10%);

$t$  – період часу від моменту початку розроблення бота до моменту отримання можливих чистих прибутків.

Тоді приведена вартість зростання всіх можливих чистих прибутків ПП, що їх може отримати потенційний інвестор від комерціалізації нашої розробки, складе:

$$\text{ПП} = \frac{855}{(1+0,1)^2} + \frac{1367}{(1+0,1)^3} + \frac{2050}{(1+0,1)^4} \approx 707 + 1027 + 1400 = 3134 \text{ тисячі грн.}$$

Теперішня вартість інвестицій  $PV$ , що повинні бути вкладені в реалізацію нашої розробки:  $PV = (1,0\dots5) \times B_{\text{заг}}$ .

Для нашого випадку  $PV = (1,0\dots5) \times 192 = 5 \times 192 = 960$  тисяч грн.

Розраховуємо абсолютний ефект від можливих вкладених інвестицій  $E_{\text{абс}}$ .

$$E_{\text{абс}} = \text{ПП} - PV, \quad (5.12)$$

де ПП – приведена вартість збільшення всіх чистих прибутків для інвестора від можливого впровадження нашої розробки, грн;

$PV$  – теперішня вартість інвестицій  $PV = 960$  тисяч грн.

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки складе:

$$E_{\text{абс}} = 3134 - 960 = 2174 \text{ тисяч грн.}$$

Оскільки  $E_{\text{абс}} > 0$ , то комерціалізація нашої розробки може бути доцільною.

Далі розрахуємо внутрішню дохідність  $E_{\text{в}}$  вкладених інвестицій:

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (5.13)$$

де  $E_{\text{абс}}$  – абсолютний ефект вкладених інвестицій;  $E_{\text{абс}} = 2174$  тис. грн;

$PV$  – теперішня вартість початкових інвестицій  $PV = 960$  тис. грн;

$T_{\text{ж}}$  – життєвий цикл розробки, роки.

$T_{\text{ж}} = 4$  років (2023-й, 2024-й, 2025-й, 2026-й роки)

Для нашого випадку отримаємо:



$$E_B = \sqrt[4]{1 + \frac{2174}{960}} - 1 = \sqrt[4]{1 + 2,2646} - 1 = \sqrt[4]{3,2646} - 1 = 1,344 - 1 = 0,344 = 34,4\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційному інвестору не вигідно буде займатися комерціалізацією нашої розробки.

Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування  $\tau_{\text{мін}}$  визначається за формулою:

$$\tau_{\text{мін}} = d + f, \quad (5.14)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d = (0,10...0,12)$ ;

$f$  – показник, що характеризує ризикованість вкладень;  $f = (0,05...0,30)$ .

Для нашого випадку отримаємо:

$$\tau_{\text{мін}} = 0,12 + 0,20 = 0,32 \text{ або } \tau_{\text{мін}} = 32\%.$$

Оскільки величина  $E_B = 34,4\% > \tau_{\text{мін}} = 32\%$ , то потенційний інвестор у принципі може бути зацікавлений у фінансуванні та комерціалізації нашої розробки.

Далі розраховуємо термін окупності коштів, вкладених у можливу комерціалізацію розробленого нами бота для нотифікації системних помилок обмінника валют «Exchanger».

Термін окупності  $T_{\text{ок}}$  розраховується за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}. \quad (5.15)$$

Для нашого випадку термін окупності  $T_{\text{ок}}$  коштів становитиме:

$$T_{\text{ок}} = \frac{1}{0,344} = 2,91 \text{ років} < 3 \text{ років},$$

що свідчить про потенційну доцільність комерціалізації розробленого нами бота для нотифікації системних помилок обмінника валют «Exchanger».

Далі проведено моделювання залежності величини внутрішньої дохідності вкладених потенційних інвестицій від рівня інфляції в країні. Як відомо, рівень інфляції в країні в 2024-му та наступних роках може зрости (наприклад, до 20%).

Прийнявши рівень інфляції у 20% отримаємо:

$$\text{ПП} = \frac{855}{(1+0,2)^2} + \frac{1367}{(1+0,2)^3} + \frac{2050}{(1+0,2)^4} \approx 594 + 791 + 989 = 2374 \text{ тисячі грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки за три роки складе:

$$E_{\text{абс}} = 2374 - 960 = 1414 \text{ тисяч грн.}$$

Внутрішня дохідність  $E_B$  вкладених інвестицій становитиме:

$$E_B = T_x \sqrt[4]{1 + \frac{E_{\text{абс}}}{PV}} - 1,$$

де  $E_{\text{абс}}$  – абсолютний ефект вкладених інвестицій;  $E_{\text{абс}} = 1414$  тисяч грн;

$PV$  –теперішня вартість початкових інвестицій  $PV = 960$  тисяч грн.

Для нашого випадку отримаємо:

$$E_B = \sqrt[4]{1 + \frac{1414}{960}} - 1 = \sqrt[4]{1 + 1,4729} - 1 = \sqrt[4]{2,4729} - 1 = 1,254 - 1 = 0,254 = 25,4\%.$$

Зроблені розрахунки у вигляді графіків наведено на рис. 5.1.

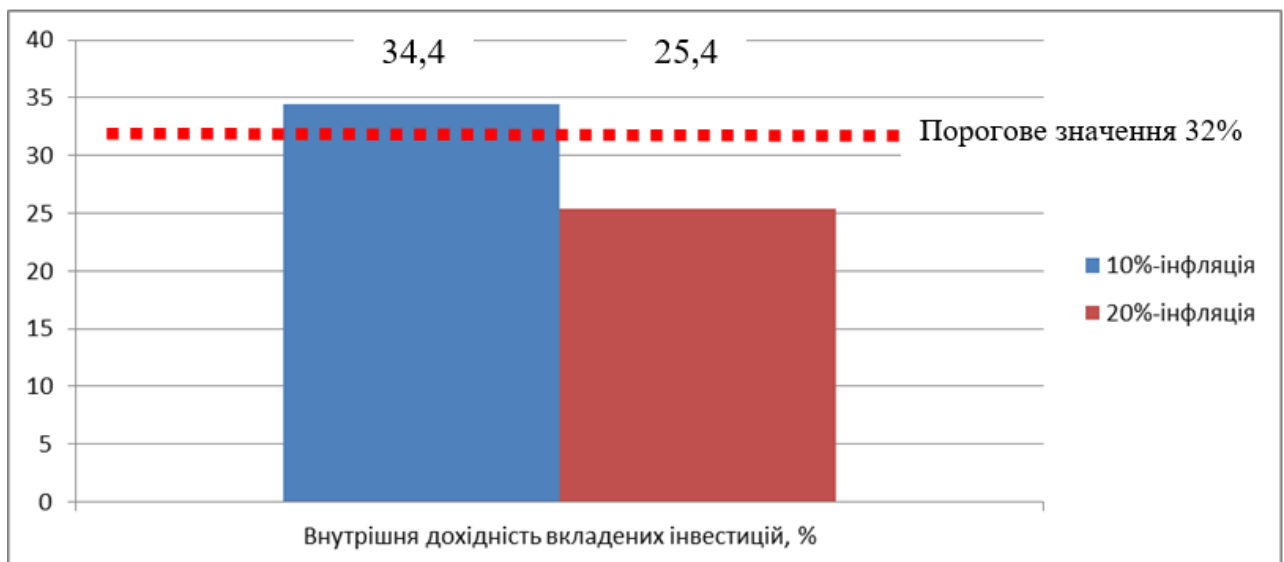


Рисунок 5.1 – Моделювання залежності величини внутрішньої дохідності потенційних інвестицій від рівня інфляції в країні

Аналіз графіка на рис 5.1 показує, що при рівні інфляції в 10% величина внутрішньої дохідності інвестицій становить  $E_B = 34,4\%$ , що більше порогового

значення  $\text{мін} = 32\%$  і тому комерціалізація нашої розробки може бути доцільною; при рівні інфляції в  $20\%$  величина внутрішньої дохідності інвестицій, вкладених в комерціалізацію нашої розробки, становить всього  $E_v = 25,4\%$ , що менше порогового значення  $\text{мін} = 32\%$ , і тому комерціалізація нашої розробки (при нинішніх умовах) буде для інвестора проблематичною. Для прийняття остаточного рішення потрібно провести додаткові обґрунтування і розрахунки (наприклад, знизити рівень прийнятого ризику, підняти ціну реалізації нашої розробки тощо).

#### 5.4 Висновки до розділу

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю 5.6:

Таблиця 5.6 – Результати економічної частини

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку	Не більше 200 тис. грн	192 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження розробки (в майбутній вартості грошей), тисяч грн	Не менше 2000 тисяч грн (за три роки)	2174 тисяч грн	Виконано
3. Внутрішня дохідність інвестицій, %	не менше 32%	34,4%	Досягнуто
4. Термін окупності інвестицій, роки	до 3-ти років	2,91 років	Виконано

Таким чином, основні техніко-економічні показники розробленого нами бота для нотифікації системних помилок обмінника валют «Exchanger», визначені у технічному завданні, виконані.

## ВИСНОВКИ

З впровадженням чат-ботів в різноманітні галузі від бізнесу до клієнтського обслуговування відкриваються нові можливості для оптимізації процесів та поліпшення взаємодії між користувачами та системами. Зростання інтересу до чат-ботів свідчить про те, що вони стають не тільки ефективним інструментом для автоматизації, але і важливим чинником поліпшення користувацького досвіду та розвитку сучасних технологій. Подальший розвиток та вдосконалення чат-ботів може відкрити нові перспективи для інтеракції людей з технологічним середовищем.

Результатом виконання даної дипломної роботи є розроблений та впроваджений telegram-бот для нотифікації системних помилок обмінника валют Exchanger. Метою роботи було покращення роботи обмінника валют шляхом автоматизації системи управління для розробки та підтримки проекту.

Перший розділ присвячено дослідженню предметної області, огляду аналогів та визначенню актуальності розробки даного telegram-бота, а також дослідженню типів чат-ботів, що допоміго визначити стратегію та основні функції розроблюваного програмного продукту. Відповідно до результатів проведеного порівняльного аналізу систем аналогів, було виявлено, що у обмінників крипто та фіатних валют Ramp і Moonpay відсутні боти нотифікації системних помилок, що підтверджує актуальність розробки запропонованого telegram-бота.

Технологічні засоби для розробки telegram-бота було розглянуто у другому розділі дипломної роботи. Основна увага була зосереджена на огляді та порівнянні мов програмування і бібліотек для розробки чат-ботів. Також досліджено особливості реляційних та нереляційних баз даних, їхніх систем управління базами даних. На основі отриманих результатів обрано Python як мову програмування, бібліотеку Telebot для забезпечення роботи з Telegram API та MySQL як систему управління базою даних. Обраний стек технологій забезпечує надійну, швидку та стабільну роботу розробленого чат-бота.

Третій та четвертий розділи присвячено опису основних етапів розробки бота для нотифікації системних помилок обмінника валют Exchanger, тестуванню функціоналу розробленого чат-бота. За результатами тестування визначено відповідність поставленим вимогам для розробки чат-боту, високу якість telegram-бота та його здатність ефективно нотифікувати про системні помилки обмінника валют Exchanger.

У рамках економічної частини дипломної роботи був проведений технологічний аудит розробленого бота для нотифікації системних помилок обмінника валют, розраховано витрати на розробку програмного продукту та економічний ефект від комерціалізації розробки.

В результаті аудиту та розрахунків було зазначено позитивні аспекти технічної ефективності бота, а також визначено потенційні можливості для підвищення економічного ефекту в разі його комерційного використання. Висновки економічної частини допомагають визначити перспективність та цілеспрямованість подальшого розвитку та вдосконалення розробленого чат-бота.

Впровадження розробленого telegram-бота для нотифікації системних помилок Exchanger визначає позитивний вплив на роботу обмінника валют, забезпечуючи ефективну автоматизацію підтримки проекту. Розроблений програмний продукт може бути масштабований для більш широкого функціоналу залежно від потреб обмінника.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Python: The Most Advanced Programming Language for Computer Science Applications. Dhruv, A., Patel, R. and Doshi, N. Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies (CESIT 2020), 2022, pages 292-299
2. Learning MySQL: Get a Handle on Your Data. Vinicius M. Grippa, Sergey Kuzmichev. O'Reilly Media, 2021, 632p.
3. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. / Укладачі В.О. Козловський, О.Й. Лесько, В.В.Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
4. Ramp. URL: <https://ramp.network/about> (дата звернення 27.09.2023)
5. MoonPay. URL: <https://www.moonpay.com/> (дата звернення 27.09.2023)
6. TIobe Index for June 2022. URL: <http://www.tiobe.com/tiobe-index/> (дата звернення 28.09.2023)
7. Requests. URL: <https://pypi.org/project/requests/> (дата звернення 30.09.2023)
8. What is an API: Definition, Types, Specifications, Documentation. URL: <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/> (дата звернення 01.10.2023)
9. Aiogram. URL: <https://pypi.org/project/aiogram/> (дата звернення 02.10.2023)
10. NoSQL databases. URL: <https://pypi.org/project/aiogram/> (дата звернення 01.12.2023)
11. Sean B. Chatbot History: What is Dr. Sbaitso. URL: <https://yakbots.com/chatbot-history-what-is-dr-sbaitso/> (дата звернення 25.09.2023)
12. What is a chatbot. URL: <https://www.oracle.com/chatbots/what-is-a-chatbot/> (дата звернення 25.09.2023)

13. Core.telegram – Документація по створенню Telegram-ботів. URL: <https://core.telegram.org/bots> (дата звернення 02.10.2023)
14. Chatbots and the role of AI. URL: <https://chatbotslife.com/chatbots-and-the-role-of-ai-448f75b291fe> (дата звернення 01.10.2023)
15. What are the benefits of using chatbots? URL: <https://www.marutitech.com/benefits-chatbot/> (дата звернення 29.09.2023)
16. Підходи до створення інтелектуальних чат-ботів. Ушакова І.О., Харків: Системи обробки інформації, 2019, випуск 2 (157), 76-83 с.
17. Mastering Bitcoin: Unlocking Digital Cryptocurrencies 1st Edition. Andreas Antonopoulos., O'Reilly Media, 2015, 296 с.
18. Створення чат-бота. URL: <https://www.unite.ai/uk/creating-a-chatbot-from-scratch-a-beginners-guide/> (дата звернення 03.10.2023)
19. Переваги використання Python для розробки чат-ботів і розмовного ШІ. URL: <https://ts2.space/uk/переваги-використання-python-для-розробки/> (дата звернення 03.10.2023)
20. Рейтинг найкращих криптообмінників в Україні. URL: <https://meta.ua/uk/exchangers/> (дата звернення 28.09.2023)
21. Telegram Push Notifications: A Guide to Telegram Bot Push Notification. URL: <https://respond.io/blog/telegram-push-notifications> (дата звернення 02.10.2023)
22. Setup Telegram bot to get alert notifications. URL: <https://medium.com/linux-shots/setup-telegram-bot-to-get-alert-notifications-90be7da4444> (дата звернення 01.10.2023)
23. Build a chat bot from scratch using Python and TensorFlow. URL: <https://handsonai.medium.com/build-a-chat-bot-from-scratch-using-python-and-tensorflow-fd189bcfae45> (дата звернення 02.10.2023)
24. Telegram. URL: <https://telegram.org/> (дата звернення 21.11.2023)

**Додатки**



**Додаток А (обов'язковий)****Технічне завдання**

Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

\_\_\_\_\_ д.т.н., проф. Олег БІСКАЛО

«\_\_» \_\_\_\_\_ 2023 року

**ТЕХНІЧНЕ ЗАВДАННЯ**

на магістерську кваліфікаційну роботу

УДОСКОНАЛЕННЯ СИСТЕМИ ОБМІНУ ВАЛЮТ EXCHANGER ШЛЯХОМ  
ІМПЛЕМЕНТАЦІЇ БОТУ ДЛЯ НОТИФІКАЦІЇ СИСТЕМНИХ ПОМИЛОК

08-31.МКР.019.02.000 ТЗ

Керівник: к.т.н., доц. каф. АІТ

\_\_\_\_\_ Володимир ГАРМАШ

«\_\_» \_\_\_\_\_ 2023 р.

Розробив студент гр. 1АКІТ-22м

\_\_\_\_\_ Ілля ЯРОВИЙ

«\_\_» \_\_\_\_\_ 2023 р.

### 1. Назва та галузь застосування.

Удосконалення системи обміну валют Exchanger шляхом імплементації боту для нотифікації системних помилок. Інформаційні системи та технології.

### 2. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №\_\_ по ВНТУ від «\_\_» \_\_\_\_\_2023р., та індивідуальне завдання на МКР, затверджене протоколом №\_\_ засідання кафедри АІТ від «\_\_» \_\_\_\_\_ 2023р.

### 3. Мета та призначення роботи.

Метою дослідження є покращення роботи обмінника валют Exchanger шляхом імплементації бота, який автоматично відстежуватиме системні помилки та збої в роботі продукту, а також надаватиме оперативні повідомлення та звіти розробникам та службі підтримки

### 4. Джерела розробки:

1) Python: The Most Advanced Programming Language for Computer Science Applications. Dhruv, A., Patel, R. and Doshi, N. Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies (CESIT 2020), 2022, pages 292-299

2) Core.telegram – Документація по створенню Telegram-ботів. URL: <https://core.telegram.org/bots> (дата звернення 02.10.2023)

3) Setup Telegram bot to get alert notifications. URL: <https://medium.com/linux-shots/setup-telegram-bot-to-get-alert-notifications-90be7da4444> (дата звернення 01.10.2023)

### 5. Показники призначення

Основні технічні вимоги та мінімальні системні вимоги до програми:

- стабільний зв'язок з мережею Інтернет;
- встановлений Telegram бот;
- версія Android від 8, або IOS від 10, або Windows від 8.

Вихідні дані для проведення робіт:

API обмінника валют Exchanger, API системи технічної підтримки, стек технологій для розробки продукту; месенджер Telegram та Telegram API.

Методи дослідження:

В даній роботі використовуються аналітичні методи та методи розробки ботів для месенджерів

Результати роботи програми:

Сповіщення про системні помилки обмінника валют успішно надходять у телеграм-бот, можливе створення запиту до техпідтримки безпосередньо з телеграм-боту, сповіщення про закриття звернення техпідтримкою надходять до чат-боту.

---

## 6. Економічні показники

До економічних показників входять:

- витрати на розробку – до 20 тис. грн.

- мінімальна дохідність – не менше 60 тис. грн.

- термін окупності – не більше 1 року

## 7. Стадії розробки:

а) Дослідження предметної галузі	<u>20.09.23</u> – <u>02.10.23</u>
б) Обґрунтування вибору технологічного стеку	<u>03.10.23</u> – <u>09.10.23</u>
в) Розробка програмного забезпечення	<u>10.10.23</u> – <u>10.11.23</u>
г) Тестування розробленого програмного забезпечення	<u>11.11.23</u> – <u>17.11.23</u>
д) Підготовка економічної частини	<u>18.11.23</u> – <u>24.11.23</u>
е) Оформлення пояснювальної записки	<u>25.11.23</u> – <u>04.12.23</u>

## 8. Порядок контролю та приймання

Рубіжний контроль провести до «\_\_» \_\_\_\_\_ 2023 р.

Попередній захист МКР провести «\_\_» \_\_\_\_\_ 2023 р.

Захист МКР провести до «\_\_» \_\_\_\_\_ 2023 р.

Розробив студент групи 1АКІТ-22м \_\_\_\_\_ Ілля ЯРОВИЙ

Додаток Б (обов'язковий)  
Графічна частина

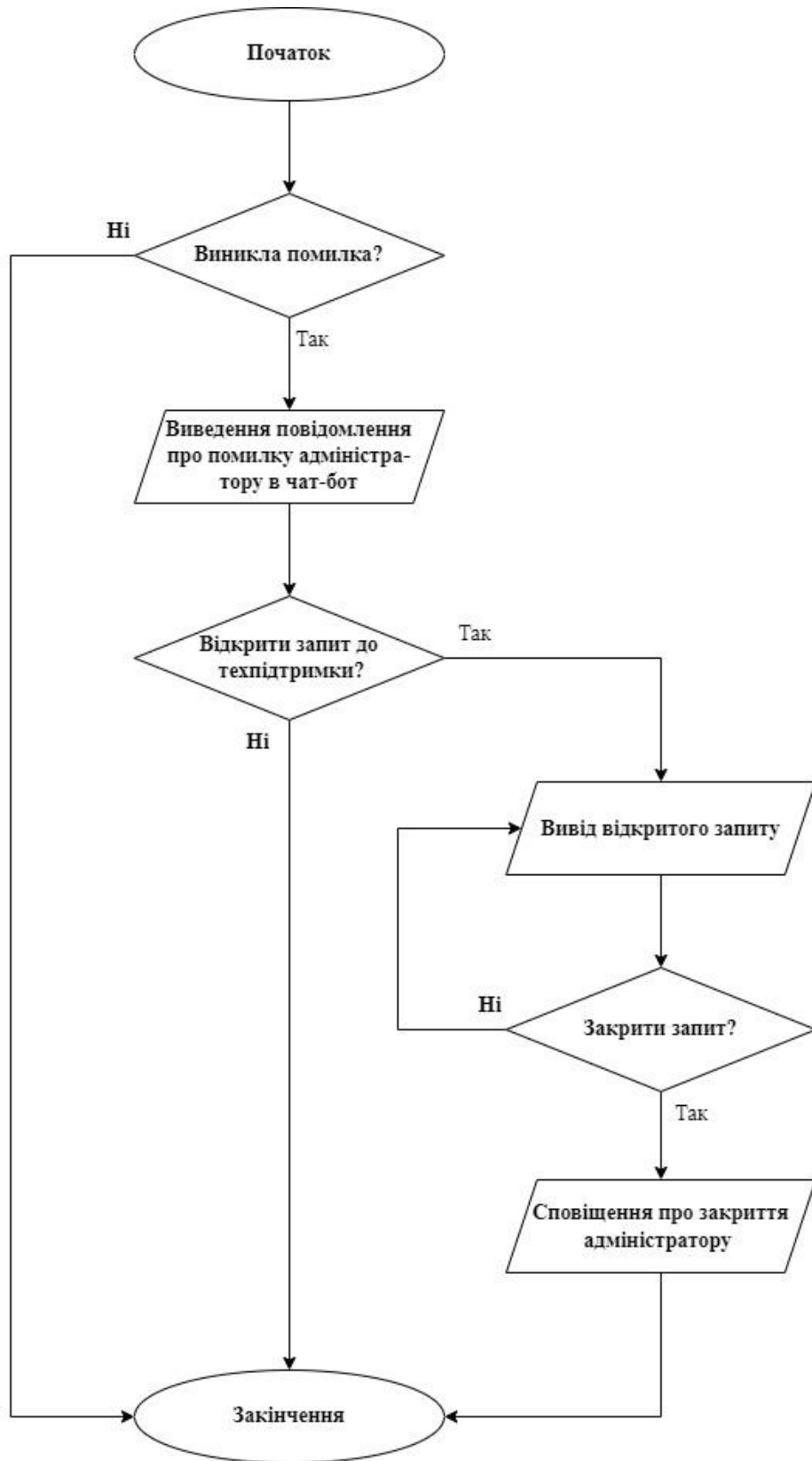


Рисунок Б.1 – Блок-схема роботи телеграм-боту

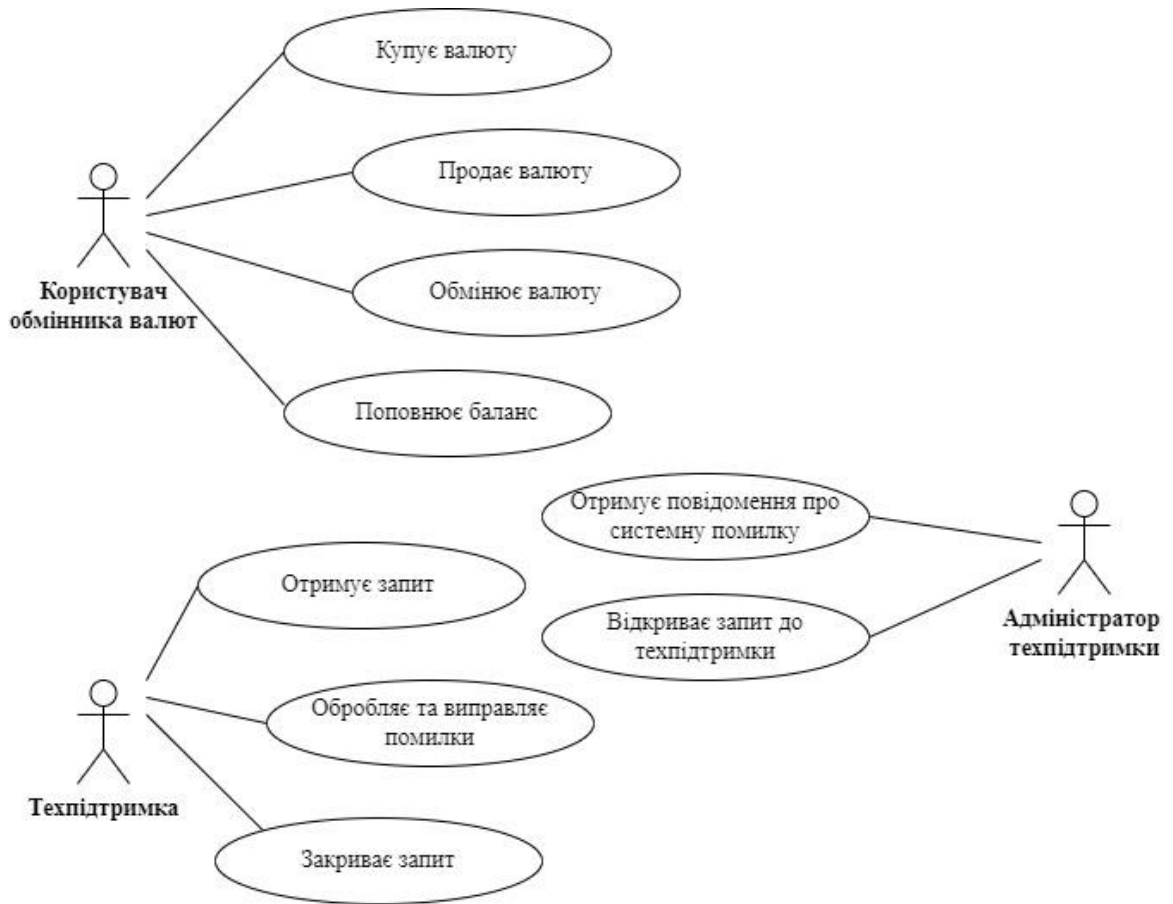


Рисунок Б.2 – UML-діаграма прецидентів

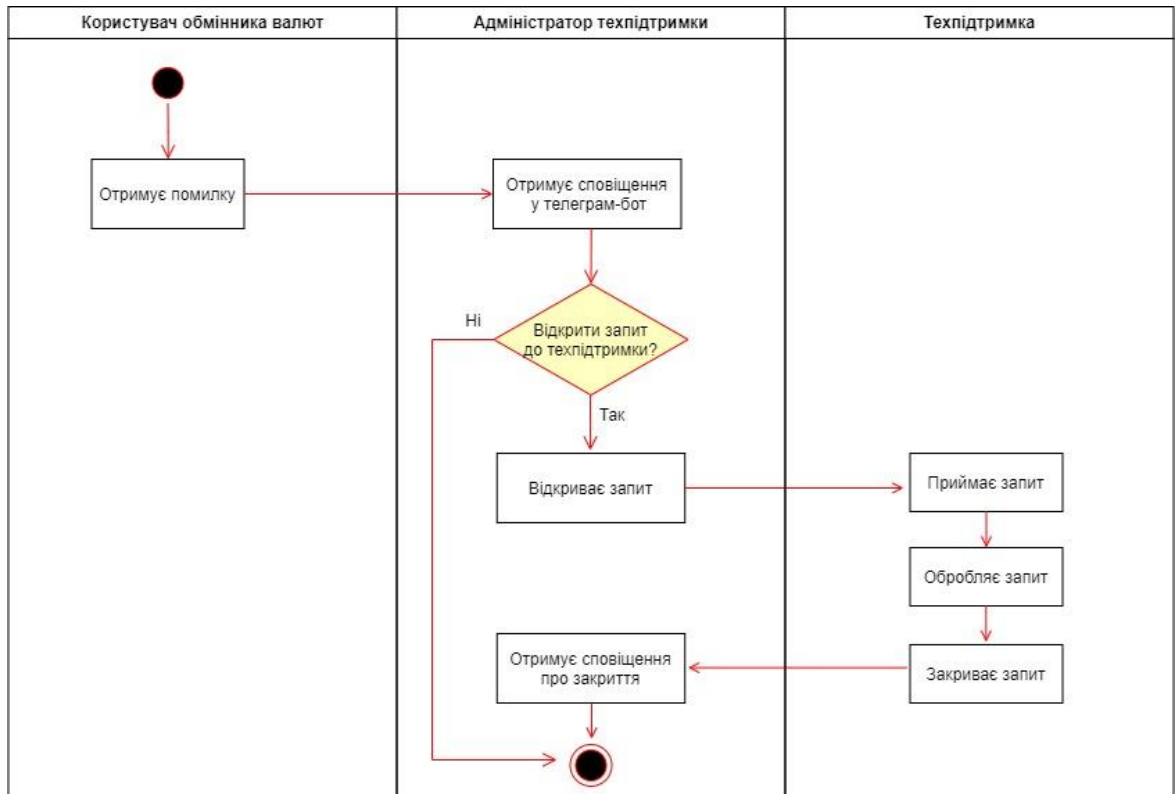


Рисунок Б.3 – UML-діаграма діяльності

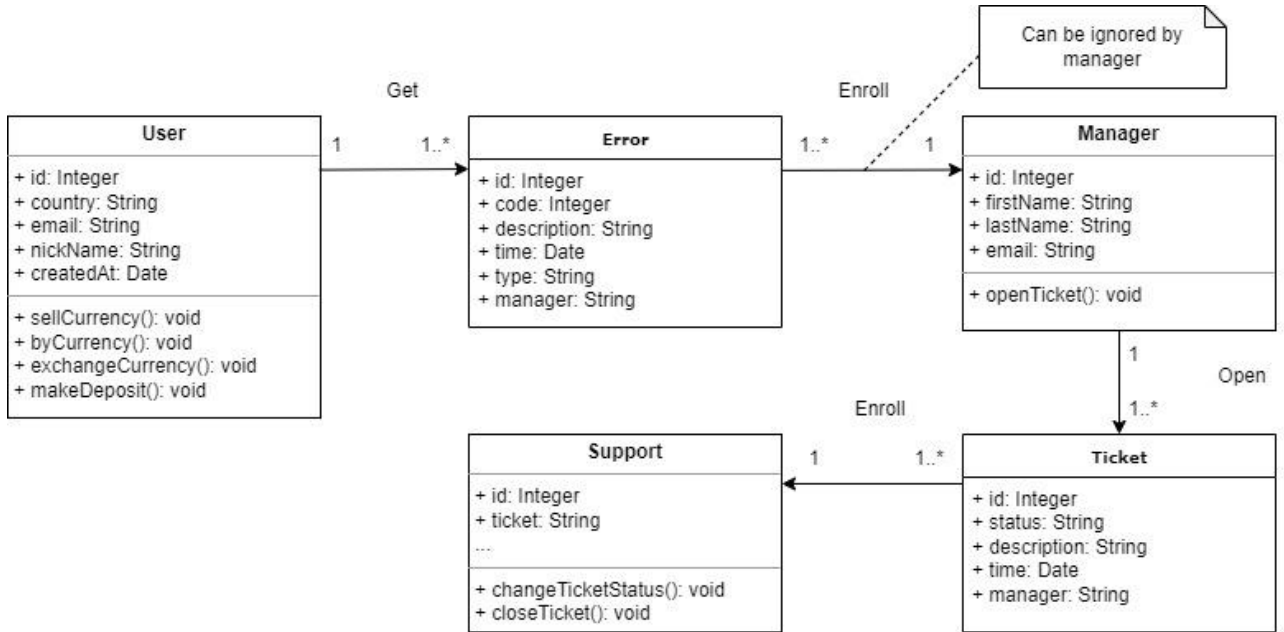


Рисунок Б.4 – UML-діаграма класів

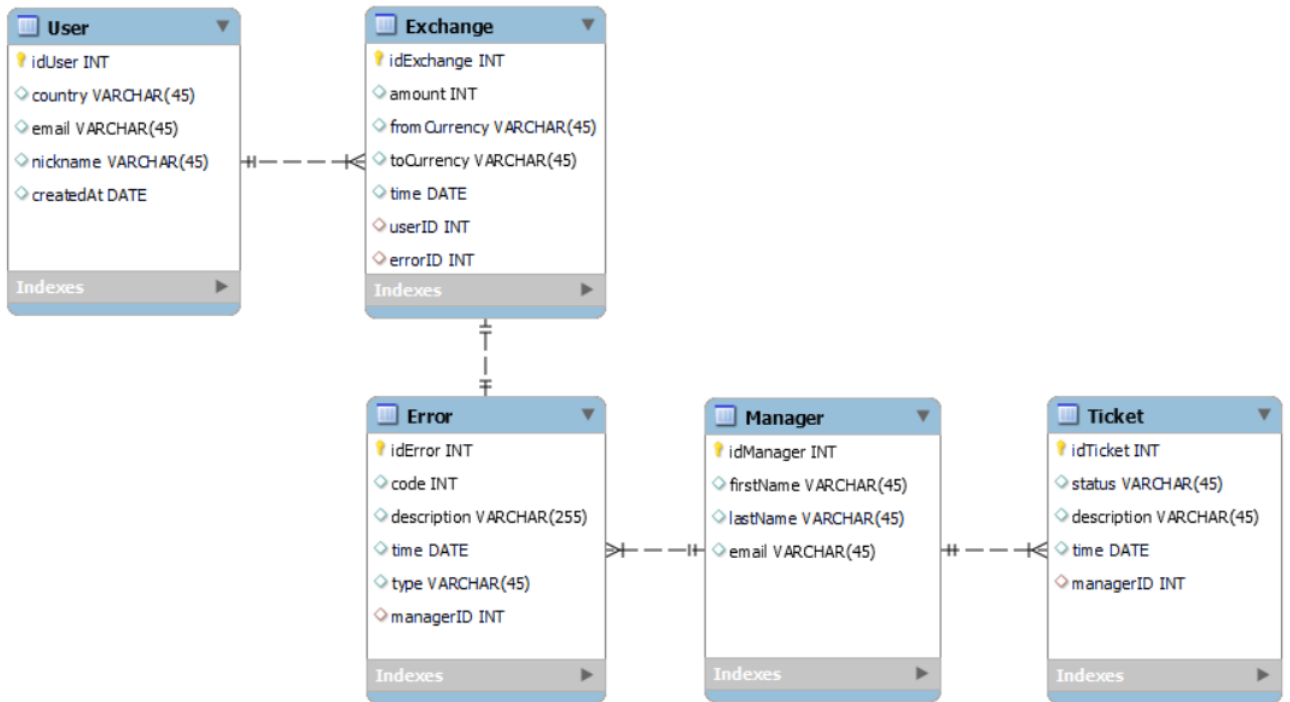


Рисунок Б.5 – Схема бази даних

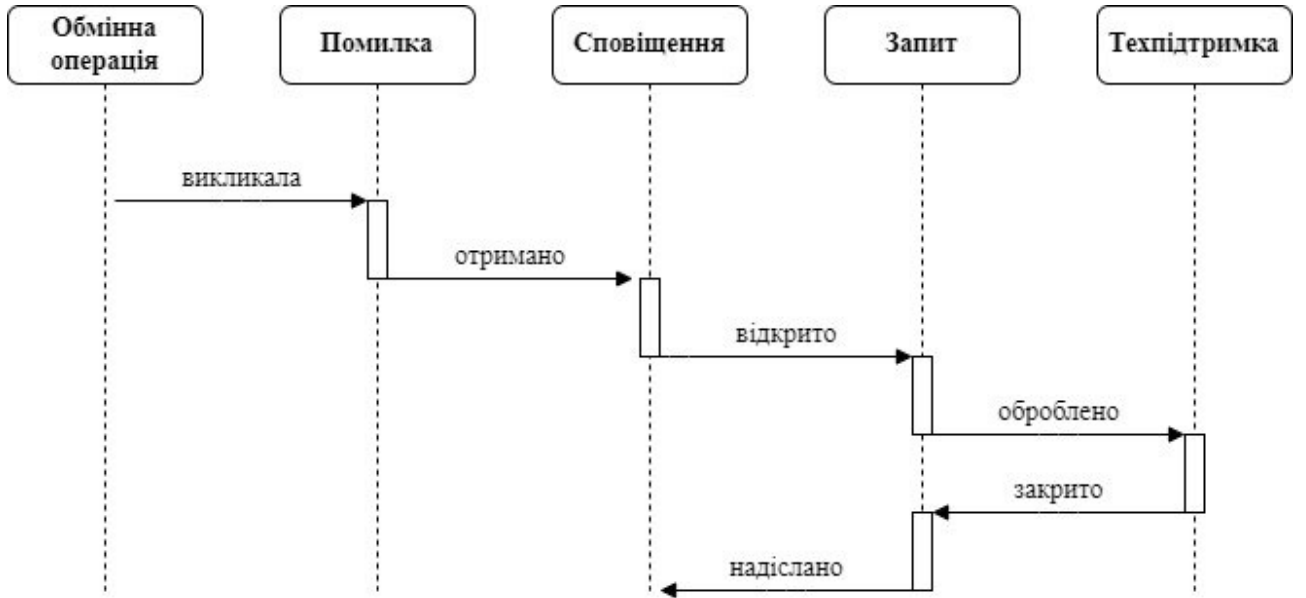


Рисунок Б.6 – UML-діаграма послідовності

**Додаток В (обов'язковий)**  
**Лістинг програмного забезпечення**

```
from datetime import time
from typing import Callable
import mysql.connector
from mysql.connector import Error
import telebot
from telebot import types
import requests

bot = telebot.TeleBot('BOT_TOKEN')
support_service = 'https://adminpanel.stg/create_ticket'

def create_db_connection():

    try:
        connection = mysql.connector.connect(
            host='DATABASE_HOST',
            database='DATABASE_NAME',
            user='DATABASE_USER',
            password='DATABASE_PASSWORD'
        )
        if connection.is_connected():
            print(f'Connected to MySQL database: {connection.database}')
            return connection

    except Error as e:
        print(f'Error: {e}')
        return None

def handle_external_service_error(user_id, status_code, error_message):
```



```

keyboard = types.ReplyKeyboardMarkup(row_width=1, resize_keyboard=True)
create_ticket_button = types.KeyboardButton("Create ticket")
keyboard.add(create_ticket_button)

```

```

bot.send_message(
    user_id,
    f"External error.\nStatus code: {status_code}\nMessage: {error_message}",
    reply_markup=keyboard
)

```

```

def make_external_request(user_id):

```

```

    try:
        response = requests.get("https://exchanger/")

        # Checking for a successful status code
        response.raise_for_status()

        # If an error occurs, we generate an exception
        if response.json().get("error"):
            raise Exception(response.json().get("error"))

```

```

    except requests.exceptions.HTTPError as e:
        # We invoke a function to handle errors
        handle_external_service_error(user_id, response.status_code, str(e))

```

```

    except Exception as e:
        # If another error occurs, we call a function to handle errors
        handle_external_service_error(user_id, 500, str(e))

```

```

def create_ticket(user_id, order_type, status_code, message, time):

```

```

    try:
        data = {

```

```

        'user_id': user_id,
        'order_type': order_type,
        'status_code': status_code,
        'message': message,
        'time': time
    }
    response = requests.post(support_service, json=data)
    response.raise_for_status()
    bot.send_message(user_id, "Successful create ticket!")

except requests.exceptions.RequestException as e:
    bot.send_message(user_id, f'Can't create a ticket. Error: {str(e)}")

@bot.message_handler(func=lambda message: True)
def handle_create_ticket_button(message):
    if message.text == "Create ticket":
        user_id = message.chat.id
        order_type = message.chat.type
        status_code = message.chat.code
        message_text = message.chat.text
        current_time = message.chat.time

        create_ticket(user_id, order_type, status_code, message_text, current_time)

def message_deletor(self, time_gap: int, chat_id: str, message_id: str) -> str:
    """
    Timed deletion of a message, time range: [0, 900], in seconds
    """
    if time_gap < 0 or time_gap > 900:
        return "time_gap_error"
    else:
        def message_deletor_func(time_gap, chat_id, message_id):
            time.sleep(int(time_gap))

```

```

self.deleteMessage(chat_id=chat_id, message_id=message_id)

if time_gap == 0:
    message_deletor_func(chat_id, message_id)
else:
    fur = self.__timer_thread_pool.submit(
        message_deletor_func, time_gap, chat_id, message_id)
    fur.add_done_callback(self.__threadpool_exception)

return "ok"

def timer(self, time_gap: int, func: Callable[..., None], args: tuple) -> str:
    """
    Single timer, time range: [0, 900], unit seconds
    """
    if time_gap < 0 or time_gap > 900:
        return "time_gap_error"
    elif type(args) is not tuple:
        return "args_must_be_tuple"
    else:
        def timer_func(time_gap, func, args):
            time.sleep(int(time_gap))
            func(*args)

        if time_gap == 0:
            func(args)
        else:
            fur = self.__timer_thread_pool.submit(
                timer_func, time_gap, func, args)
            fur.add_done_callback(self.__threadpool_exception)

    return "ok"

bot.polling(none_stop=True)

```

## Додаток Г (обов'язковий)

### Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Удосконалення системи обміну валют Exchanger шляхом імплементації боту для нотифікації системних помилок»

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра АІТ

#### Показники звіту подібності Unicheck

Оригінальність 99,0 %

Схожість 1,0 %

Аналіз звіту подібності (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
  - Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
  - Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку \_\_\_\_\_  
(підпис)

Роман МАСЛІЙ

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи \_\_\_\_\_  
(підпис)

Ілля ЯРОВИЙ

Керівник роботи \_\_\_\_\_  
(підпис)

Володимир ГАРМАШ