

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ” 2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

на тему: Чат-бот для пошуку необхідної інформації про допомогу

Виконав: студент 4 курсу, групи ІВ-93

(шифр групи)

Мельник Антон Станіславович

(прізвище, ім’я, по батькові)

(підпис)

Керівник

Луцький Г. М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант

(нормоконтроль) ст. викл. Виноградов Ю. М.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент

(підпис)

Національний технічний університет України
«Київський політехнічний інститут»
ім. Ігоря Сікорського
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки
Рівень вищої освіти – перший (бакалавр)
Освітньо-професійна програма
“Комп’ютерні системи та мережі”
спеціальність 123 «Комп’ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
С. Г. Стіренко
(підпис) (ініціали, прізвище)
“ ____ ” _____ 2023 р.

ЗАВДАННЯ
на бакалаврський дипломний проект студента

Мельника Антона Станіславовича
(прізвище, ім’я, по батькові)

1. Тема проекту (роботи) Чат-бот для пошуку необхідної інформації про допомогу

керівник проекту (роботи) Луцький Г. М., к.т.н., доцент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом по університету від “ ____ ” _____ 20__ р. №__

2. Термін здачі студентом закінченого проекту (роботи)

3. Вихідні дані до роботи наукова та технічна документація про реалізацію окремих модулів системи для вибору найоптимальнішого рішення щодо реалізації поставленої задачі з розробки програмного забезпечення.

4. Зміст пояснювальної записки: опис предметної області і напрямків дослідження, аналіз та характеристика об’єкту досліджень, аналіз існуючих рішень, опис архітектури системи.

5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо):

Принципова схема системи, структурна схема системи, діаграма послідовності.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	ст. викл. Виноградов Ю. М.		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Затвердження теми роботи	06.03.2023-07.03.2023	
2.	Вивчення та аналіз завдання	07.03.2023-13.04.2023	
3.	Написання вступної частини та огляду предметної області	13.04.2023-23.04.2023	
4.	Розробка архітектури системи	23.04.2023-01.05.2023	
5.	Програмна реалізація системи	01.05.2023-12.05.2023	
6.	Виправлення помилок	12.05.2023-01.06.2023	
7.	Оформлення документації дипломної роботи	01.06.2023-03.06.2023	
8.	Передзахист	-	
9.	Захист	-	

Студент

(підпис)

Мельник А. С.
(ініціали, прізвище)

Керівник проекту

(підпис)

Луцький Г. М.
(ініціали, прізвище)

АННОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (60 с., 18 рис., 4 додатки)

Об'єкт розробки: чат-бот месенджеру Telegram.

Метою дипломного проекту є дослідження використання телеграм ботів для забезпечення допомоги людям під час війни. А також розробка відповідного бота на платформі Telegram.

Під час розробки:

- проведений аналіз предметної області;
- проведений аналіз засобів та методик створення чат-ботів;
- виконано розробку чат-боту.

Розробка чат-боту виконана за допомогою мови програмування Python та її бібліотек Telebot та requests.

Ключові слова: чат-бот, Telegram, Python, API.

ANNOTATION

The qualification work includes an explanatory note (60 p., 18 figures, 4 appendixes)

Object of development: Telegram messenger chatbot.

The aim of the thesis project is to investigate the use of telegram bots to provide assistance to people during wartime. As well as the development of a corresponding bot on the Telegram platform.

During development:

- the analysis of the subject area was carried out;
- an analysis of means and methods of creating chatbots was carried out;
- development of a chat bot was completed.

The chatbot was developed using the Python programming language and its Telebot and requests libraries.

Keywords: chatbot, Telegram, Python, API.

№ рядка	Формат	Позначення			Найменування			Кільк.	Примітка
					Документація загальна				
	A4	ІАЛЦ.467200.001 ОА			Чат-бот для пошуку необхідної інформації про допомогу Опис альбому			2	
	A4	ІАЛЦ.467200.002 ТЗ			Чат-бот для пошуку необхідної інформації про допомогу Технічне завдання			3	
	A4	ІАЛЦ.467200.003 ПЗ			Чат-бот для пошуку необхідної інформації про допомогу Пояснювальна записка			56	
	A4	ІАЛЦ.467200.004 Д1			Чат-бот для пошуку необхідної інформації про допомогу Принципова схема			1	
	A4	ІАЛЦ.467200.005 Д2			Чат-бот для пошуку необхідної інформації про допомогу Структурна схема			1	
	A4	ІАЛЦ.467200.006 Д3			Чат-бот для пошуку необхідної інформації про допомогу Діаграма класів			1	
	A4	ІАЛЦ.462619.007 Д4			Чат-бот для пошуку необхідної інформації про допомогу Програмний код			6	
					ІАЛЦ.467200.001 ОА				
Зм.	Арк.	№ докум.	Підпис	Дата					
Розробив	Мельник А. С.				Чат-бот для пошуку необхідної інформації про допомогу Опис альбому		Літ.	Аркуш	Аркушів
Перевірив								1	1
Реценз.							КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-93		
Н. контр.									
Затвердив									

ЗМІСТ

1.	НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2.	ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
3.	МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4.	ДЖЕРЕЛА РОЗРОБКИ	2
5.	ТЕХНІЧНІ ВИМОГИ	3
5.1.	Вимоги до програмного продукту, що розробляється.....	3
5.2.	Вимоги до інструментального програмного забезпечення	3
6.	ЕТАПИ РОЗРОБКИ.....	3

					ІАЛЦ.467200.003 ТЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Мельник А. С.			Чат-бот для пошуку необхідної інформації про допомогу Технічне завдання	Літ.	Аркуш
Перевірив							Аркушів
Реценз.						1	3
Н. Контр.						КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-93	
Затвердив							

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Областю застосування розроблюваної системи є чат-боти на основі популярних месенджерів.

Метою розробки системи є використання її у повсякденному житті та створення альтернативи іншим ботам такого формату.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського дипломного проекту, затверджене кафедрою обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою розробки постає створення чат-боту для пошуку необхідної інформації про допомогу в одному з найпопулярніших месенджерів, який буде доступним і зрозумілим для будь-яких користувачів.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами розробки є науково-технічна література, технічна документація, публікації в періодичних виданнях та мережі Інтернет.

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

Система, що розроблюється, повинна відповідати поставленим вимогам, що наведені нижче:

- Простий і зрозумілий інтерфейс системи;
- Технічна коректність;
- Актуальна інформація;
- Швидкий доступ до системи;

5.2. Вимоги до інструментального програмного забезпечення

Для розробки проекту на локальному ПК необхідне наступне програмне забезпечення:

- Операційна система Windows або сімейств Linux та MacOS;
- Встановлений месенджер Telegram;
- Встановлений Git;

6. ЕТАПИ РОЗРОБКИ

Назва етапів виконання	Термін виконання
Затвердження теми роботи	06.03.2023-07.03.2023
Вивчення та аналіз завдання	07.03.2023-13.04.2023
Написання вступної частини та огляду предметної області	13.04.2023-23.04.2023
Розробка архітектури системи	23.04.2023-01.05.2023
Програмна реалізація системи	01.05.2023-12.05.2023
Виправлення помилок	12.05.2023-01.06.2023
Оформлення документації дипломної роботи	01.06.2023-03.06.2023
Передзахист	-
Захист	-

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	5
1.1 Актуальність	5
1.1.1 Мета дослідження	5
1.1.2 Розробка завдань	6
1.2 Сфери застосування	6
1.3 Концепція чат-бота	7
1.4 Чат бот у контексті Telegram	8
1.4.1 Аналіз допоміжних програмних засобів та засобів розробки	11
1.5 Огляд існуючих рішень	15
1.6 Аналіз вимог до програмного забезпечення	20
1.6.1 Функціональні вимоги	20
1.6.2 Нефункціональні вимоги	21
Висновок до розділу 1	23
РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	24
2.1 Вибір мови програмування	24
2.2 Архітектура програмного забезпечення	25
2.2.1 Telegram Bot API	25
2.2.2 Діаграма діяльності	26
2.2.3 Діаграма послідовності	29
2.3 Опис функцій	30
2.3.1 Опис кнопок телеграм бота	31
Висновок до розділу 2	33

					ІАЛЦ 467200.003 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив		Мельник А. С.			Чат-бот для пошуку необхідної інформації про допомогу Пояснювальна записка	Лит.	Аркуш	Аркушів
Перевірив							1	56
Т. Контр.						КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-93		
Н. Контр.								
Затвердив								

РОЗДІЛ 3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	34
3.1 Опис процесів тестування	34
3.2 Аналіз якості ПЗ	39
3.3 Опис контрольного прикладу	40
Висновки до 3 розділу	47
 РОЗДІЛ 4. ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	 48
4.1 Розгортання програмного забезпечення	48
4.2 Підтримка програмного забезпечення	50
Висновки до 4 розділу	53
ВИСНОВКИ	54
ДОДАТКИ	55

СПИСОК СКОРОЧЕНЬ

ПК – персональний комп'ютер

ПЗ – програмне забезпечення

ОС – операційна система

IDE – Integrated Development Environment, інтегроване середовище розробки

API – Application Programming Interface, прикладний програмний інтерфейс,

SDK – Software development kit

					ІАЛЦ 467200.003 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Під час війни та конфлікту доступ до критично важливих ресурсів, таких як притулок, гуманітарна допомога та можливості волонтерства, може бути складним для постраждалих осіб. Крім того, багато людей можуть бути готові запропонувати допомогу, але можуть не знати, з чого почати. Ця проблема є особливо гострою у випадку війни в Україні, де тисячі людей були переміщені та стикаються зі значними проблемами в доступі до допомоги.

Для вирішення цього питання в цій дипломній роботі пропонується розробка чат-бота для пошуку допомоги під час війни, з акцентом на війну в Україні. Чат-бот дозволить людям знайти інформацію про притулки, гуманітарну допомогу та можливості волонтерства, а також пропонувати житло, транспорт та будь-яку іншу допомогу, яка їм може знадобитися.

Ця дипломна робота буде включати підхід кейс-стаді, розглядаючи розробку та впровадження чат-бота для пошуку допомоги в умовах війни в Україні. Зокрема, він вивчатиме технічні вимоги до розробки чат-бота, такі як необхідне програмне та апаратне забезпечення, а також потенційні проблеми та етичні міркування розробки такої системи.

Крім того, дослідження оцінюватиме ефективність чат-бота у зв'язку людей із необхідними ресурсами та сприянні пропозиціям допомоги. Це буде зроблено за допомогою підходу змішаних методів, включаючи опитування та інтерв'ю з особами, які використовували чат-бот, а також аналіз даних про використання.

Загалом ця дипломна робота має на меті сприяти нашому розумінню потенціалу чат-ботів для пошуку допомоги під час війни, зосереджуючись на війні в Україні. Результати цього дослідження можуть мати значні наслідки для розвитку подібних систем в інших районах, уражених конфліктом, а також для ширшої сфери гуманітарної допомоги.

					ІАЛЦ 467200.003 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність

Неможливо переоцінити актуальність розробки чат-бота для цієї мети, особливо в кризові часи, такі як війна в Україні. Чат-бот може надавати допомогу в режимі реального часу та доступ до інформації, яка є важливою для людей, які терміново потребують допомоги.

Надаючи інформацію про притулки, гуманітарну допомогу та можливості волонтерства, чат-бот може допомогти швидко та ефективно зв'язати людей, які потребують, з необхідними ресурсами. Це може бути особливо критичним під час війни та конфлікту, коли люди можуть бути виселені зі своїх домів або зіткнутися з іншими проблемами, які потребують негайної допомоги.

Крім того, чат-бот може допомогти зв'язати людей, які готові запропонувати допомогу, наприклад проживання, транспорт чи інші види допомоги, з тими, хто її найбільше потребує. Це може створити відчуття спільності та допомогти людям підтримувати один одного у важкі часи.

Загалом, чат-бот для пошуку допомоги є важливим інструментом для тих, хто постраждав від війни в Україні. Це може допомогти покращити доступ до критично важливих ресурсів, з'єднати людей, які потребують допомоги, з тими, хто може допомогти, і, зрештою, значно змінити життя тих, хто постраждав від конфлікту. Тому вкрай важливо терміново розробити такий чат-бот, щоб допомогти тим, хто цього потребує.

1.1.1 Мета дослідження

Метою дипломного проекту є дослідження використання телеграм ботів для забезпечення допомоги людям під час війни. А також розробка відповідного бота на платформі Telegram.

					ІАЛЦ 467200.003 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

1.1.2 Розробка завдань

Для досягнення поставленої мети було розроблено такі завдання:

- дослідити актуальність теми
- дослідження сфер застосування
- визначення концепції телеграм бота
- зробити огляд існуючих рішень
- розробити функціональні та нефункціональні вимоги
- розробити програмний код для бота
- провести тестування бота

1.2 Сфери застосування

Чат-бот для пошуку допомоги можна застосовувати в різних сферах, де людям може знадобитися допомога, зокрема, але не обмежуючись цим:

1. **Гуманітарна допомога:** чат-бот може використовуватися гуманітарними організаціями для зв'язку людей, які постраждали від криз, таких як війна чи стихійні лиха, з необхідними ресурсами, включаючи їжу, притулок і медичну допомогу.
2. **Волонтерство:** чат-бот може сприяти волонтерству, надаючи інформацію про можливості волонтерства та підбираючи волонтерів до організацій, які потребують їхньої допомоги.
3. **Бездомність:** чат-бот може допомагати людям, що залишились без дому, надаючи інформацію про місця для постраждалих, банки їжі та інші ресурси.
4. **Догляд за людьми похилого віку:** чат-бот може допомагати людям похилого віку, які залишились самі внаслідок війни, надаючи інформацію про ресурси, такі як послуги догляду вдома або транспортна допомога.

Загалом чат-бот можна застосовувати в різних сферах для надання допомоги та зв'язку людей із необхідними ресурсами. Його потенціал

застосування величезний, і він може допомогти покращити доступ до критично важливих ресурсів, особливо для людей, які опинилися у складних ситуаціях або кризах.

1.3 Концепція чат-бота

Чат-бот — це комп'ютерна програма, призначена для імітації розмовиз людьми, як правило, через текстові взаємодії. Чат-боти можуть бути запрограмовані відповідати на введення користувача попередньо визначеними повідомленнями, або вони можуть використовувати обробку природної мови (NLP), щоб розуміти запити користувачів і відповідати на них у більш розмовній манері. [1]

Однією з концепцій чат-бота є віртуальний помічник, який допомагає користувачам знайти допомогу під час кризи, наприклад стихійного лиха чи війни. Чат-бот дозволить користувачам шукати такі ресурси, як притулок, їжа та медична допомога, а також зв'язувати їх із волонтерами чи іншими ресурсами в їхній місцевості. Чат-бот може використовувати геолокаційні дані, щоб ідентифікувати місцезнаходження користувачів і надавати їм інформацію про найближчі доступні ресурси. [1]

Чат-бот також міг би надавати користувачам інформацію про те, як залишатися в безпеці під час кризи, наприклад контактну інформацію для екстрених випадків і поради щодо того, як підготуватися до надзвичайних ситуацій. Крім того, чат-бот міг би надавати користувачам оновлену інформацію про стан кризи та будь-які нові події, такі як зміни в наказах про евакуацію або перекриття доріг. [2]

Інша концепція чат-бота — це віртуальний консьєрж для готелів чи інших підприємств гостинності. Чат-бот дозволить гостям робити запити або задавати запитання природною мовою, наприклад, замовляти обслуговування номерів або запитувати рекомендації щодо місцевих пам'яток. Чат-бот міг би використовувати алгоритми машинного навчання,

щоб вчитися на взаємодії гостей і з часом покращувати свої відповіді. Крім того, чат-бот може інтегруватися з іншими готельними системами, такими як системи резервування або бронювання, щоб надати гостям бездоганний досвід. [2]

Загалом, концепція чат-бота є потужним інструментом для надання персоналізованої допомоги та інформації користувачам у різних контекстах, чи то під час кризи, чи в готельній обстановці. Завдяки прогресу в НЛП і машинному навчанні чат-боти стають дедалі складнішими та здатними обробляти складні взаємодії, що робить їх цінним активом для підприємств і організацій, які прагнуть надавати виняткове обслуговування клієнтів.

1.4 Чат бот у контексті Telegram

Чат-бот у контексті Telegram – це сторонній додаток, який працює в екосистемі Telegram. Користувачі взаємодіють з ботами надсилаючи їм повідомлення, команди та inline-запити. Для керування Telegram-ботом існує спеціальне API, яке вимагає використання HTTPS-протоколу.

За допомогою Telegram-боту можливо [2]:

1. Отримання кастомізованих сповіщень та новин.
2. Інтеграція зі сторонніми сервісами: Gmail, Wiki, YouTube, Github тощо.
3. Отримання платежів від користувачів.
4. Створення корисних утиліт: перекладачі з різних мов, для форматування тексту, для нагадування про події тощо.
5. Розробка ігор, що базуються на технології HTML5.

Telegram-бот з точки зору месенджера – це спеціальний аккаунт, з яким користувачі можуть взаємодіяти двома способами:

1. Відкрити чат з ботом, або ж додати його до групи, після чого надсилати текст або команди через поле вводу повідомлень.
 2. Відкрити будь-який чат та вписати “@ім’я бота” в поле вводу повідомлень.
- Таким чином працюють з inline-ботами.

Варто зазначити, що:

1. Ініціювати спілкування з ботом може тільки користувач, але не навпаки. Це попереджає можливість спаму від недоброчесних ботів.
2. Боти мають обмежене хмарне сховище для повідомлень. Потрібно заздалегідь створити локальне сховище для зберігання застарілих повідомлень, якщо у цьому є необхідність.
3. Ім’я ботів завжди мають закінчення “bot”.

Подумайте про API для ботів Telegram. Сервери Telegram повинні схвалити кожного бота. Спеціальний єдиний у своєму роді маркер авторизації (далі -) створюється під час створення бота та надається разом із URL-адресою для кожного запиту (наприклад, «123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11»). Протокол HTTPS має використовуватися для всіх викликів API Telegram Bot, який доступний за адресою "https://api.telegram.org/bot/METHOD_NAME". Приймаються запити HTTP GET і POST. Методи включення параметрів у запит такі [2]:

1. Текст URL-запиту.
2. Application/x-www-form-urlencoded є другим.
3. Application/json, третє.
4. Дані в складі/формі.

Частина URL-адреси, яка називається рядком запиту URL-адреси, містить пари "ключ=значення", де ключ представляє певний параметр. Структура рядка запиту: URL?param1=1¶m2=2¶m3=3

де URL означає унікальний ідентифікатор ресурсу? безпосередньо відокремлює URL-адресу від рядка запиту, а & відокремлює кілька аргументів.

Тип тіла запиту HTTP application/x-www-form-urlencoded передає дані запиту у форматі "ключ=значення". Заголовок запиту Content-kind визначає тип. [3]

Вищезазначені методи можуть використовувати унікальне кодування символів, таке як кодування URL-адреси або кодування у відсотках. Ці кодування використовуються для вирішення проблеми використання заборонених символів (наприклад, пробілу в URL-адресі). Процес кодування включає заміну 8-бітових символів, які заборонені, на дозволені символи. На рис. 1.1 показано кодування символів. [3]

'	:	'	/	'	?	'	#	'	[	'	]	'	@	'	!	'	\$	'	&	'	"	'	(	'	)	'	*	'	+	'	,	'	;	'	=	'	%	'	'		
%3A	%2F	%3F	%23	%5B	%5D	%40	%21	%24	%26	%27	%28	%29	%2A	%2B	%2C	%3B	%3D	%25	%20	or	+																				

Рисунок 1.1 – Кодування символів по механізму URL encoding

Коли в URL-адресі використовується пробіл, він кодується як "%20", а коли використовується Application/x-www-form-urlencoded, він кодується як "+". Дані запиту у форматі JSON доставляються за допомогою типу тіла запиту HTTP application/json. Заголовок Content-Type запиту містить опис типу. JSON — це текстовий стандарт обміну даними для опису елементів і структур даних між комп'ютерами. Мова JavaScript має вбудовану підтримку JSON для серіалізації та десеріалізації об'єктів. [3]

Є два підтримуваних варіанти структур:

По-перше, пари ключ-значення.

2. Набір значень, організованих у вигляді масиву, вектора, послідовності тощо.

Тип тіла запиту HTTP multipart/form-data призначений для транспортування двійкових файлів. Заголовок Content-Type запиту містить опис типу.

RFC7578 містить детальне пояснення передачі файлів за допомогою протоколу HTTP. Telegram завжди містить логічне поле «ok» і текст «description» в об'єкті JSON, який він повертає у відповідь на виклик Telegram Bot API. [10]

Параметр "ok" ідентичний "true", якщо запит був успішним, і "false" в іншому випадку. Поле «description» містить опис помилки, якщо запит не виконано, а додаткове поле «error_code» доставляється разом із кодом помилки. Зокрема, нечутливість до регістру поширюється на всі функції API Telegram Bot, і потрібне кодування даних UTF-8. [3]

1.4.1 Аналіз допоміжних програмних засобів та засобів розробки

Існує кілька допоміжних програм і засобів розробки, які можна використовувати для прискорення та покращення процесу розробки програми для розділення файлів. Нижче наведено кілька найпопулярніших інструментів:

1. Інтегровані середовища розробки (IDE): такі IDE, як Eclipse, Visual Studio та NetBeans, надають ряд функцій, які роблять розробку швидшою та легшою, наприклад підсвічування коду, автозавершення та налагодження.

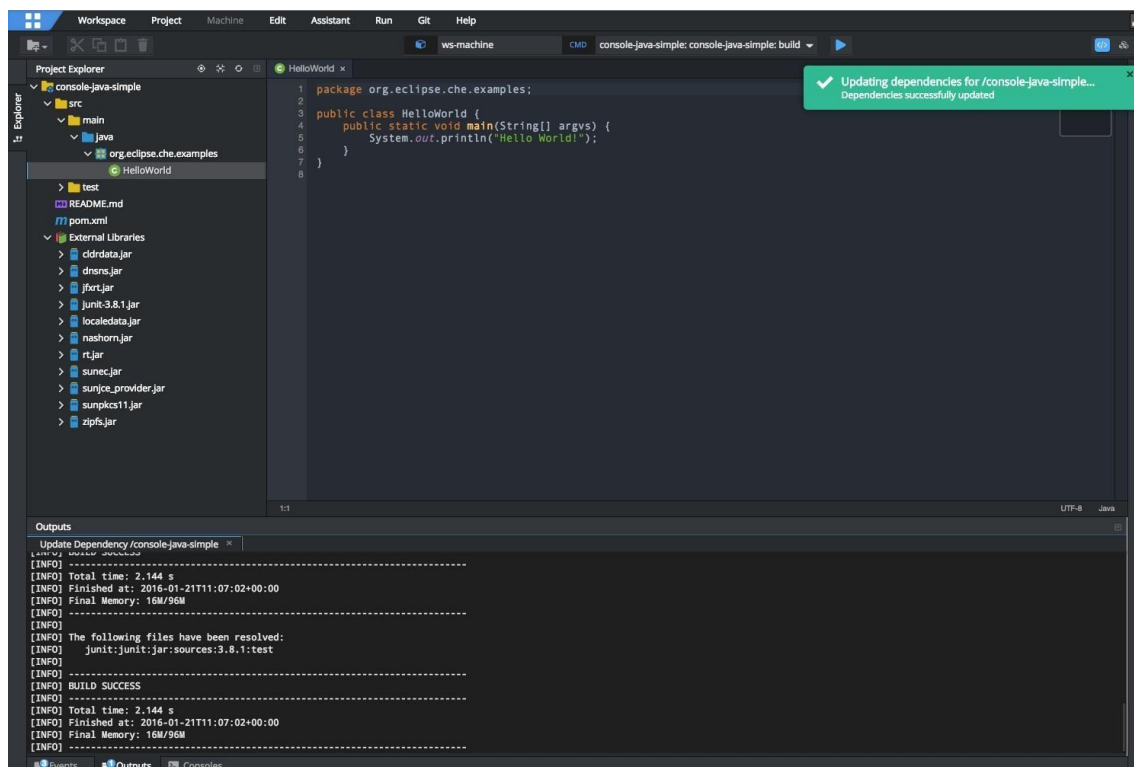


Рисунок 1.2 – Середовище розробки Eclipse

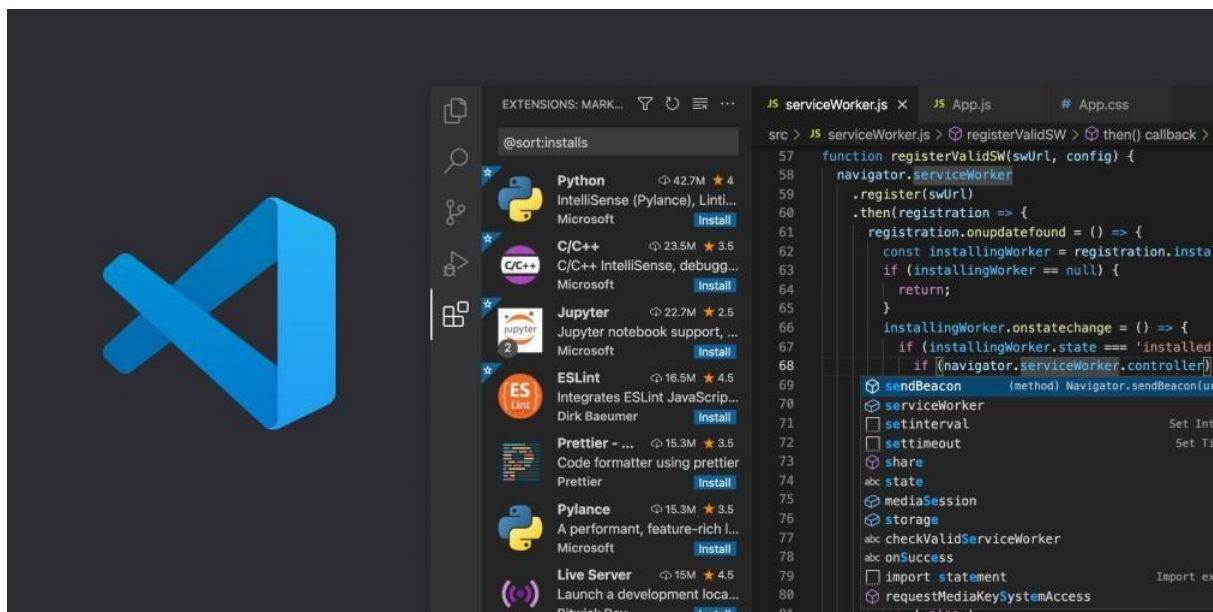


Рисунок 1.3 – Середовище розробки VS

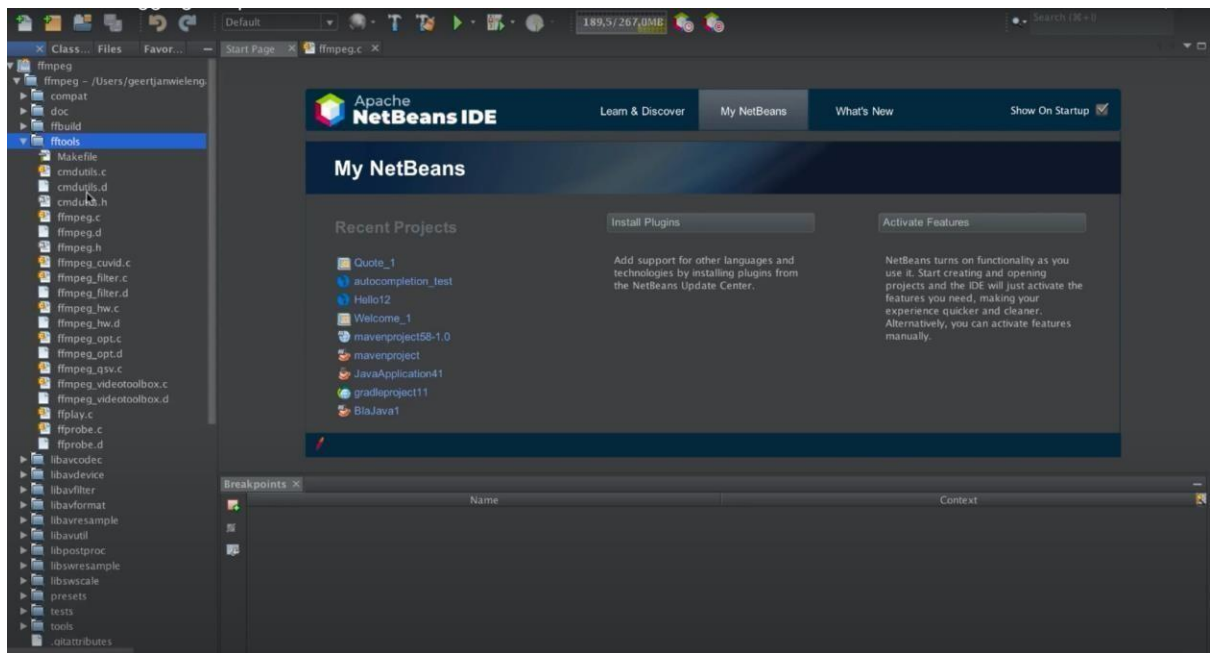


Рисунок 1.4 – Середовище розробки NetBeans

Обирати середовище розробки можна залежно від мови, вподобань, операційної системи тощо.

2. Системи контролю версій: системи контролю версій, такі як Git і SVN, використовуються для керування змінами в кодовій базі, відстеження помилок і співпраці з іншими розробниками.

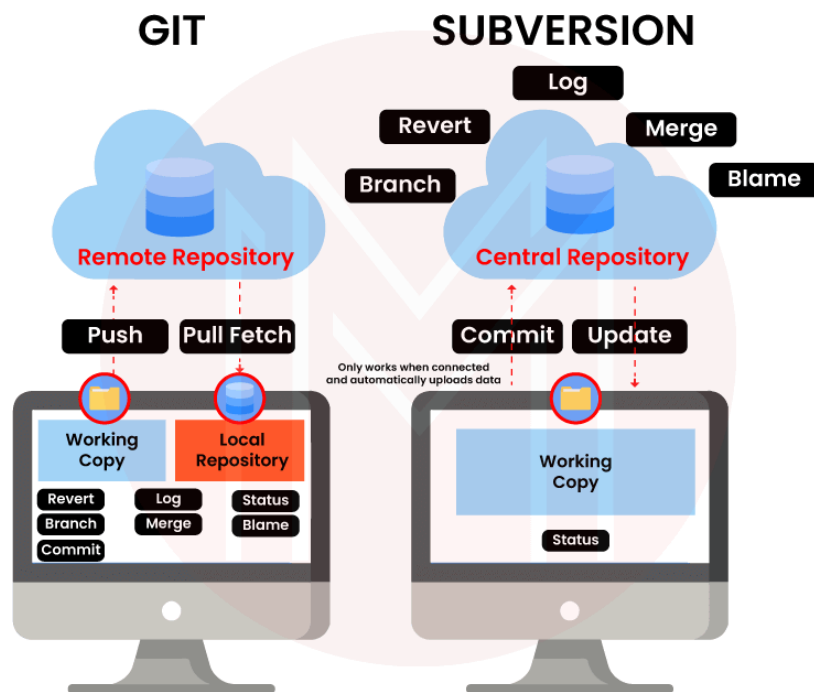


Рисунок 1.5 – Система контролю версій Git та SVN

3. **Бібліотеки та фреймворки:** Бот запускається сам, надсилає довідкові повідомлення, автентифікує користувачів, шукає ресурси на основі місцезнаходження користувача та зіставляє користувачів із доступними ресурсами на основі місцезнаходження та потреб за допомогою API Telegram і бібліотек Python, таких як telegram і запити. Бібліотека запитів використовується для надсилання запитів HTTP до зовнішніх API, таких як Google Places API, тоді як бібліотека telegram використовується для створення об'єкта Bot і надсилання повідомлень користувачеві. Для обробки різних типів користувацьких даних, включаючи інструкції, повідомлення та розташування, бот використовує різноманітні обробники, зокрема CommandHandler, MessageHandler і Filters. [10]

4. **Інструменти тестування:** інструменти тестування, такі як JUnit, NUnit і TestNG, можна використовувати для автоматизації тестування та забезпечення правильного функціонування програмного забезпечення.

5. **Інструменти збірки.** Інструменти збірки, такі як Maven, Gradle і Ant, автоматизують процес збирання, спрощуючи упаковку програмного забезпечення та розгортання його у виробництві.

6. **Інструменти документації:** такі інструменти, як Doxygen і Javadoc, можна використовувати для автоматичного створення документації для програмного забезпечення, що полегшує розробникам розуміння того, як використовувати програмне забезпечення та як воно працює.

Загалом кажучи, створення утиліти розділювача файлів може бути суттєво покращено за допомогою допоміжного програмного забезпечення та інструментів розробки. Ці технології можуть пришвидшити розробку, покращити якість коду та полегшити спільну роботу розробників.

					ІАЛЦ 467200.003 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

1.5 Огляд існуючих рішень

Це – груповий чат для обговорення тем, пов'язаних із підтвердженням особи, наприклад підтвердження ідентифікаційного документа, процедури КУС (знай свого клієнта) та запобігання шахрайству. Чат є публічним, тобто будь-хто може приєднатися та взяти участь у обговореннях.

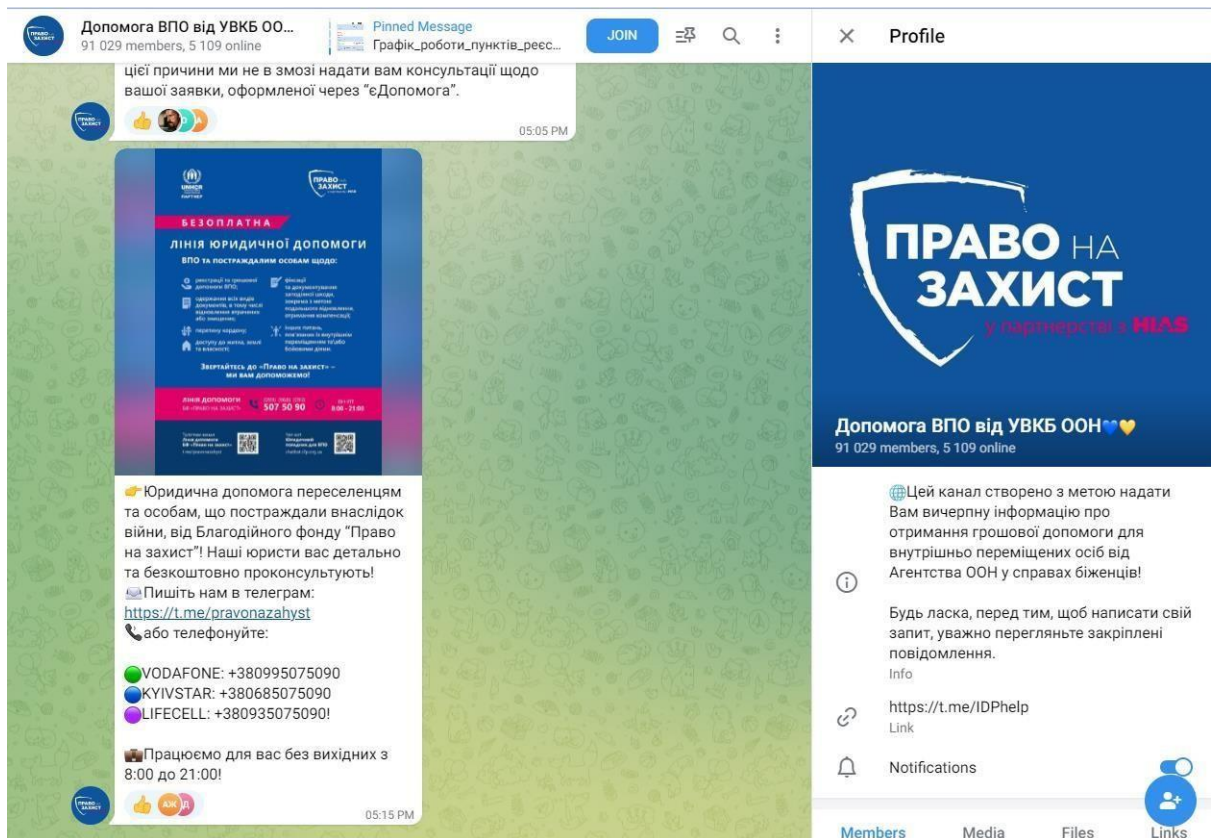


Рисунок 1.6 – Чат-бот допомоги ВПО

Чат-бот Telegram для підтримки ВПО (Програма міжнародного розвитку), надає допомогу біженцям та шукачам притулку. Бот пропонує різні послуги, такі як відповіді на поширені запитання, надання юридичної допомоги, зв'язок із волонтерами тощо. Користувачі можуть взаємодіяти з ботом, надсилаючи повідомлення та вибираючи параметри в меню, наданому ботом.

Чат-бот доступний кількома мовами, щоб задовольнити ширшу аудиторію. Він призначений для надання швидкої та ефективної підтримки

					ІАЛЦ 467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

тим, хто її потребує, особливо в ситуаціях, коли доступ до інформації та допомоги обмежений.

Група Telegram «Humanitarna» - група гуманітарної допомоги, зосереджена на наданні підтримки потерпілим від війни в Україні.

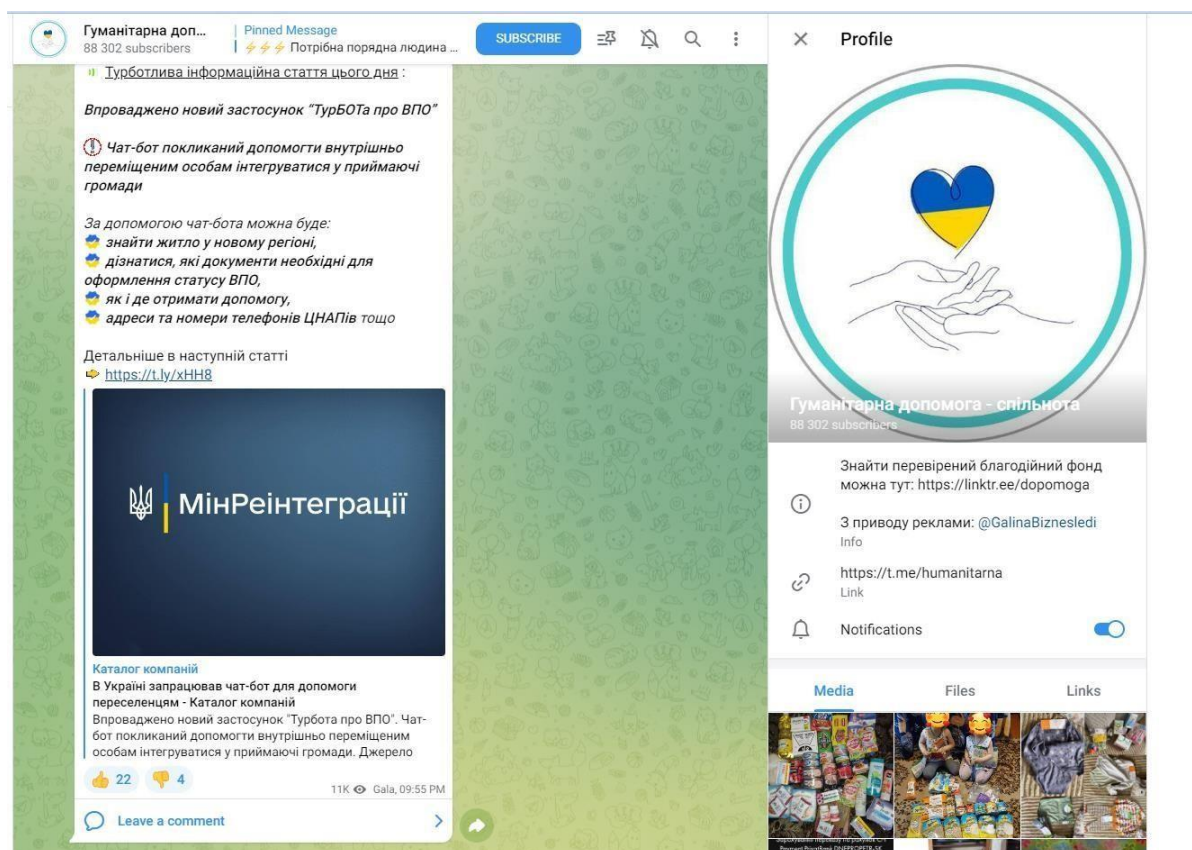


Рисунок 1.7 – Чат гуманітарної допомоги

Функції цієї групи можуть включати:

1. Збір і розподіл пожертвувань на підтримку незахищених груп, таких як біженці, діти та люди з обмеженими можливостями.
2. Надання інформації та ресурсів, щоб допомогти людям отримати доступ до критично важливих послуг, таких як охорона здоров'я та соціальні послуги.
3. Підвищення обізнаності про виклики та проблеми, з якими стикаються потерпілі від війни в Україні, та адвокація політики та дій, спрямованих на вирішення цих проблем.

4. Координація зусиль волонтерів для надання безпосередньої допомоги людям, які її потребують, наприклад, шляхом організації програм розподілу їжі, надання освітньої підтримки або допомоги людям у судових процесах.
5. Створення мереж і партнерства з іншими організаціями та окремими особами, які працюють над підтримкою потерпілого населення в Україні, щоб максимізувати вплив і ефективність.

Група Telegram «Ukraine_refugees_help», присвячена наданню допомоги та підтримки біженцям, які постраждали від конфлікту в Україні.

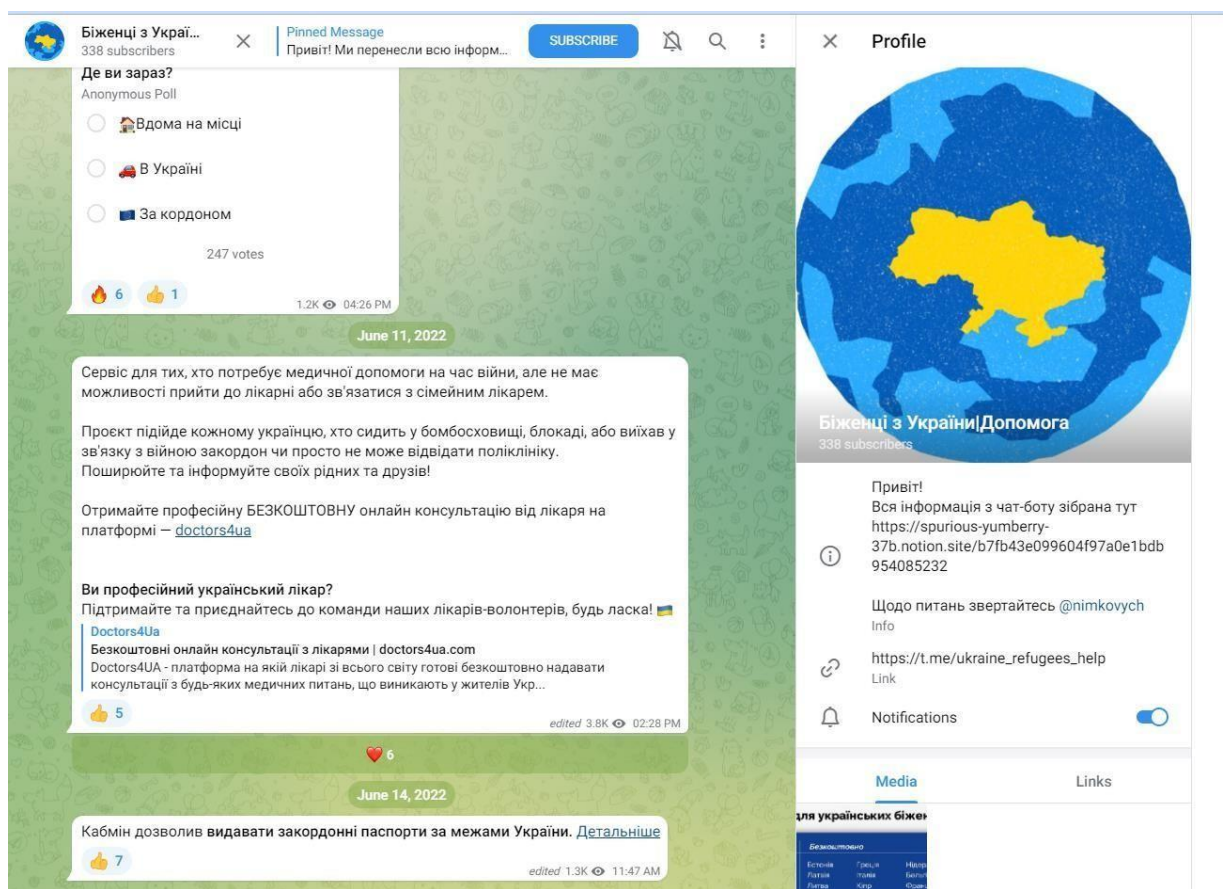


Рисунок 1.8 – Чат-бот допомоги біженцям

Функції цієї групи включають:

- Обмін інформацією та ресурсами, пов'язаними з потребами біженців, такими як інформація про законні права, варіанти житла та ресурси охорони здоров'я.

- Надання місця для біженців, щоб спілкуватися один з одним і ділитися своїм досвідом, а також ставити запитання та отримувати підтримку та поради від інших членів групи.
- Координація зусиль волонтерів і пожертвувань для підтримки біженців, наприклад шляхом збору та розподілу їжі, одягу та інших необхідних речей.
- Адвокація прав і потреб біженців в Україні та підвищення обізнаності про виклики та проблеми, з якими стикається ця вразлива груп населення.

Telegram-група «Укр_допомога» виглядає як спільнота людей, які займаються наданням допомоги та підтримки людям, які постраждали від конфлікту в Україні.

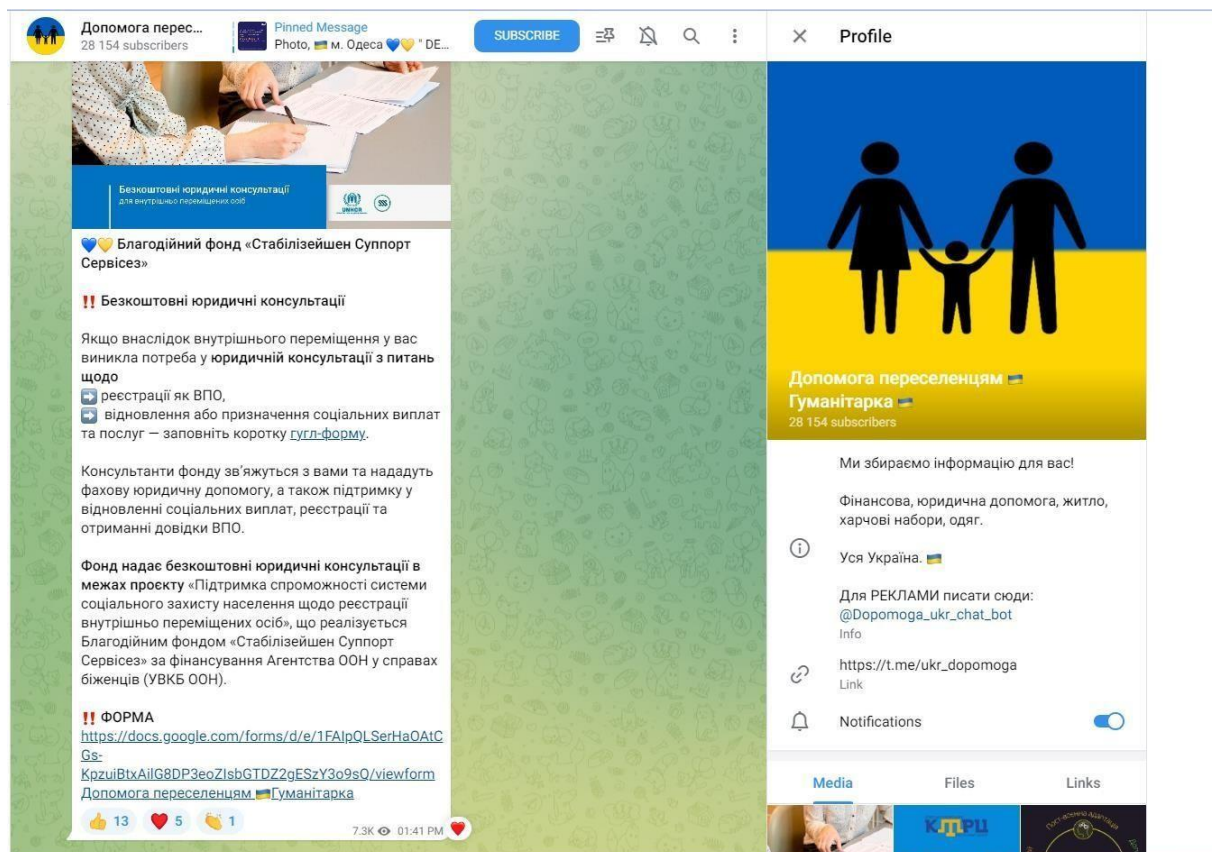


Рисунок 1.9 – Чат-бот допомоги біженцям

Функції цієї групи можуть включати:

1. Збір і розповсюдження пожертвувань на підтримку постраждалих від конфлікту людей, таких як біженці, сім'ї солдатів, поранених або переміщених осіб.
2. Надання інформації та ресурсів, щоб допомогти людям отримати доступ до критично важливих послуг, таких як охорона здоров'я, житло та соціальні послуги.
3. Координація зусиль волонтерів для надання безпосередньої допомоги людям, які її потребують, наприклад, шляхом організації програм розподілу їжі, надання освітньої підтримки або допомоги людям у судових процесах.
4. Підвищення обізнаності про виклики та проблеми, з якими стикаються люди, які постраждали від конфлікту в Україні, та адвокація політики та дій, спрямованих на вирішення цих проблем.
5. Розбудова мереж і партнерства з іншими організаціями та окремими особами, які працюють на підтримку людей, які постраждали від конфлікту в Україні, з метою максимізації впливу та ефективності.

					ІАЛЦ 467200.003 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

1.6 Аналіз вимог до програмного забезпечення

1.6.1 Функціональні вимоги

Функціональні вимоги — це особливі навички або функції, якими повинна володіти система чи частина програмного забезпечення, щоб виконувати завдання, для яких вони були розроблені. Ці специфікації окреслюють те, що система повинна досягти і як вона повинна діяти у відповідь на вхідні дані від користувачів або інші фактори. [12]

Функціональні вимоги:

- Автентифікація користувача: чат-бот повинен вимагати від користувачів автентифікації в телеграмі для доступу до його функцій.
- Пошук ресурсів: чат-бот має дозволяти користувачам шукати такі ресурси, як притулок, гуманітарна допомога та можливості для волонтерства.
- Зіставлення ресурсів: чат-бот повинен мати можливість зіставляти користувачів із доступними ресурсами на основі їхнього розташування та потреб.
- Допомога користувачам: чат-бот повинен мати можливість надавати допомогу користувачам, які потребують допомоги у використанні системи.

1.6.2 Нефункціональні вимоги

Нефункціональні вимоги — це обмеження або критерії, які описують, як система повинна поводитися або працювати, а не те, що вона повинна робити. Ці вимоги зосереджені на таких аспектах, як продуктивність, зручність використання, безпека та ремонтпридатність. [14]

Нефункціональні вимоги:

- Надійність: чат-бот має бути доступним 24/7 і мати мінімальний час простою, щоб гарантувати, що користувачі можуть отримати доступ до його функціональних можливостей у будь-який час.
- Безпека: чат-бот має бути безпечним, щоб захистити дані користувача та запобігти несанкціонованому доступу.
- Продуктивність: чат-бот повинен мати швидкий час відгуку та здатність обробляти велику кількість запитів.
- Масштабованість: чат-бот повинен бути розроблений для роботи зі зростаючою кількістю користувачів і ресурсів.
- Зручність використання: чат-бот має бути зручним і простим у навігації навіть для користувачів, які не розбираються в техніці.

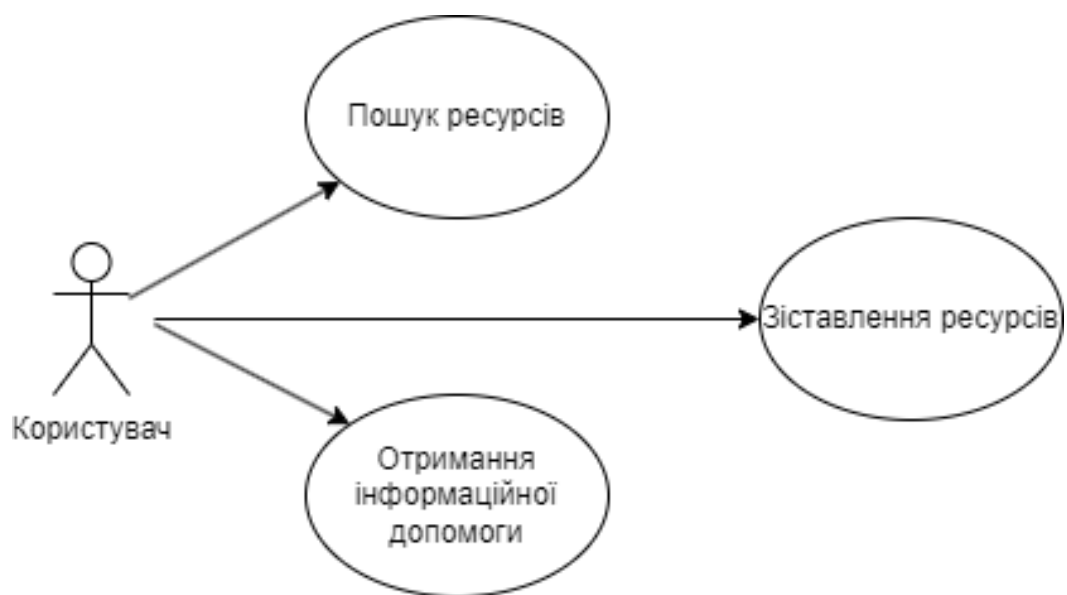


Рисунок 1.10 – Діаграма варіантів використання для користувача

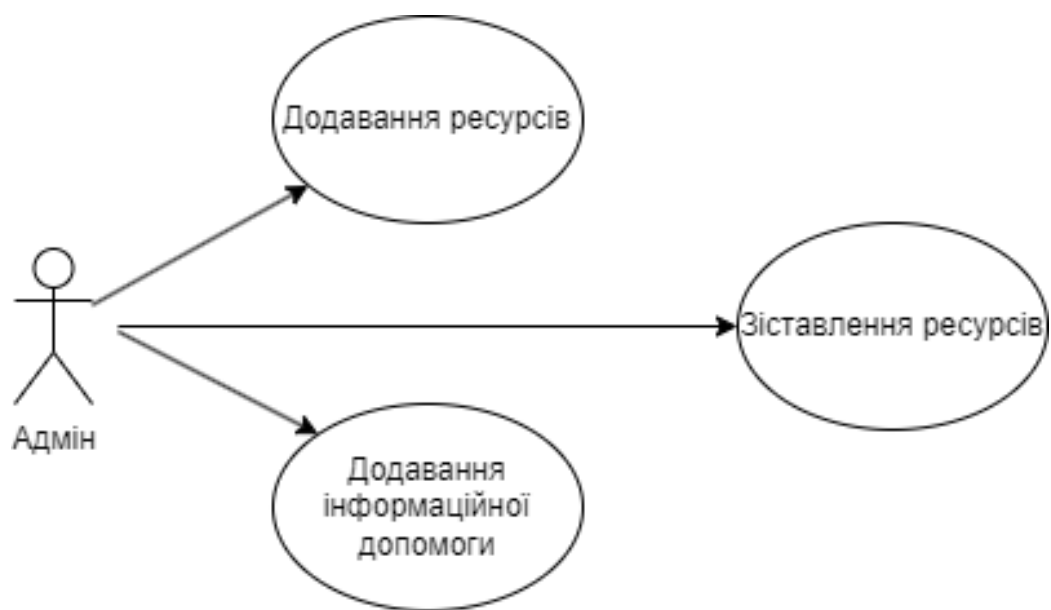


Рисунок 1.11 – Діаграма варіантів використання для адміністратора

Висновок до розділу 1

Загалом було описано сфери застосування чат бота для допомоги, описано концепцію чат ботів у контексті Telegram та описано існуючі рішення. Було розроблені функціональні та нефункціональні вимоги допоможуть переконатися, що чат-бот для пошуку допомоги є надійним, безпечним і зручним для користувача, надаючи критичні ресурси людям, які постраждали від криз, таких як війна в Україні. Також було розглянуто варіанти використання чат бота допомоги.

					ІАЛЦ 467200.003 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вибір мови програмування

Python — це популярна мова програмування для створення чат-ботів у Telegram через її читабельність, простоту та велику кількість інструментів і фреймворків, які полегшують створення та розгортання чат-ботів. [10] Оскільки Python є інтерпретованою мовою, код можна запускати негайно без необхідності окремого етапу компіляції, що робить його ідеальним для швидкого створення прототипів і розробки. Крім того, Python має значну та активну спільноту розробників, яка пропонує багато інформації та допомоги програмістам, які використовують цю мову.

Розробники можуть використовувати Telegram's Bot API для створення чат-ботів Python. Розробники можуть надсилати й отримувати повідомлення, контролювати взаємодію користувачів і отримувати доступ до інших можливостей платформи Telegram через API Telegram Bot. Використовуючи лише кілька рядків коду, ви можете створити робочого чат-бота за допомогою бібліотек Python, таких як `python-telegram-bot` і `telepot`, які пропонують прості оболонки через API Telegram Bot. [10]

Кілька модулів обробки природної мови (NLP) у Python, як-от `sraCy`, `NLTK` і `TextBlob`, спрощують розуміння та відповідь на введені користувачем дані більш розмовною манерою. Ці бібліотеки включають методи аналізу настроїв, токенизації та синтаксичного аналізу, які можна використовувати для інтерпретації введених користувачем даних і отримання відповідних відповідей. [10]

Загалом простота використання, гнучкість і потужна екосистема бібліотек роблять Python чудовим вибором для розробки чат-ботів у Telegram. Його багатий набір функцій і потужні можливості NLP

дозволяють створювати складні та захоплюючі чат-боти, які можуть надавати персоналізовану допомогу та підтримку користувачам.

2.2 Архітектура програмного забезпечення

2.2.1 Telegram Bot API

Telegram Bot API — це набір API, який дозволяє розробникам створювати та взаємодіяти з ботами Telegram. Все більш популярний хмарний сервіс обміну повідомленнями під назвою Telegram робить сильний акцент на безпеці та анонімності. По суті, боти Telegram — це автоматизовані програми, які можуть спілкуватися з користувачами, відповідати на їхні повідомлення та виконувати різні дії відповідно до введених користувачами. [10]

Telegram Bot API надає розробникам широкий спектр функцій і функцій, як-от:

1. **Надсилання та отримання повідомлень:** розробники можуть використовувати API для надсилання та отримання тексту, зображень та інших типів повідомлень.
2. **Керування користувачами:** API дозволяє розробникам керувати користувачами, наприклад додавати або видаляти користувачів із групи чату.
3. **Спеціальні клавіатури:** розробники можуть створювати власні клавіатури для своїх ботів, що дозволяє користувачам легше з ними взаємодіяти.

4. **Вбудований режим:** вбудований режим дозволяє розробникам вставляти своїх ботів у розмову в чаті, дозволяючи користувачам взаємодіяти з ботом, перебуваючи в розмові.

5. **Веб-хуки:** API дозволяє розробникам налаштовувати веб-хуки, які можна використовувати для отримання оновлень про взаємодію користувача з ботом.

6. **Платежі:** API включає підтримку платежів, що дозволяє розробникам створювати ботів, які можуть приймати платежі за товари чи послуги.

Загалом, Telegram Bot API надає розробникам потужний набір інструментів для створення та керування ботами Telegram. Широкий набір функцій і можливостей робить його популярним вибором для розробників, які хочуть створювати ботів для різних цілей, включаючи підтримку клієнтів, доставку новин тощо.

2.2.2 Діаграма діяльності

Діаграма діяльності (діаграма активності) дозволяє відображати послідовності дій, що реалізовані у програмному забезпеченні. [14]

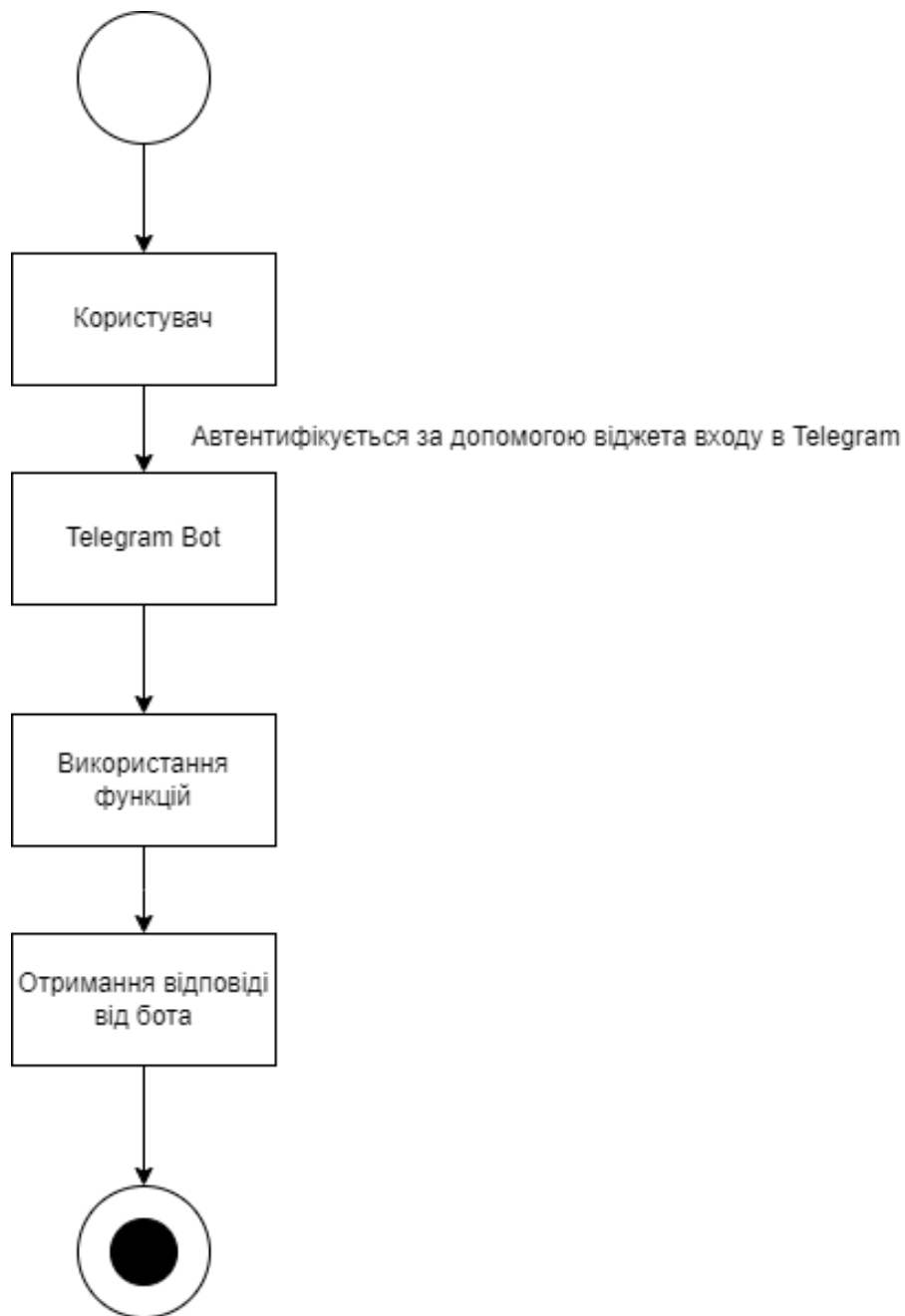


Рисунок 2.1 – Діаграма діяльності

Діаграма активності окреслює різні взаємодії, які відбуваються між користувачем, ботом Telegram, профілем користувача, API пошуку ресурсів, системою відповідності ресурсів і системою допомоги

користувачам у типовому сценарії використання. Діаграма показує потік інформації та дії між цими компонентами.

Спочатку користувач проходить автентифікацію за допомогою бота Telegram за допомогою віджета входу в Telegram, який генерує маркер автентифікації. Потім користувач може шукати ресурси, надіславши ключове слово боту за допомогою команди /search. Потім бот пропонує користувачеві надіслати своє місцезнаходження за допомогою віджета розташування Telegram.

API пошуку ресурсів отримує місцезнаходження та ключове слово користувача та повертає список ресурсів, які відповідають критеріям пошуку користувача. Потім ресурси надсилаються назад до профілю користувача, де вони можуть їх переглянути.

Потім користувач може використовувати команду /match, щоб зіставити себе з доступними ресурсами на основі свого розташування та потреб. Бот пропонує користувачеві надіслати своє місцезнаходження та тип необхідного ресурсу. Потім система зіставлення ресурсів зіставляє користувача з доступними ресурсами на основі його розташування та потреб. Відповідні ресурси потім надсилаються назад до профілю користувача.

Користувач також може скористатися командою /help, щоб отримати допомогу з використанням системи. Потім бот надає допомогу на основі запиту користувача.

Загалом діаграма активності надає візуальне представлення різних компонентів, які беруть участь у типовій взаємодії між користувачем і ботом Telegram із функціями автентифікації користувача, пошуку ресурсів, зіставлення ресурсів і допомоги користувачам.

2.2.3 Діаграма послідовності

Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. [14]

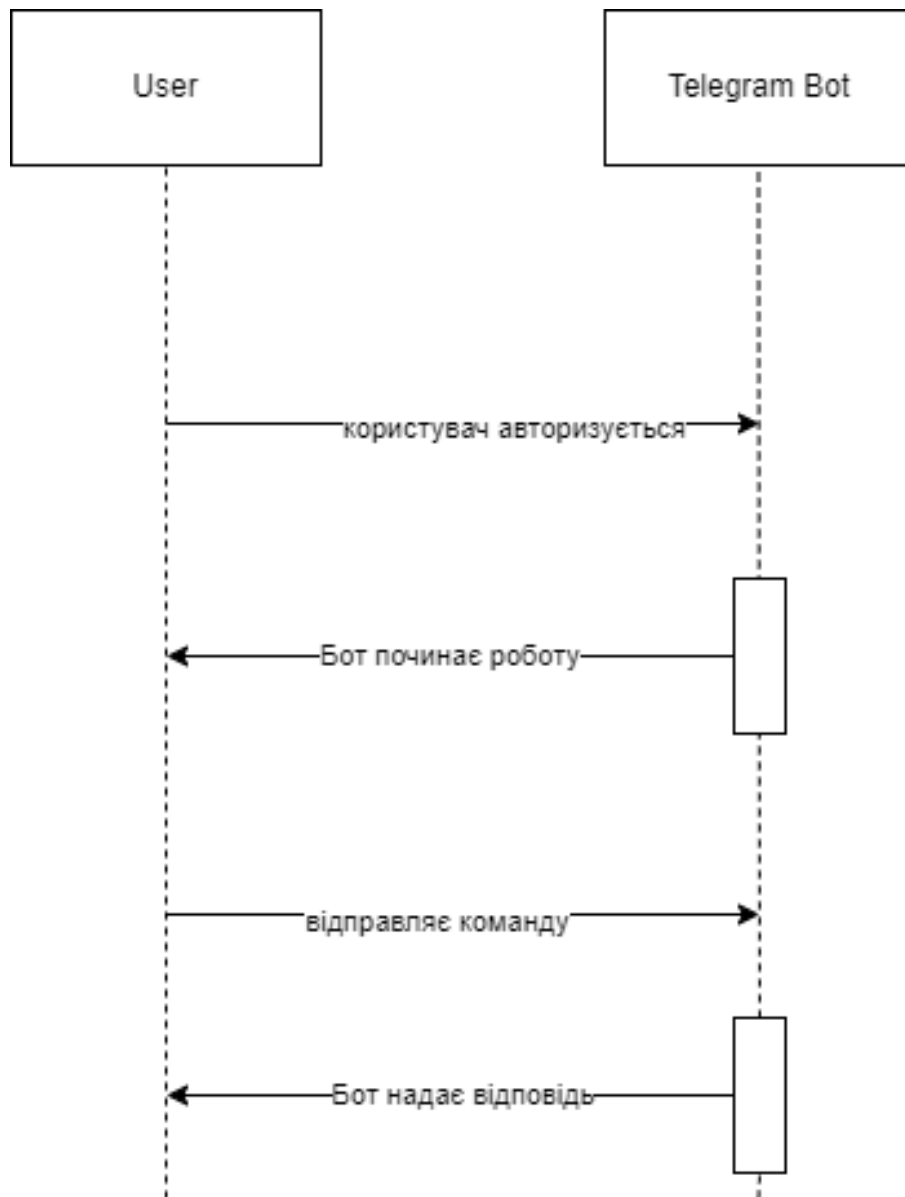


Рисунок 2.2 – Діаграма послідовності

Діаграма послідовності ілюструє етапи типового сценарію використання бота Telegram із функціями автентифікації користувача, пошуку ресурсів, зіставлення ресурсів і допомоги користувачам. На діаграмі показано взаємодію між користувачем та ботом Telegram та функціями бота.

Загалом діаграма послідовності показує, як користувач може взаємодіяти з ботом Telegram для пошуку ресурсів і отримання допомоги, а також як бот спілкується з профілем користувача, API пошуку ресурсів, системою відповідності ресурсів і системою допомоги користувачам для задоволення потреб користувача.

2.3 Опис функцій

Нижче описані функції, що забезпечують роботу бота для допомоги.

Метод	Опис
start(update, context)	Обробляє команду /start і відправляє вітальне повідомлення користувачу, пропонуючи вибрати категорію ресурсу.
search(update, context)	Обробляє команду /search і відправляє повідомлення користувачу, щоб вибрати категорію ресурсу.
handle_message(update, context)	Обробляє повідомлення від користувача і виконує відповідні дії залежно від отриманого тексту повідомлення.

В програмі використовуються кнопки для спрощення взаємодії з користувачем. Після виклику команди /start або /search, користувачу буде

надіслано повідомлення з кнопками категорій ресурсів: "Прихисток □", "Гуманітарна допомога □", "Волонтерство □" та "Запропонувати допомогу □".

При натисканні на одну з цих кнопок, відповідне повідомлення буде надіслано на сервер Telegram, де воно буде оброблено функцією `handle_message()`. У цій функції виконується перевірка тексту повідомлення користувача і відповідної реакції на кожен варіант.

Наприклад, якщо користувач натисне кнопку "Прихисток □", буде створено відповідне повідомлення зі списком прихистків, і воно буде надіслано користувачу. Те ж саме стосується інших кнопок: "Гуманітарна допомога □", "Волонтерство □" та "Запропонувати допомогу □".

У випадку, якщо користувач відправить повідомлення, яке не співпадає з жодним із запрограмованих варіантів, він отримає повідомлення "Я не розумію вашого запиту".

Таким чином, кнопки використовуються для спрощення навігації користувача та вибору категорій ресурсів у боті.

Загалом, цей код забезпечує базову структуру для створення чат-бота, який можна використовувати для пошуку та зіставлення користувачів із ресурсами на основі їхнього розташування та потреб.

2.3.1 Опис кнопок телеграм бота

ReplyKeyboardMarkup є класом з бібліотеки `python-telegram-bot`, який дозволяє створювати клавіатуру з кнопками для відповідей в Telegram боті. Він дозволяє налаштовувати розмір клавіатури, одноразове відображення кнопок та їх розміщення. [10]

Конструктор **ReplyKeyboardMarkup** приймає один обов'язковий параметр - двовимірний масив кнопок. Кожен внутрішній масив

представляє один рядок кнопок, а кожен елемент масиву – об'єкт `KeyboardButton`, який містить текст кнопки.

Наприклад, в рядку коду `ReplyKeyboardMarkup` ([[`KeyboardButton`('Прихисток □'), `KeyboardButton`('Гуманітарна допомога □')], [`KeyboardButton`('Волонтерство □'), `KeyboardButton`('Запропонувати допомогу □')]), `resize_keyboard=True`, `one_time_keyboard=True`) створюється клавіатура з двома рядками кнопок. Перший рядок містить кнопки "Прихисток □" та "Гуманітарна допомога □", а другий рядок містить кнопки "Волонтерство □" та "Запропонувати допомогу □".

Додаткові параметри, які можна вказати в конструкторі `ReplyKeyboardMarkup`, включають:

- `resize_keyboard`: визначає, чи має бути клавіатура автоматично змасштабована для відображення на всьому екрані. За замовчуванням встановлено значення `True`. [10]
- `one_time_keyboard`: визначає, чи має бути клавіатура прихована після натискання на одну з кнопок. За замовчуванням встановлено значення `False`. [10]

Після створення об'єкта `ReplyKeyboardMarkup` його можна передати як параметр у відповідний метод `send_message()` для надсилання повідомлення з клавіатурою в Telegram.

Висновок до розділу 2

Підсумовуючи, розробка телеграм бота для надання допомоги вимагає ретельного розгляду різних факторів, таких як вибір мови програмування, архітектури програмного забезпечення та додаткових інструментів. Python є підходящою мовою для цього проекту завдяки своїй простоті, гнучкості та підтримці великої бібліотеки. API Telegram є корисним інструментом для розробки взаємодії користувача та бота.

					ІАЛЦ 467200.003 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис процесів тестування

Тестування є важливим кроком у розробці програмного забезпечення, і воно допомагає переконатися, що програмне забезпечення працює належним чином і відповідає потребам користувача.

Ось кілька кроків, які можна виконати, щоб протестувати бота Telegram:

1. Модульне тестування: Модульне тестування передбачає перевірку окремих функцій або методів у коді, щоб переконатися, що вони працюють належним чином. Це можна зробити за допомогою тестових фреймворків, таких як Pytest або Unittest.
2. Інтеграційне тестування: інтеграційне тестування передбачає тестування взаємодії між різними компонентами бота. Наприклад, перевірити, чи обробник команд працює належним чином, коли він отримує дані від користувача.
3. Наскрізне тестування: наскрізне тестування передбачає тестування всієї системи, включаючи взаємодію між ботом і Telegram API. Це можна зробити за допомогою таких засобів, як Selenium або Cypress.
4. Тестування навантаження: тестування навантаження передбачає тестування продуктивності бота під високим навантаженням, щоб переконатися, що він може обробляти велику кількість користувачів і запитів.
5. Тестування прийнятності для користувачів: Тестування прийнятності для користувачів передбачає тестування бота реальними користувачами, щоб переконатися, що він відповідає їхнім потребам і функціонує належним чином.

Окрім цих типів тестування, також важливо переконатися, що бот є безпечним і що дані користувача захищені. Це може включати перевірку вразливостей і впровадження передових методів безпеки.

Загалом, тестування є важливою частиною процесу розробки програмного забезпечення, і важливо ретельно протестувати бота Telegram, щоб переконатися, що він відповідає потребам користувачів і функціонує належним чином.

В таблицях 1.2 - 1.4 наведені варіанти тестування програмного забезпечення.

Таблиця 3.1 – Варіант тестування UC-1

Use case name	Авторизація
Use case ID	UC-01
Goals	Автентифікація користувача
Actors	Гість (неzareєстрований користувач)
Trigger	Користувач хоче скористатись ботом
Pre-conditions	-
Flow of Events	Користувач переходить на сторінку бота. Для користування потрібно перейти в телеграм, щоб бути авторизованим і мати доступ до бота.
Extension	У випадку , якщо авторизація не пройдена, то доступу до боту не буде.
Post-Condition	Отримання доступу до бота.

Таблиця 3.2 – Варіант тестування UC-2

Use case name	Пошук ресурсів
Use case ID	UC-02
Goals	Пошук ресурсів
Actors	Користувач
Trigger	Користувач хоче знайти ресурси
Pre-conditions	Користувач запустив бота
Flow of Events	Користувач обирає функцію “пошук ресурсів”
Extension	Якщо потрібних ресурсів нема, бот видать повідомлення про помилку
Post-Condition	Отримання списку ресурсів

Таблиця 3.3 – Варіант тестування UC-3

Use case name	Зіставлення ресурсів
Use case ID	UC-03
Goals	Зіставлення ресурсів
Actors	Користувач
Trigger	Користувач хоче знайти доступні ресурси
Pre-conditions	Користувач запустив бота
Flow of Events	Користувач обирає функцію “пошук ресурсів”
Extension	Якщо доступних ресурсів нема, бот видає повідомлення про помилку
Post-Condition	Отримання доступних ресурсів

Таблиця 3.4 – Варіант використання UC-4

Use case name	Допомога
Use case ID	UC-04
Goals	Допомога користувачам
Actors	Користувач
Trigger	Користувач хоче знайти відповіді на питання
Pre-conditions	Користувач запустив бота
Flow of Events	Користувач обирає функцію FAQ
Extension	Якщо FAQ відсутні, користувач отримує повідомлення
Post-Condition	Отримання допомоги

3.2 Аналіз якості ПЗ

Аналіз якості програмного забезпечення (ПЗ) є важливою складовою розробки програмного продукту. Цей процес допомагає виявити потенційні проблеми, дефекти, а також оцінити загальну якість ПЗ перед його випуском.

Для проведення аналізу якості ПЗ можна використовувати різні методи та інструменти. Ось декілька загальних кроків, які можна виконати під час аналізу якості ПЗ:

1. Визначення вимог: Першим кроком є розуміння вимог до програмного продукту. Це допоможе визначити, які аспекти якості ПЗ є найважливішими для конкретного проекту.
2. Виявлення дефектів: Застосовуються різні методи для виявлення дефектів, такі як ручне тестування, автоматизоване тестування, валідація вхідних даних, перевірка на забезпечення безпеки тощо. Ці методи допомагають виявити проблеми, помилки та неполадки в роботі ПЗ.
3. Вимірювання якості: Використовуються різні метрики для вимірювання якості ПЗ, такі як частота виявлення дефектів, час відновлення, час реакції, продуктивність, надійність тощо. Ці метрики допомагають оцінити якість ПЗ з різних аспектів.
4. Аналіз результатів: Після проведення тестування та вимірювання якості необхідно проаналізувати отримані результати. Це дозволить виявити проблемні області та прийняти відповідні заходи для поліпшення якості ПЗ.

5. План дій: На основі аналізу результатів можна розробити план дій для виправлення виявлених проблем та поліпшення якості ПЗ. Цей план може включати в себе виправлення дефектів, оптимізацію процесів, оновлення документації тощо.
6. Повторення процесу: Аналіз якості ПЗ - це ітеративний процес. Після впровадження виправлень та поліпшень слід повторити процес аналізу якості, щоб перевірити, чи були досягнуті поставлені цілі та якість ПЗ вдосконалилась.

У додаток до вищевказаних кроків, можуть використовуватись спеціалізовані методи інспекції коду, аудит безпеки, проведення аналізу витоку пам'яті, профілінг продуктивності та інші підходи в залежності від потреб проекту та вимог до якості ПЗ.

3.3 Опис контрольного прикладу

Для роботи з телеграм ботом нам потрібно запустити його.

					ІАЛЦ 467200.003 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

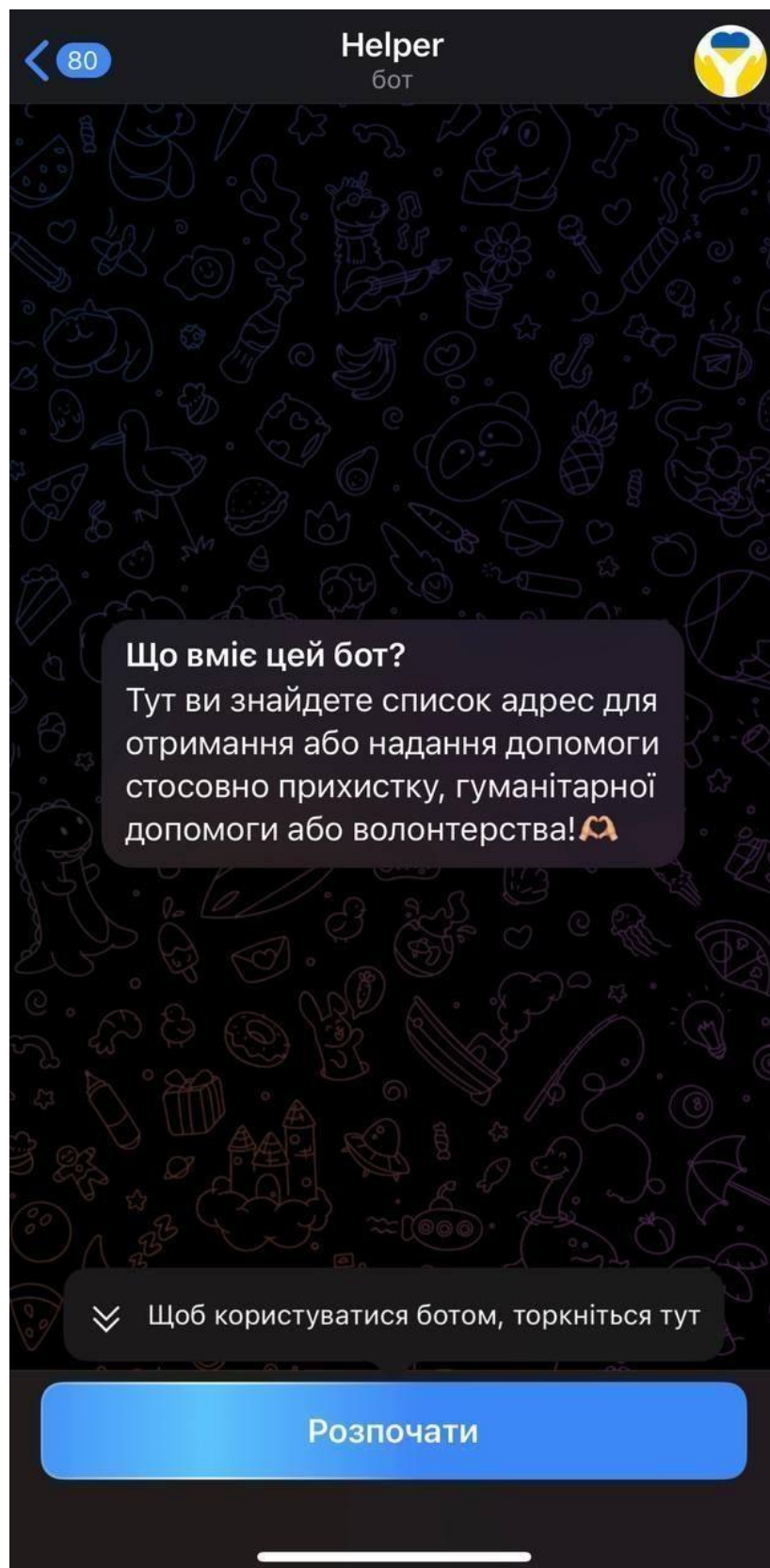


Рисунок 3.1 – Запуск бота

Після запуску отримуємо повідомлення від бота.

					ІАЛЦ 467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

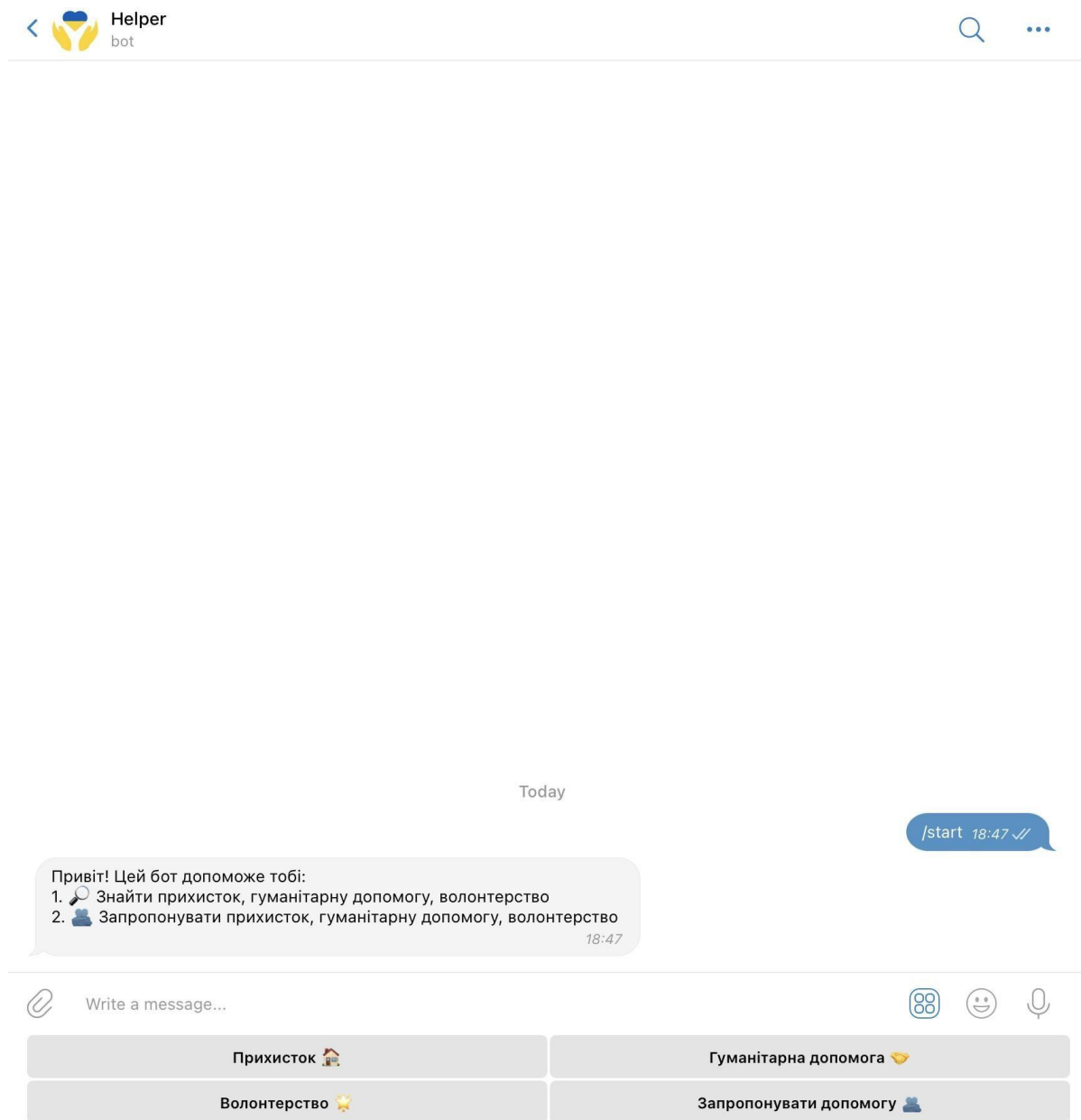


Рисунок 3.2 – Функціонал бота

Далі обираємо що саме нам потрібно. Наприклад, “Прихисток”.

					ІАЛЦ 467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

Today

/start 18:47 ✓✓

Привіт! Цей бот допоможе тобі:

1. 📍 Знайти прихисток, гуманітарну допомогу, волонтерство
2. 🗣️ Запропонувати прихисток, гуманітарну допомогу, волонтерство

18:47

Прихисток 🏠 18:48 ✓✓

🏠 Список адрес куди можна прийти за прихистком:

📍 м. Київ, вул. Металістів, 8.
Контактний номер: 📞 +380987630200

📍 м. Львів, вул. Городоцька, 128.
Контактний номер: 📞 +380987630300

📍 м. Харків, вул. Острозька, 34.
Контактний номер: 📞 +380686632300

📍 м. Суми, вул. Вербицького, 11.
Контактний номер: 📞 +380686632300

18:48



Write a message...



Рисунок 3.3 – Функціонал бота

Як бачимо при натисканні на кнопку, отримуємо список актуальних адрес та контактних номерів, куди можна звернутись за отриманням прихистку. Наступним розглянемо “Волонтерство”.

					ІАЛЦ 467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

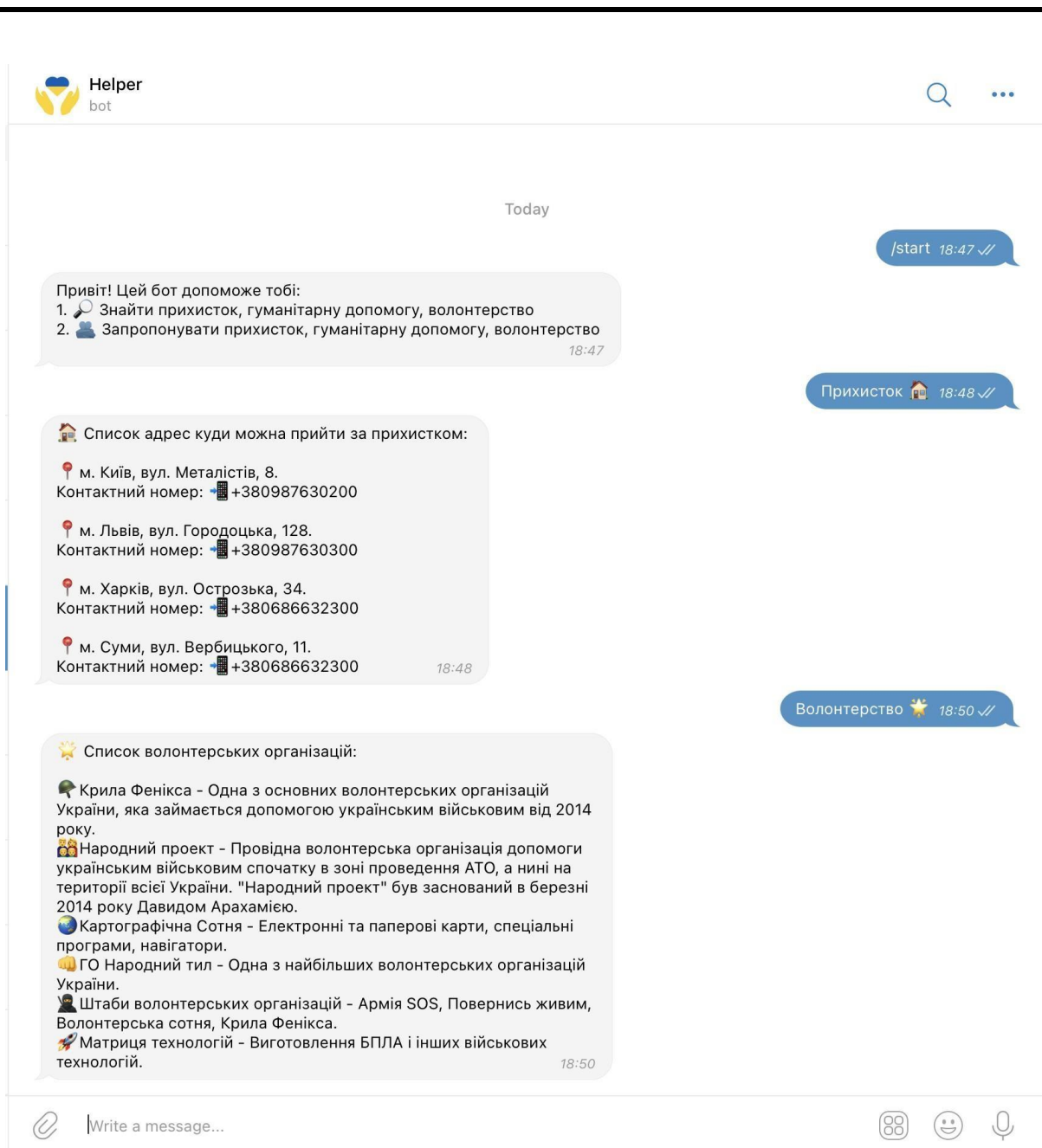


Рисунок 3.4 – Функціонал бота

Наступним розглянемо “Гуманітарна допомога”.

					ІАЛЦ 467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

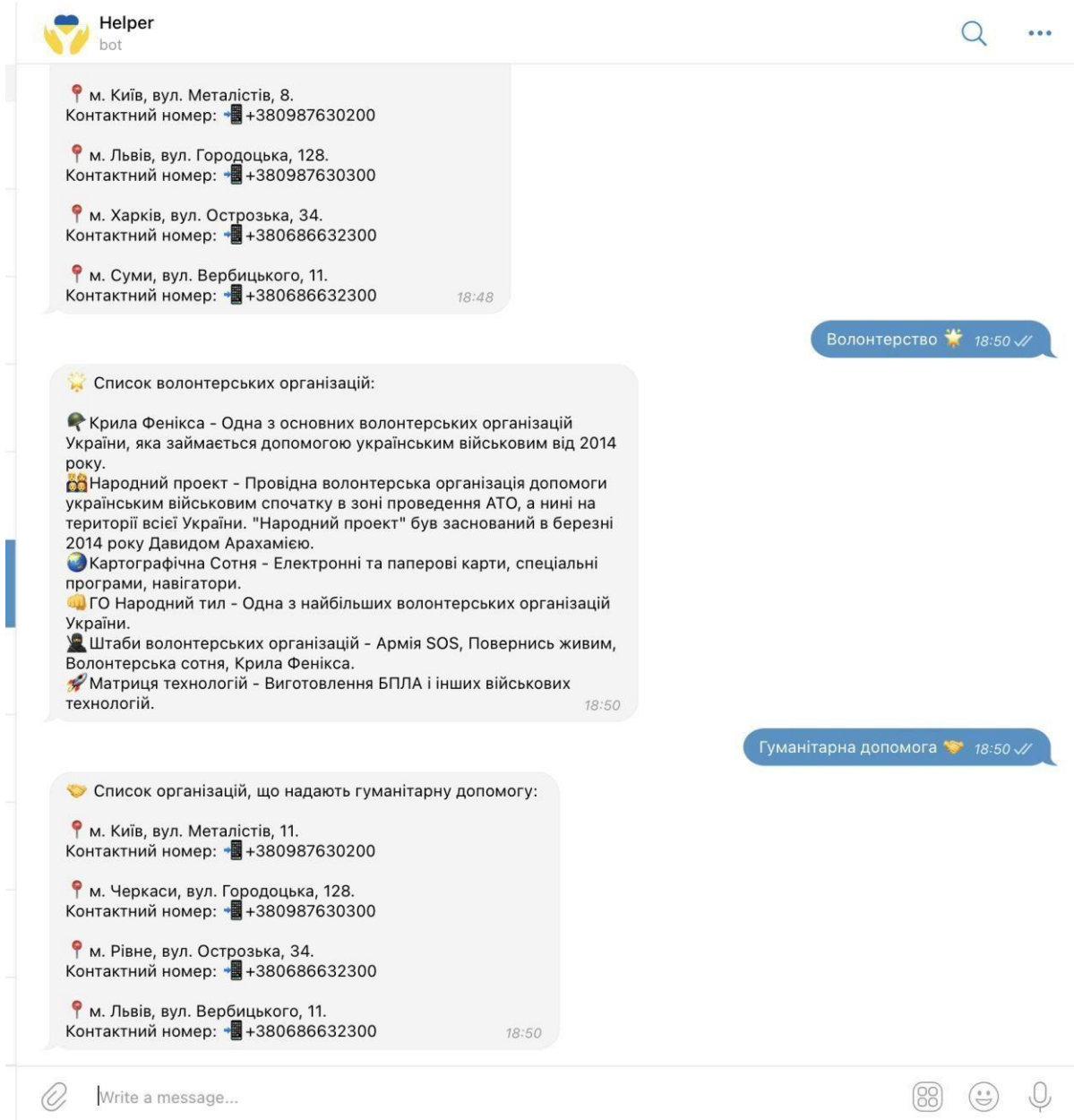


Рисунок 3.5 – Функціонал бота

Як бачимо при натисканні на кнопку, отримуємо список актуальних адрес та контактних номерів, куди можна звернутись за отриманням гуманітарної допомоги.

Наступним розглянемо “Запропонувати допомогу”.

					ІАЛЦ 467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

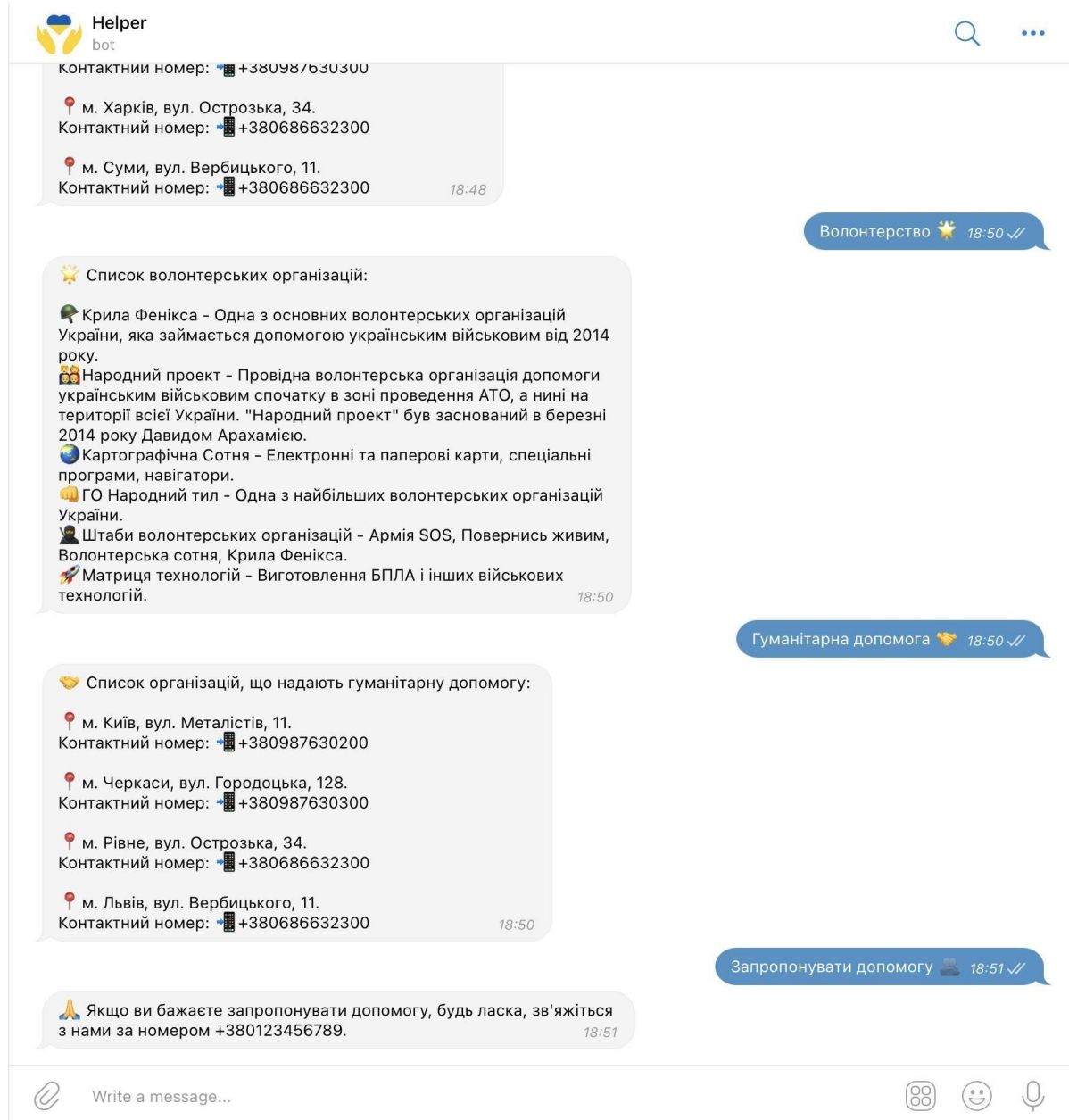


Рисунок 3.6 – Функціонал бота

Як бачимо при натисканні на кнопку, отримуємо контактний номер, куди можна звернутись за інформацією про надання допомоги.

					ІАЛЦ 467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

Висновки до 3 розділу

Підсумовуючи, тестування та аналіз якості програмного забезпечення є важливими частинами процесу розробки програмного забезпечення, щоб гарантувати, що програмний продукт або система відповідає потребам і очікуванням користувачів. Тестування може виконуватися на різних рівнях, таких як модульне тестування, інтеграційне тестування, наскрізне тестування, тестування навантаження та тестування прийнятності користувача, щоб перевірити функціональність, продуктивність, безпеку та зручність використання програмного продукту.

Аналіз якості програмного забезпечення передбачає оцінку та вимірювання різних аспектів програмного забезпечення, щоб визначити сфери, які потрібно вдосконалити, і переконатися, що програмне забезпечення є безпечним, надійним, ефективним, зручним для обслуговування та відповідає вимогам користувачів. Такі показники, як щільність дефектів, покриття коду, тестове покриття, показники продуктивності, і задоволеність користувачів можна використовувати для оцінки ефективності процесів тестування та розробки та прийняття обґрунтованих рішень щодо покращення якості програмного забезпечення.

Виконуючи ретельне тестування та аналіз якості програмного забезпечення, розробники програмного забезпечення можуть створювати високоякісні програмні продукти, які відповідають потребам користувачів і створюють комфортне середовище для роботи з ними.

РОЗДІЛ 4. ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Щоб розгорнути цього бота Telegram, потрібно виконати такі дії [15]:

1. Розмістити сценарій бота: розмістити сценарій бота можна в хмарній службі, як-от Heroku, Amazon Web Services (AWS), Google Cloud Platform або Microsoft Azure. Крім того, ви можете розмістити його на віртуальному приватному сервері (VPS) від хостинг-провайдера, наприклад DigitalOcean або Linode.
2. Установити необхідні залежності: включаючи бібліотеку API Telegram і бібліотеку запитів.
3. Налаштувати обліковий запис бота Telegram. Потрібно створити обліковий запис бота Telegram і отримати токен. Цей маркер використовуватиметься для автентифікації вашого бота на серверах Telegram.
4. Налаштувати сценарій бота: у сценарії замінити текст-заповнювач "YOUR_TELEGRAM_BOT_TOKEN" на ваш фактичний маркер бота Telegram. Вам також потрібно буде замінити URL-адреси-заповнювачі на фактичні URL-адреси пошуку ресурсів і відповідних API.
5. Запустити сценарій бота: коли сценарій налаштовано, його можна запустити за допомогою команди на зразок `python bot.py`. Переконайтеся, що сценарій працює постійно, щоб ваш бот міг відповідати на повідомлення в будь-який час.
6. Перевірка бота: коли бот запущений, його можна протестувати, надіславши йому повідомлення в Telegram. Наприклад, надіслати такі команди, як `/start`, `/help`, `/search` і `/match`, щоб перевірити, чи правильно відповідає бот.

					ІАЛЦ 467200.003 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

7. Підтримка: після того, як бот розгорнуто, треба відстежувати його на наявність помилок і робити будь-які необхідні оновлення або виправляти помилки.

Розглянемо розгортання бота на платформі heroku.

Розгортання бота Telegram на Heroku передбачає наступні кроки [15]:

- Створення облікового запису Heroku: робимо обліковий запис Heroku на <https://signup.heroku.com>.
- Встановлення Heroku CLI: встановлюємо Heroku Command Line Interface (CLI) на комп'ютер. Це дозволить нам взаємодіяти з Heroku з командного рядка.
- Створення нової програми Heroku: створюємо нову програму на Heroku, виконавши таку команду у своєму терміналі: `heroku create <TelegramBot>`. Це створить нову програму на Heroku та додасть новий віддалений репозиторій до вашої конфігурації git.
- Налаштування змінних середовища: установимо ТОКЕН бота Telegram та будь-які інші змінні середовища, необхідні для запуску бота. Встановити змінні середовища на Heroku можна за допомогою Heroku CLI: `heroku config:set TELEGRAM_TOKEN=<your-telegram-bot-token>`.
- Відправка коду у Heroku: виконавши таку команду у своєму терміналі: `git push heroku master`.
- Масштабування програми: масштабувати програму до одного dyno (веб-процесу), виконавши таку команду в терміналі: `heroku ps:scale web=1`.

Сервер запустить бота та переконається, що вона працює принаймні на одному динаміку.

					ІАЛЦ 467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

- Перевірка роботи: перевіримо, чи бот запущений на Heroku, перейшовши на URL-адресу, надану Heroku, у терміналі або виконавши таку команду: `heroku open`
- Тестування: перевірте свого бота Telegram, надіславши повідомлення боту в Telegram. Якщо все налаштовано правильно, ми повинні отримати відповідь від бота.

4.2 Підтримка програмного забезпечення

Підтримка програмного забезпечення для бота Telegram передбачає забезпечення постійного обслуговування та усунення несправностей, щоб переконатися, що бот функціонує належним чином і відповідає потребам своїх користувачів. Це включає в себе виправлення помилок і вирішення проблем, які виникають під час використання, а також оновлення та вдосконалення бота з часом.

Є кілька ключових сфер підтримки, які можуть знадобитися для бота Telegram, зокрема:

1. Технічна підтримка: це передбачає вирішення будь-яких технічних проблем, які виникають із ботом, наприклад, проблеми з підключенням, помилки сервера або проблеми з інтеграцією API.
2. Підтримка користувачів: це передбачає вирішення будь-яких проблем або запитань, які виникають у користувачів бота, наприклад, як використовувати певні функції або як усунути проблеми з ботом.
3. Технічне обслуговування та оновлення: це передбачає регулярний моніторинг і підтримку бота, щоб переконатися, що він функціонує належним чином, а також оновлення та вдосконалення бота з часом.

Щоб забезпечити ефективну підтримку програмного забезпечення для бота Telegram, важливо мати команду, яка має досвід розробки ботів, а також досвід роботи з конкретною мовою програмування та API, які

використовуються для створення бота. Крім того, наявність хороших навичок спілкування та співпраці може допомогти забезпечити швидке й ефективне вирішення запитів на підтримку.

Документація ResourceBot

Ласкаво просимо до ResourceBot, вашого особистого засобу пошуку ресурсів у Telegram! Ця документація допоможе вам використовувати ResourceBot і його функції.

Початок роботи

Щоб почати використовувати ResourceBot, просто знайдіть його в Telegram і натисніть кнопку «Пуск» або надішліть боту команду «/start». Це ініціалізує бота, і ви отримаєте вітальне повідомлення з інструкціями щодо автентифікації.

Автентифікація ResourceBot вимагає автентифікації для доступу до інформації про місцезнаходження. Щоб пройти автентифікацію, просто клацніть віджет входу, наданий ResourceBot, або надішліть команду "/authenticate". Це перенаправить вас на сторінку входу у веб-переглядачі, де ви зможете авторизуватися за допомогою облікових даних Telegram.

Пошук ресурсів

Для пошуку ресурсів надішліть команду "/search". ResourceBot запропонує вам поділитися своїм місцезнаходженням за допомогою віджета розташування, наданого Telegram. Натисніть кнопку «Поділитися місцем», щоб поділитися своїм місцем розташування. Після того, як ваше місцезнаходження буде надано, ResourceBot відобразить список доступних ресурсів на основі вашого місцезнаходження.

Зіставлення з ресурсами

Щоб зіставити з доступними ресурсами, надішліть команду "/match". ResourceBot запропонує вам поділитися своїм місцем розташування та бажаним ключовим словом. Поділіться своїм місцезнаходженням за допомогою віджета місцезнаходження, наданого Telegram, і вкажіть бажане

ключове слово в команді. ResourceBot підбере вам доступні ресурси на основі вашого місцезнаходження та бажаного ключового слова.

Часті запитання

Щоб переглянути список поширених запитань, надішліть команду `"/help"`. ResourceBot надасть вам список типових запитань і відповідей.

Ми сподіваємось, що ця документація була корисною у використанні ResourceBot.

Якщо у вас виникнуть додаткові запитання чи сумніви, зв'яжіться з нами за адресою test@icloud.com. Дякуємо за використання ResourceBot!

					ІАЛЦ 467200.003 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

Висновки до 4 розділу

В даному розділі було описано розгортання програмного забезпечення на платформі Heroku, хмарній платформі, яка дозволяє легко розгортати та масштабувати веб-додатки. Було виділено різні кроки, пов'язані з розгортанням бота, наприклад створення облікового запису Heroku, налаштування Heroku CLI, створення програми Heroku, надсилання коду в програму Heroku та налаштування змінних середовища. Також було описано способи підтримки програмного забезпечення.

Нарешті, було досліджено важливість підтримки програмного забезпечення для бота, яка передбачає забезпечення постійного обслуговування та оновлень, щоб гарантувати, що бот залишається функціональним і відповідає потребам своїх користувачів. Це включає моніторинг бота на наявність помилок, виправлення помилок і додавання нових функцій за потреби.

					ІАЛЦ 467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

ВИСНОВКИ

У цій дипломній роботі обговорювалися випадки використання чат-ботів, які надають допомогу, ідея чат-ботів у контексті Telegram та існуючі рішення. Щоб допомогти зробити чат-бот для пошуку допомоги надійним, безпечним і зручним для користувача, одночасно надаючи необхідні ресурси тим, хто постраждав від криз, таких як війна в Україні, були розроблені функціональні та нефункціональні критерії. Були й інші ідеї щодо використання допоміжного чат-бота.

Підсумовуючи, створення бота підтримки Telegram передбачає ретельний розгляд ряду змінних, включаючи вибір мови програмування, архітектуру програмного забезпечення та інші інструменти. Python є відповідною мовою для цього проекту через його легкість використання, адаптивність і широку підтримку бібліотек. Корисним ресурсом для створення взаємодії між користувачем і роботом є Telegram API.

Також було проведено тестування та аналіз якості програмного забезпечення, що є важливими частинами процесу розробки програмного забезпечення, щоб гарантувати, що програмний продукт або система відповідає потребам і очікуванням користувачів.

Також було описано процес розгортання програмного забезпечення за допомогою хмарної платформи Heroku, яка спрощує масштабування та публікацію веб-додатків. Було детально описано багато процедур, необхідних для розгортання бота, включаючи налаштування змінних середовища, створення облікового запису Heroku, налаштування Heroku CLI, створення програми Heroku та надсилання коду до програми Heroku. Також були включені параметри підтримки програмного забезпечення.

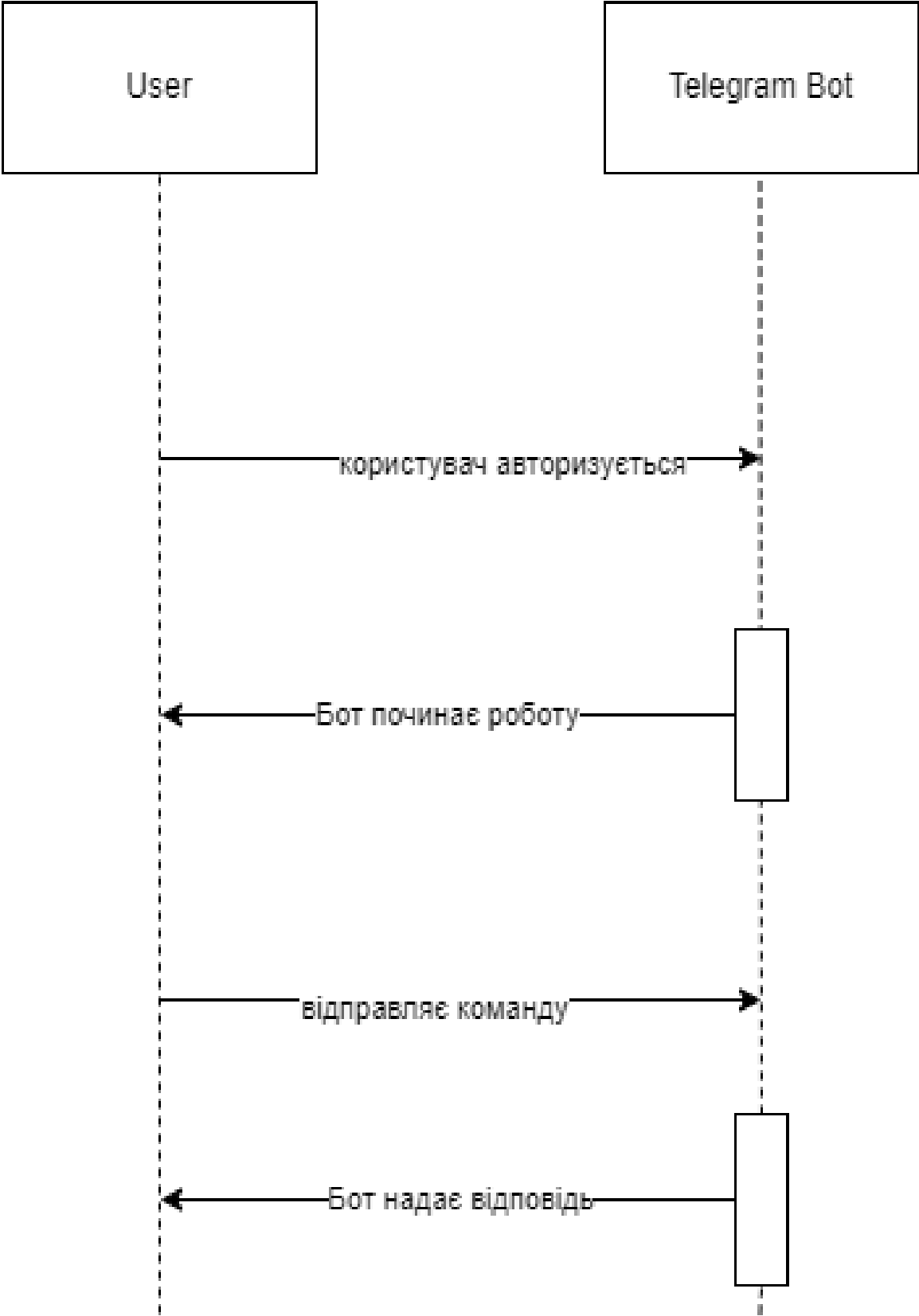
					ІАЛЦ 467200.003 ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Провотар О.І. Особливості та проблеми віртуального спілкування за допомогою чат-ботів / О.І. Провотар, Х.А. Ключко // Наукові праці ВНТУ: Інформаційні технології та комп'ютерна техніка. – 2013. – № 3. – 6 с.
2. Сухас У. Oracle Intelligent Bots: Чат-боты с искусственным интеллектом [Интернет ресурс]. – Oracle, 2017. – Режим доступа: <https://www.oracle.com/a/ocom/docs/chatbots.pdf> вільний.
3. Українець И. Системы аналитики для чат-ботов [Интернет-ресурс]. – Режим доступа: <http://ilyaukrainets.ru/chatbot/sistemy-analitiki-dlya-chat-botov/> вільний.
4. A curated directory of chat bot resources & tools [Интернет-ресурс]. – Режим доступа: <http://www.botsfloor.com/botstash/products/?category=Development%20Platforms> вільний.
5. Chatbot Report 2018: Current landscape of how people create chatbots and how users expect to interact with them [Интернет-ресурс]. – Режим доступа: <https://elearningindustry.com/chatbots-for-learning-support-10-reasonshttps://naiz.chat/NAIZ-report-18072018.pdf> вільний.
6. Messaging apps are now bigger than social networks [Интернет-ресурс]. – Режим доступа: <https://www.businessinsider.com/the-messaging-app-report-2015-11> вільний.
7. Sheth B. The Bot Lifecycle. What to know before you make your chatbot [Интернет ресурс]. – Режим доступа: <https://chatbotsmagazine.com/the-bot-lifecycle1ff357430db7> вільний.
8. Michiels. E. Modelling Chatbots with a Cognitive System Allows for a Differentiating User Experience [Интернет-ресурс]. – Режим доступа: <http://ceurws.org/Vol-2027/paper24.pdf> вільний.
9. Muldowney O. Chatbots. An Introduction And Easy Guide To Making Your Own / O. Muldowney. – Dublin: Curses & Magic, 2017. – 69 p.

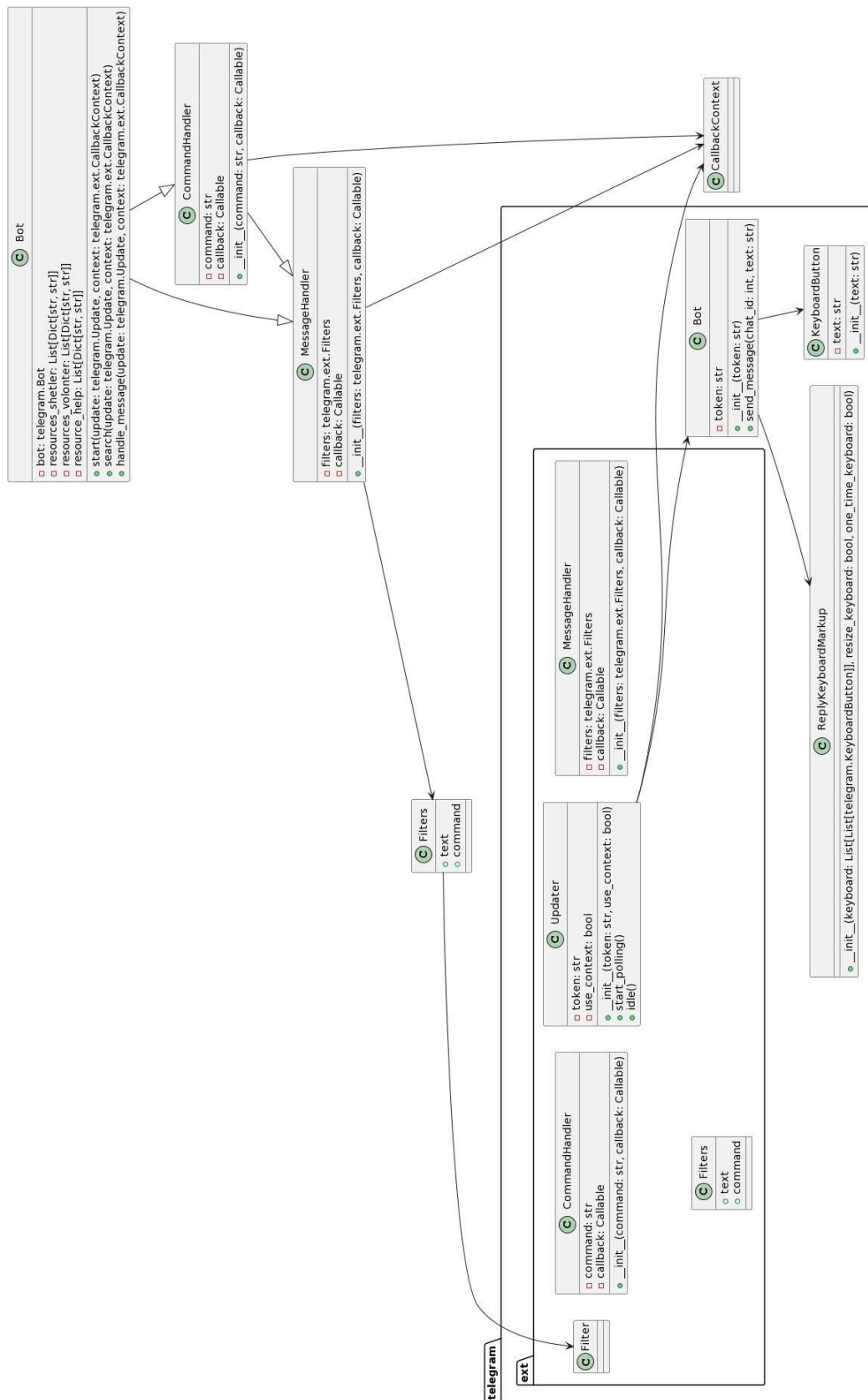
10. Telegram Bot API [Інтернет-ресурс]. – Режим доступу: <https://core.telegram.org/bots/api> вільний.
11. Ушакова Г. Д. Особенности виртуального общения посредством чатов / Г.Д. Ушакова, Ю.В. Балабанова // Филологический журнал : межвузовский сборник научных статей. – 2004. – Вып. XII. – С. 59 – 61.
11. Beaver L. The Chatbots Explainer [Інтернет-ресурс] / Web-сайт: BI Intelligence Copyright. – Режим доступу: <https://www.businessinsider.com/intelligence/chatbots-explainer> вільний.
12. Що таке функціональні вимоги: приклади, визначення, повний посібник [Електронний ресурс] – Режим доступу до ресурсу: <https://visuresolutions.com/uk/blog/functional-requirements/>.
13. Путівник мовою програмування Python [Електронний ресурс] – Режим доступу до ресурсу: <https://pythonguide.rozh2sch.org.ua/>.
14. Types of UML Diagrams [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lucidchart.com/blog/types-of-UML-diagrams>.
15. How to Deploy a Telegram Bot using Heroku for FREE [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/how-to-deploy-a-telegram-bot-using-heroku-for-free-9436f89575d2?gi=271cfb3577fb>.

Додаток 1
До дипломного проєкту
на тему: «Чат-бот для пошуку
необхідної інформації про
допомогу»



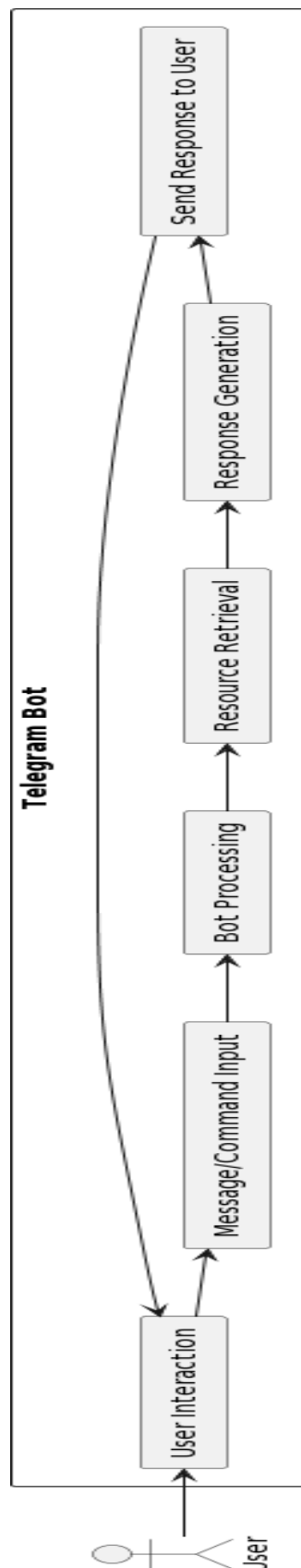
					ІАЛЦ 467200.004 Д1			
					Чат-бот для допомоги Блок-схема алгоритму дій (Діаграма послідовності)			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив		Мельник А. С.						
Перевірив								
Т. Контр.								
Н. Контр.					КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-93			
Затвердив								
					Лит.	Аркуш	Аркушів	
						1	1	

Додаток 2
До дипломного проєкту
на тему: «Чат-бот для пошуку
необхідної інформації про
допомогу»



					ІАЛЦ 467200.005 Д2				
Зм.	Арк.	№ докум.	Підпис	Дата	Чат-бот для допомоги Діаграма класів (Структурна схема)	Лит.	Аркуш	Аркушів	
Розробив		Мельник А. С.					1	1	
Перевірів									
Т. Контр.									
Н. Контр.									
Затвердив						КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-93			

Додаток 3
До дипломного проєкту
на тему: «Чат-бот для пошуку
необхідної інформації про
допомогу»



					ІАЛЦ 467200.006 ДЗ						
Зм.	Арк.	№ докум.	Підпис	Дата	Чат-бот для допомоги Діаграма взаємодії (Функціональна схема)			Лит.	Аркуш	Аркушів	
Розробив		Мельник А. С.								1	1
Перевірів											
Т. Контр.											
Н. Контр.											
Затвердив											
					КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-93						

Додаток 4
До дипломного проєкту
на тему: «Чат-бот для пошуку
необхідної інформації про
допомогу»

```

import telegram
from telegram import ReplyKeyboardMarkup, KeyboardButton
from telegram.ext import Updater, CommandHandler,
MessageHandler, Filters

# Telegram bot token
TOKEN = '5766786622:AAHqFxsQXLuimE84BmDxPzgyTipuNVyEK94'

# Create an instance of the Telegram bot
bot = telegram.Bot(token=TOKEN)

# Mockup data for available resources
resources_shetler = [
    {
        'name': 'Shelter A',
        'location': 'м. Київ, вул. Металістів, 8.',
        'category': 'Прихисток',
        'contact': '+380987630200'
    },
    {
        'name': 'Shelter B',
        'location': 'м. Львів, вул. Городоцька, 128.',
        'category': 'Прихисток',
        'contact': '+380987630300'
    },
    {
        'name': 'Shelter C',
        'location': 'м. Харків, вул. Острозька, 34.',
        'category': 'Прихисток',
        'contact': '+380686632300'
    }
]

```

					ІАЛЦ 467200.007 Д4						
Зм.	Арк.	№ докум.	Підпис	Дата	Чат-бот для пошуку необхідної інформації про допомогу Програмний код						
Розробив	Мельник А. С.										
Перевірив											
Т. Контр.											
Н. Контр.											
Затвердив					Лит. Аркуш Аркушів 1 6 КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-93						


```

},
{
    'name': 'Shelter D',
    'location': 'м. Суми, вул. Вербицького, 11.',
    'category': 'Прихисток',
    'contact': '+380686632300'
}
]

resources_volonter = [
    {
        'name': 'Крила Фенікса',
        'description': 'Одна з основних волонтерських
організацій України, яка займається допомогою українським
військовим від 2014 року.'
    },
    {
        'name': 'Народний проект',
        'description': 'Провідна волонтерська організація
допомоги українським військовим спочатку в зоні проведення АТО,
а нині на території всієї України. "Народний проект" був
заснований в березні 2014 року Давидом Арахамією.'
    },
    {
        'name': 'Картографічна Сотня',
        'description': 'Електронні та паперові карти, спеціальні
програми, навігатори.'
    },
    {
        'name': 'ГО Народний тил',
        'description': 'Одна з найбільших волонтерських
організацій України.'
    },

```

					ІАЛЦ 467200.007 Д4	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

```
{
    'name': 'Штаби волонтерських організацій',
    'description': 'Армія SOS, Повернись живим, Волонтерська
сотня, Крила Фенікса.'
},
{
    'name': 'Матриця технологій',
    'description': 'Виготовлення БПЛА і інших військових
технологій.'
}
]
```

```
resource_help = [
    {
        'location': 'м. Київ, вул. Металістів, 11.',
        'contact': '+380987630200'
    },
    {
        'location': 'м. Черкаси, вул. Городоцька, 128.',
        'contact': '+380987630300'
    },
    {
        'location': 'м. Рівне, вул. Острозька, 34.',
        'contact': '+380686632300'
    },
    {
        'location': 'м. Львів, вул. Вербицького, 11.',
        'contact': '+380686632300'
    }
]
```

```
# Command handler for /start command
```

					ІАЛЦ 467200.007 Д4	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

```

def start(update, context):
    context.bot.send_message(chat_id=update.effective_chat.id,
                             text="Привіт! Цей бот допоможе
тобі:\n"

                                "1.   □   Знайти   прихисток,
гуманітарну допомогу, волонтерство\n"

                                "2.   □   Запропонувати
прихисток, гуманітарну допомогу, волонтерство",

reply_markup=ReplyKeyboardMarkup([[KeyboardButton('Прихисток
□'), KeyboardButton('Гуманітарна допомога □')],

[KeyboardButton('Волонтерство □'),
KeyboardButton('Запропонувати допомогу □')]],

resize_keyboard=True, one_time_keyboard=True))

# Command handler for /search command
def search(update, context):
    context.bot.send_message(chat_id=update.effective_chat.id,
text="Виберіть категорію ресурсу:",

reply_markup=ReplyKeyboardMarkup([[KeyboardButton('Прихисток
□'), KeyboardButton('Гуманітарна допомога □')],

[KeyboardButton('Волонтерство □'),
KeyboardButton('Запропонувати допомогу □')]],

resize_keyboard=True, one_time_keyboard=True))

# Message handler for handling user messages
def handle_message(update, context):
    user_message = update.message.text.lower()

```

```

    if user_message == 'прихисток ☐':
        response = "☐ Список адрес куди можна прийти за
прихистком:\n\n"
        for resource in resources_shetler:
            response +=
f"{resource['location']}\nКонтактний номер:
{resource['contact']}\n\n"

context.bot.send_message(chat_id=update.effective_chat.id,
text=response)

    elif user_message == 'гуманітарна допомога ☐':
        response = "☐ Список організацій, що надають гуманітарну
допомогу:\n\n"
        for resource in resource_help:
            response +=
f"{resource['location']}\nКонтактний номер:
{resource['contact']}\n\n"

context.bot.send_message(chat_id=update.effective_chat.id,
text=response)

    elif user_message == 'волонтерство ☐':
        response = "☐ Список волонтерських організацій:\n\n"
        for resource in resources_volonter:
            response += f"{resource['name']} -
{resource['description']}\n"

context.bot.send_message(chat_id=update.effective_chat.id,
text=response)

    elif user_message == 'запропонувати допомогу ☐':
        response = "☐ Якщо ви бажаєте запропонувати допомогу,
будь ласка, зв'яжіться з нами за номером +380123456789."

```

					ІАЛЦ 467200.007 Д4	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

```

context.bot.send_message(chat_id=update.effective_chat.id,
text=response)

    else:

context.bot.send_message(chat_id=update.effective_chat.id,
text="Я не розумію вашого запиту.")


# Create the Telegram updater and pass it the bot's token
updater = Updater(token=TOKEN, use_context=True)


# Get the dispatcher to register handlers
dispatcher = updater.dispatcher


# Add command handlers
dispatcher.add_handler(CommandHandler('start', start))
dispatcher.add_handler(CommandHandler('search', search))


# Add message handler
dispatcher.add_handler(MessageHandler(Filters.text          &
~Filters.command, handle_message))


# Start the bot
updater.start_polling()
updater.idle()

```

					ІАЛІЦ 467200.007 Д4	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		