

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Навчально-науковий
Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістра

здобувача _____ Дугенця Олександра Олександровича _____
(П.І.Б.)

академічної групи _____ 123М -24-1 _____
(шифр)

спеціальності _____ 123 «Комп'ютерна інженерія» _____
(код і назва спеціальності)

за освітньо-професійною програмою _____ «Комп'ютерна інженерія» _____
(офіційна назва)

на тему _____ «Обґрунтування структури комп'ютерної системи ЦНАП
м. Дніпра з інтегрованим інтелектуальним чатботом» _____
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
кваліфікаційної роботи	проф. Цвіркун Л.І.			
розділів:				
синтез системи	проф. Цвіркун Л.І.			
розроблення програмного забезпечення	ас. Панферова Я.В.			
Рецензент				
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро
2025

ЗАТВЕРДЖЕНО:

завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)

_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)

" ____ " _____ 2025 року.

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістра

студенту Дугенець О.О. академічної групи 123М-24-1
(прізвище, ініціали) (шифр)

спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

за освітньо-професійною програмою 123 «Комп'ютерна інженерія»
(офіційна назва)

на тему «Обґрунтування структури комп'ютерної системи ЦНАП
м. Дніпра з інтегрованим інтелектуальним чатботом»

затверджена наказом ректора НТУ «Дніпровська політехніка» від 13.10.2025
№ 1165-с

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел сформулювати наукове завдання, конкретизувати предмет та мету досліджень	10.10.2025
Теоретичний	Обґрунтувати теоретичну базу розв'язання наукового завдання, якому присвячено роботу	24.10.2025
Синтез системи	Розробити комп'ютерну систему ЦНАП з інтегрованим інтелектуальним чатботом	14.11.2025
Розроблення програмного забезпечення	Розробити програмне забезпечення комп'ютерної системи ЦНАП	28.11.2025
Експериментальний розділ	Провести і обробити результати експериментів	05.12.2025

Завдання видано _____ проф. Цвіркун Л.І.
(підпис керівника) (прізвище та ініціали)

Дата видачі 05 вересня 2025 р.

Дата подання до атестаційної комісії 15.12.2025 р.

Прийнято до виконання _____ Дугенець О.О.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 81 с., 28 рис., 5 табл., 1 дод., 18 джерел.
КОМП'ЮТЕРНА СИСТЕМА, ІНТЕЛЕКТУАЛЬНИЙ ЧАТБОТ, ПРОГРАМА,
МОДЕЛЮВАННЯ, КОНТРОЛЬ, MONGODB, CHATGPT, СМО

Об'єкт дослідження: комп'ютерна система ЦНАП м. Дніпра з інтегрованим інтелектуальним чатботом.

Мета роботи: обґрунтування структури комп'ютерної системи ЦНАП м. Дніпра з інтегрованим інтелектуальним чатботом.

У вступі показано актуальність, мету, завдання, об'єкт, предмет, ідею дослідження.

У розділі «Стан питання і постановка задачі» розглянута сфера застосування об'єкта дослідження. Виконано аналіз відомих інтелектуальних чатботів, які вони надають послуги, особливості організації їх меню та технології їх розробки. Сформульовано мету і завдання роботи.

У теоретичному розділі описані методи обробки інформації, розроблена модель інтеграції інтелектуального чатбота в комп'ютерну систему ЦНАП, обґрунтована структура комп'ютерної системи ЦНАП м. Дніпра.

У розділі «Синтез системи» проаналізовані складові компоненти системи, вибрано обладнання та розроблена структурна схема комп'ютерної система ЦНАП м. Дніпра.

У розділі «Розроблення програмного забезпечення» обоснована структура і методи реалізації програмного забезпечення, розроблено програмне забезпечення чатбота комп'ютерної система ЦНАП м. Дніпра з підключеною базою даних MongoDB та моделлю ChatGPT.

В експериментальному розділі поставлена задача експерименту і обоснована методика його проведення та перевірена робота комп'ютерної системи комп'ютерної система ЦНАП м. Дніпра з інтегрованим інтелектуальним чатботом.

ЗМІСТ

	Стор
Перелік скорочень, умовних познач, одиниць і термінів	6
Вступ	7
1 Стан питання і постановка завдання	9
1.1 Аналіз сфери надання адміністративних послуг та умов застосування комп'ютерної системи	9
1.2 Характеристика Центру надання адміністративних послуг міста Дніпро	12
1.3 Огляд працюючих чатботів в м. Дніпро та визначення можливих напрямків рішення для нових	14
1.4 Постановка завдання	20
2 Теоретична частина	21
2.1 Аналіз чатботів месенджера Телеграм	21
2.2 Реєстрація чатбота в месенджері Telegram	27
2.3 Особливості створення програмного забезпечення ЦНАП м. Дніпра	32
2.3.1 Особливості створення чатбота	32
2.3.2 Особливості підключення бази даних MongoDB до чатбота	36
2.3.3 Особливості підключення моделі ChatGPT до бота	39
2.4 Розробка моделі чатбота Центра надання адміністративних послуг	42
3 Синтез комп'ютерної системи ЦНАП м. Дніпра	46
3.1 Вимоги до системи в цілому	46
3.2 Технічні вимоги до комп'ютерної системи ЦНАП м. Дніпра	46
3.2.1 Вимоги до надійності комп'ютерної системи	46
3.2.2 Вимоги до безпеки комп'ютерної системи	47
3.2.3 Вимоги до функцій, виконуваних системою	47
3.2.4 Вимоги до технічного забезпечення системи	48
3.2.5 Вимоги до програмного забезпечення	48
3.3 Розробка структурної схеми комп'ютерної системи ЦНАП м. Дніпра	49

4	Розроблення програмного забезпечення ЦНАП м. Дніпра	50
4.1	Призначення і сфера застосування програми	50
4.2	Обґрунтування технічних характеристик програми	50
4.2.1	Постановка завдання на розробку програми	50
4.2.2	Структура і алгоритм функціонування програми	50
4.2.3	Вибір складу програмних засобів	50
4.3	Опис розробленої програми	51
4.3.1	Загальні відомості	51
4.3.2	Функціональне призначення	51
4.3.3	Використовувані технічні засоби	51
4.3.4	Опис логічної структури	51
4.3.5	Вхідні дані та вихідні дані	56
5	Експериментальний розділ	57
5.1	Постановка завдання експерименту і обґрунтування методики	57
5.2	Вимоги до експерименту.....	59
5.3	Програма експерименту	59
5.4	Результати експерименту	60
	Висновки	63
	Перелік посилань	64
Додаток А	Текст програми	66

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

БД – база даних;

КС – комп'ютерна система;

КМ – корпоративна мережа;

ПК – персональний комп'ютер;

ПЗ – програмне забезпечення;

ЛК – локальна мережа;

СМО – систем масового обслуговування;

ШІ – штучний інтелект;

LAN – Local Area Network ;

MAN – Metropolitan Area Network ;

MongoDB – документоорієнтована нереляційна база даних

TCP/IP – Transmission Control Protocol/Internet Protocol;

UTP – Unshielded twisted pair ;

WAN – Wide Area Network.

ВСТУП

Комп'ютерні системи на сьогодні являються головною частиною підприємств, так як забезпечують передавання інформації не тільки між окремими підрозділами, а і зв'язок і з зовнішнім світом [14].

Не являється виключенням з цього і Центр надання адміністративних послуг м. Дніпра. Комп'ютерна система об'єднує чотири відділи ЦНАП ("Правобережний", "Лівобережний", "Північний" та "Я-Ветеран") і віддалене робоче місце адміністратора по вул. Воскресенська, 9 [7].

Всі верстви населення так чи інакше потребують зручного сервісу при отриманні адміністративних послуг. Одним із елементів такого сервісу являється доступ до інформації Центрів через чатботи месенджерів із застосуванням інтелектуальних технологій [12].

Мета кваліфікаційної роботи обґрунтування структури комп'ютерної системи ЦНАП з інтегрованим інтелектуальним чатботом.

Для досягнення поставленої мети необхідно розглянути теорію та приклади реалізації комп'ютерних систем та мереж Центрів надання адміністративних послуг, особливості їх проектування, далі розробити структурну схему комп'ютерної системи та програмне забезпечення включаючи і інтелектуальний чатбот.

Мета і завдання дослідження: *Метою роботи є обґрунтування структури комп'ютерної системи Центра надання адміністративних послуг міста Дніпра з інтегрованим інтелектуальним чатботом.*

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз сфери надання адміністративних послуг та умов застосування системи та огляд відповідних відділів;
- виконати аналіз розроблених інтелектуальних чатботів в сфері надання адміністративних послуг;
- вивчити особливості розробки комп'ютерних систем і інтеграцію в них інтелектуальних чатботів мовою Python;

- розробити структуру комп'ютерної системи Центра надання адміністративних послуг міста Дніпра;
- розробити програмне забезпечення комп'ютерної системи Центра надання адміністративних послуг міста Дніпра з інтегрованим інтелектуальним чатботом;
- провести експериментальні дослідження і обґрунтувати структуру комп'ютерної системи Центра надання адміністративних послуг міста Дніпра з інтегрованим інтелектуальним чатботом.

Об'єкт дослідження: комп'ютерна система Центра надання адміністративних послуг міста Дніпра з інтегрованим інтелектуальним чатботом.

Предмет дослідження: структура комп'ютерної системи Центра надання адміністративних послуг міста Дніпра.

Методи дослідження: математичні методи проектування комп'ютерних систем, експертного оцінювання, теорія моделювання та дизайну інтерфейсу користувача.

Наукові положення:

1. Встановлено, що введення інтелектуального чатбота в комп'ютерну систему Центра надання адміністративних послуг міста Дніпра забезпечує збільшення показник задоволеності обслуговуванням на 20.6%.

Наукові результати:

1. Обґрунтовано, що для підвищення ефективності чатбота Центра надання адміністративних послуг необхідно підключення штучного інтелекту.

Обґрунтованість і достовірність наукових положень, висновків і рекомендацій підтверджуються результатами експериментальних досліджень.

Практичне значення отриманих результатів полягає в розробці програмного забезпечення, яке дозволяє при наданні адміністративних послуг підключати інтелектуальний чатбот, що підвищує ефективність обслуговування.

1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз сфери надання адміністративних послуг та умов застосування комп'ютерної системи

Нормативна база надання адміністративних послуг в Україні базується на Законі України «Про адміністративні послуги» та Законі «Про адміністративну процедуру», які встановлюють загальні правила та принципи, а також на низці підзаконних актів Кабінету Міністрів України (Постанови № 44, № 118, № 588), що регулюють роботу Центрів надання адміністративних послуг (ЦНАП), ведення реєстрів та порталу, а також спеціальних законах, що стосуються окремих сфер діяльності (наприклад, про реєстрацію бізнесу чи паспортні послуги).

Перелік адміністративних послуг в Україні охоплює широкий спектр питань: від реєстрації місця проживання, паспортних послуг (ID-картки, закордонні паспорти), соціального захисту (субсидії, допомоги) та реєстрації бізнесу, до питань будівництва, земельних відносин, комунального господарства, сім'ї та дітей, які надаються переважно через Центри Надання Адміністративних Послуг (ЦНАП) [7]. Конкретний перелік залежить від органу місцевого самоврядування, але загалом це послуги, пов'язані з довідками, дозволами, реєстрацією та видачою документів, як-от оформлення документів, реєстрація нерухомості чи бізнесу, отримання соціальних послуг.

Основні категорії послуг представлено в таблиці 1.2.

В Україні підхід до надання адміністративних послуг базується на принципах сервісної держави, клієнтоорієнтованості, цифровізації (через портал Дія та електронні сервіси) та децентралізації, що реалізується через Центри надання адміністративних послуг (ЦНАП) для забезпечення зручності, прозорості та ефективності, мінімізуючи бюрократію, хоча ще існують виклики щодо повного охоплення та уніфікації послуг.

Таблиця 1.1 – Основні категорії адміністративних послуг в Україні

№	Категорія	Назви послуг
1.	Реєстрація та громадянство	Реєстрація/зняття з реєстрації місця проживання, видача паспортів (ID, закордонний), довідки з реєстрів
2.	Підприємництво	Реєстрація бізнесу, отримання дозволів (наприклад, на розміщення реклами, тимчасових споруд)
3.	Земля та будівництво	Містобудівні умови та обмеження, оформлення прав на землю, дозвільні документи на будівництво
4.	Майно та нерухомість	Реєстрація речових прав, видача свідоцтв про право власності
5.	Сім'я та діти	Питання опіки, усиновлення, оформлення свідоцтва про народження, допомога при народженні ("єМалятко")
6.	Соціальний захист	Призначення субсидій, допомоги, видача посвідчень ветеранам та іншим категоріям

Ключовими принципами та інструментами такого підходу є наступні:

1. ЦНАП як основний центр: Об'єднання різних послуг в одному місці (від реєстрації місця проживання до отримання дозволів), що спрощує доступ для громадян.
2. Дія та цифровізація: Портал та додаток "Дія" дозволяють подавати заяви та отримувати послуги онлайн, що є важливим кроком до сервісної держави.

3. Клієнтоорієнтованість: Орієнтація на потреби громадянина, спрощення процедур, скорочення часу очікування та підвищення якості обслуговування.
4. Децентралізація: Передача повноважень на місцевий рівень (громади), що дозволяє адаптувати послуги до місцевих потреб.
5. Уніфікація та стандартизація: Прагнення до стандартизації вимог та процесів для уникнення різноспрямованості та підвищення сумісності.

Послуга при такому підході проходить реєстрацію звернення, обробку та оформлення результату і видачу кінцевого документа (наприклад, через код доступу).

Основними цілями такого підходу є: підвищення довіри населення до влади, подолання корупції та бюрократії та формування ефективної взаємодії між державою та громадянами.

І це дає наступні виклики: потрібне подальше збільшення кількості послуг, доступних в електронному вигляді, необхідність подальшої стандартизації та усунення опору з боку деяких відомств та розробки підходів для максимального суспільно-корисного результату за мінімальних ресурсів.

1.2 Характеристика Центру надання адміністративних послуг міста Дніпро

В місті Дніпро надання адміністративних послуг виконується відповідно до конституції та законів України та затвердженого Дніпровською міською Радою «Положення про Центр надання адміністративних послуг м. Дніпра» (рішення міської ради від 25.09.2013 №41/40, у редакції рішення міської ради від 23.02.21 №83/4).

Центр надання адміністративних послуг (ЦНАП) м. Дніпра» підпорядкований управлінню Департаменту, підконтрольний та підзвітний Дніпровській міській раді, її виконавчому комітету, міському голові та секретарю Дніпровської міської ради [7].

Метою створення та функціонування ЦНАП є забезпечення зручного та оперативного отримання адміністративних послуг суб'єктами звернень, максимального спрощення організації та технологій їх виконання, відкритість і прозорість умов при розгляді звернень громадян та суб'єктів господарювання, реалізація принципу дистанціювання суб'єкта звернення від суб'єкта надання послуг.

Центр надання адміністративних послуг міста Дніпро) включає наступні структурні підрозділи: чотири відділи і віддалене робоче місце адміністратора по вул. Воскресенська, 9 (таблиця 1.2).

Найбільш затребуваними функціями ЦНАП міста Дніпро являються наступні:

1. Надання витягу з Єдиного державного реєстру МВС.
2. Видача/заміна паспорта громадянина для виїзду за кордон.
3. Видача паспорта громадянина для виїзду за кордон для дитини.
4. Отримання субсидії на оплату житлово-комунальних послуг
5. Внесення змін до реєстраційних даних фізичної особи-підприємця.
6. Звернення громадян до міської ради.

Таблиця 1.2 – Структурні підрозділи Центра надання адміністративних послуг міста Дніпра

№	Назва	Адреса	Е-mail	Телефон
1	Відділ ЦНАП "Правобережний"	м. Дніпро, вул. Марії Кюрі, 5 (ТРЦ «NEO PLAZA») 2поверх;	adminposlug@dniprorada.gov.ua dnepr-cnap@ukr.net	097 807 37 07; 099 203 09 25
2	Відділ ЦНАП "Лівобережний"	м. Дніпро, пр.Слобожанський, (Газети «Правда»), 31д	adminposlug@dniprorada.gov.ua dnepr-cnap@ukr.net	068 237 52 76; 099 602 10 02
3	Відділ ЦНАП "Північний"	м. Дніпро, вул. Нижньодніпровська, 17 (КАРАВАН)	dnepr-cnap@ukr.net adminposlug@dniprorada.gov.ua	098 780 62 39
4	Відділ ЦНАП "Я- Ветеран"	м. Дніпро, проспект Слобожанський, 8	adminposlug@dniprorada.gov.ua dnepr-cnap@ukr.net	066 319 39 96; 098 449 52 52
5	Віддалене робоче місце адміністратора ЦНАП міста Дніпра	м. Дніпро, вул. Воскресенська, 9	adminposlug@dniprorada.gov.ua dnepr-cnap@ukr.net	066 546 46 40

7. Надання довідки про притягнення до кримінальної відповідальності, відсутність (наявність) судимості або обмежень, передбачених кримінально-процесуальним законодавством України

У середньому один ЦНАП надає понад 100 видів послуг – і їх перелік постійно розширюється.

1.3 Огляд працюючих чатботів в м. Дніпро та визначення можливих напрямків рішення для нових

Чатбот – це комп'ютерна програма, яка здатна взаємодіяти з користувачами у формі текстового або голосового чату. Чатботи часто використовуються для надання клієнтської підтримки, продажів або навіть просто для розмови [4].

Чатботи створюються для роботи в месенджерах і замінюють людей, імітуючи спілкування. Вони можуть інформувати людей, тобто підготовлювати відповіді на їх різноманітні запитання. Вони формують обмін повідомлень в системі «людина-комп'ютер» [11, 12, 16].

Такі боти можуть повідомляти курси валют, прогноз погоди, розклад руху, поширювати рекламу, формувати звернення до органів влади тощо.

Основними перевагами чатботів є:

- підприємство економить кошти не наймаючи нового робітника;
- зручність застосування, як особистого помічника;
- можливість організації через нього нового каналу для продажу;
- простота застосування різних алгоритмів спілкування.

Розглянемо які чатботи працюють в м. Дніпро.

У Дніпрі працює чатбот ТОВ «ЄРЦ КП», який призначений для контролю стану особового рахунку, отримання інформації щодо заборгованості за комунальні послуги, внесення показників приладів обліку за одним або кількома особовими рахунками та сплати рахунків за комунальні послуги. Також можливо дізнатися адреси центрів обслуговування абонентів та увійти до особистого кабінету КП «Дніпроводоканал» (рисунок 1.1).

Тепер відстежувати нарахування за комунальні послуги та проводити сплату за них стало ще простіше. Щоб працювати з чатботом в Viber, треба відсканувати QR-код або треба перейти за посиланням [6].

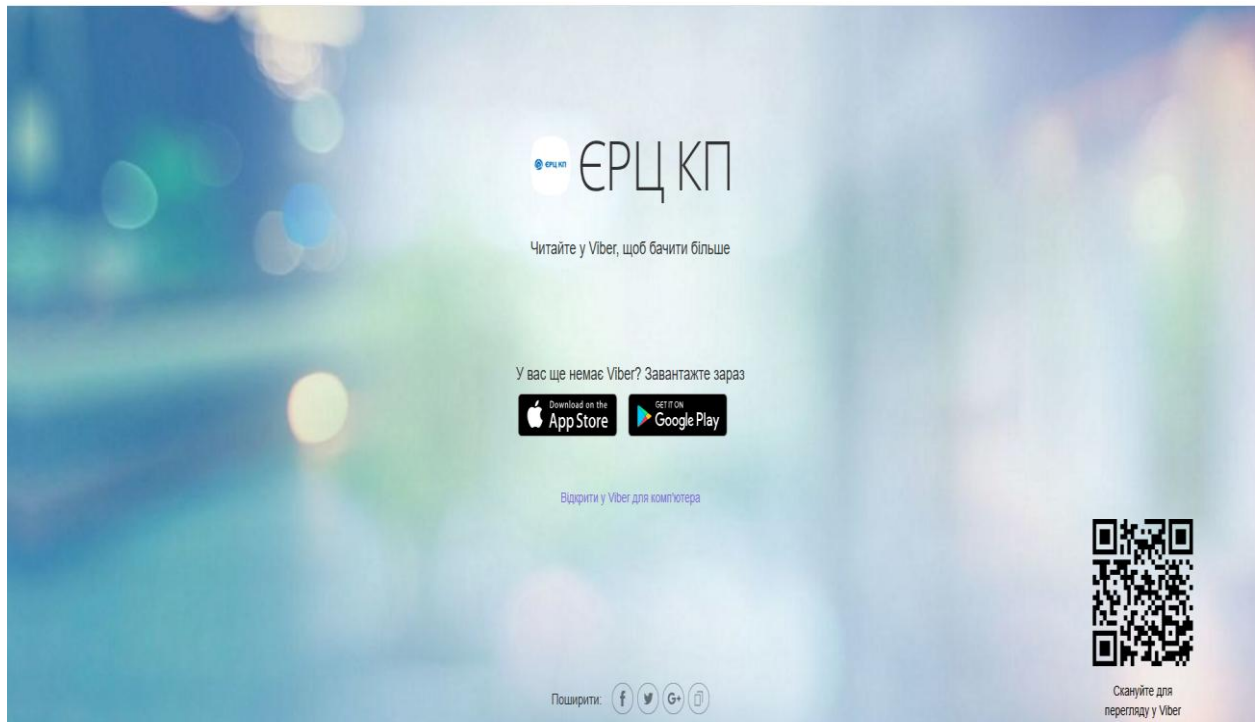


Рисунок 1.1 – Початкова сторінка чатбота ТОВ «ЄРЦ КП»

У Дніпрі працює чатбот контакт-центру міськради (dniprotoc_бот) [7]. Цей бот допомагає сформувати звернення від жителів Дніпра стосовно аварійних ситуацій, вказати адресу, додати фото з місця події, контролювати хід виконання по зверненню, отримувати повідомлення про зміну статусу звернення, змінювати особисті дані, переглядати історію звернень та оцінювати якість їх виконання. Початкова сторінка чатбота контакт-центру міськради показана на рисунку 1.2.

Також у Дніпрі працює чатбот розкладу транспорту Дніпра. Початкова сторінка чатбота показана на рисунку 1.3, а меню, після команди /start – на рисунку 1.4.

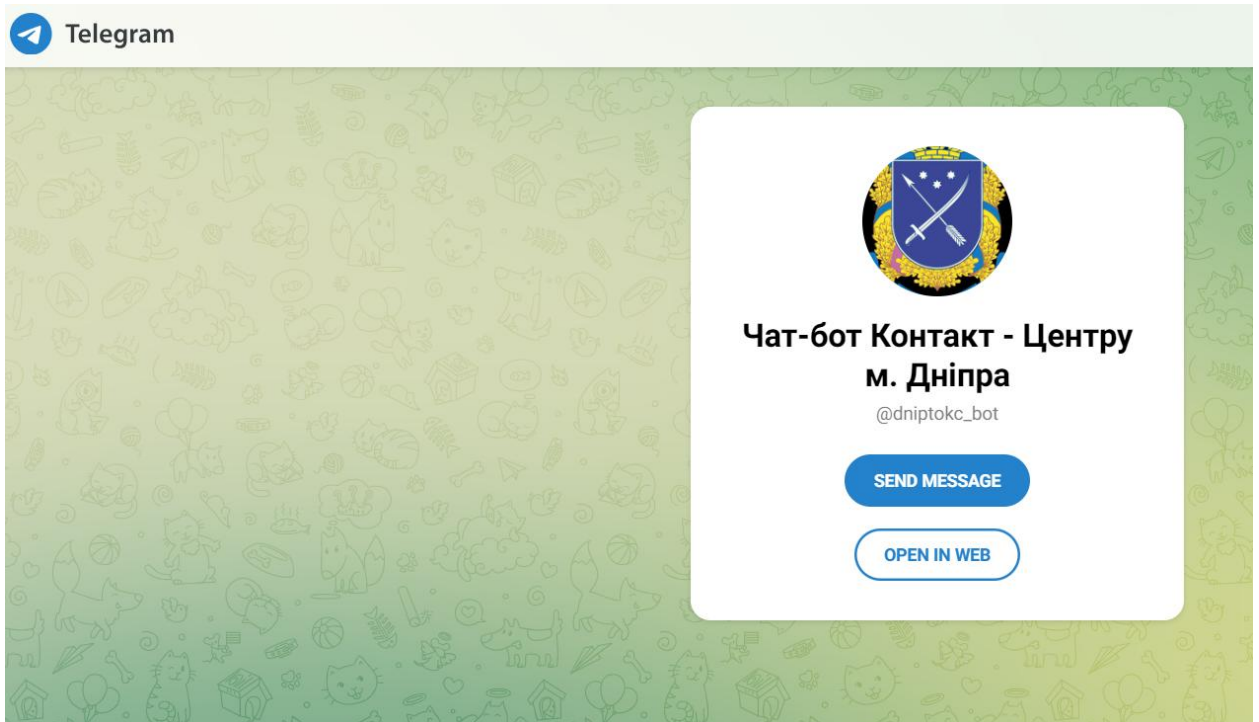


Рисунок 1.2 – Початкова сторінка чатбота контакт-центру міськради

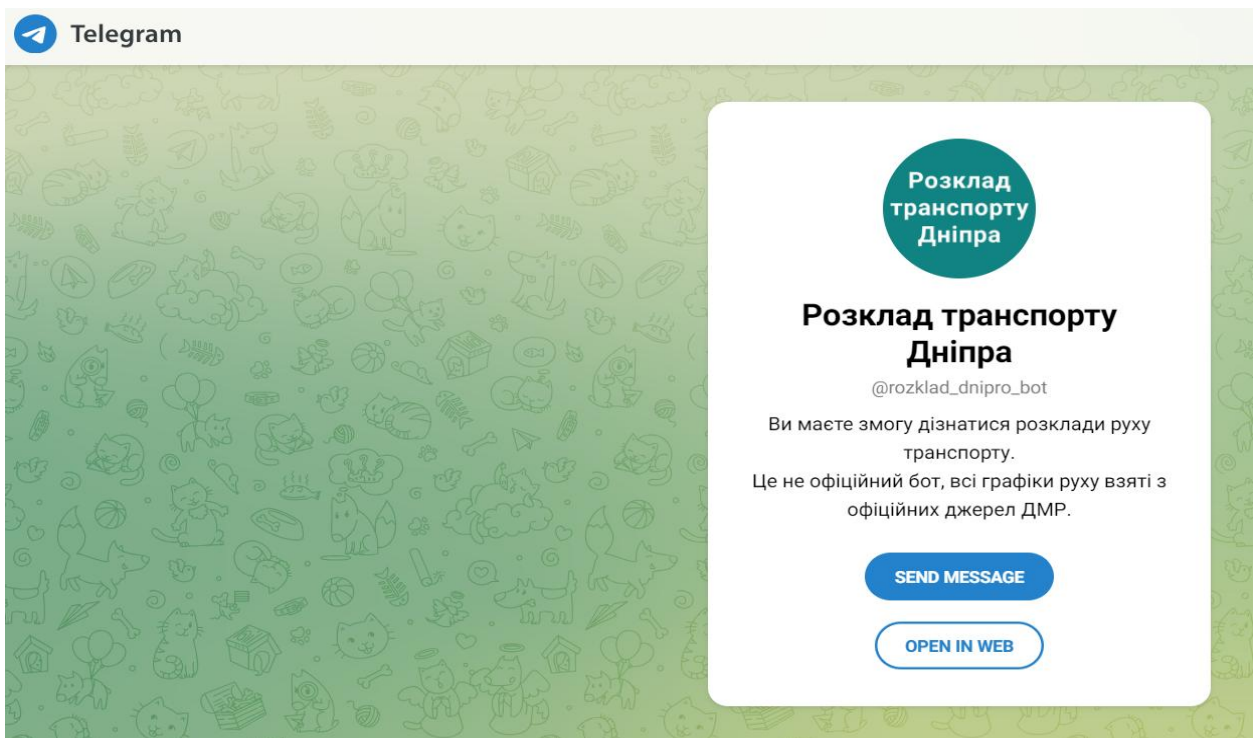


Рисунок 1.3 – Початкова сторінка чатбота розкладу транспорту Дніпра



Розклад транспорту Дніпра
bot

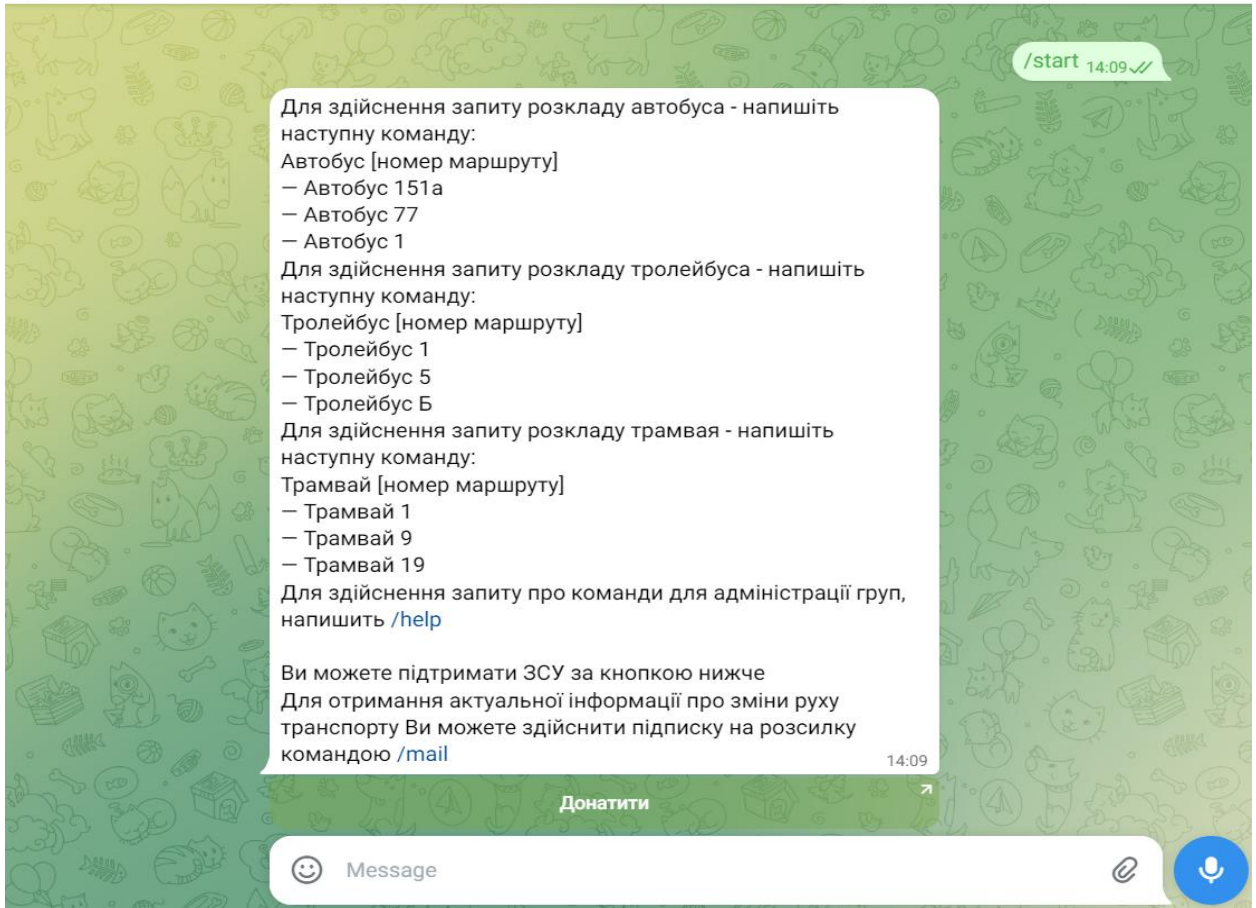


Рисунок 1.4 – Сторінка чатбота розкладу транспорту Дніпра по команді /start

Телеграм-бот Krim_dnp_bot допомагає боротися зі зрадниками, рисунок 1.5 [https://adm.dp.gov.ua/news/meshkanci-dnipropetrovshchini-mozhut-povidomlyati-pro-zradnikiv-cherez-specialnij-policejskij-chat-bot].

На нього можна присилати фото та іншу інформацію на тих хто працює на ворога.

Для мешканців м. Дніпро став доступний чатбот Google Bard [6] на базі штучного інтелекту, який може створювати макети запрошень, планувати вечірки, підготувляти перелік аргументів для окремих рішень тощо (рисунок 1.6). Чатбот Google Bard може допомогти по напрямам, які показані на рисунку 1.7.

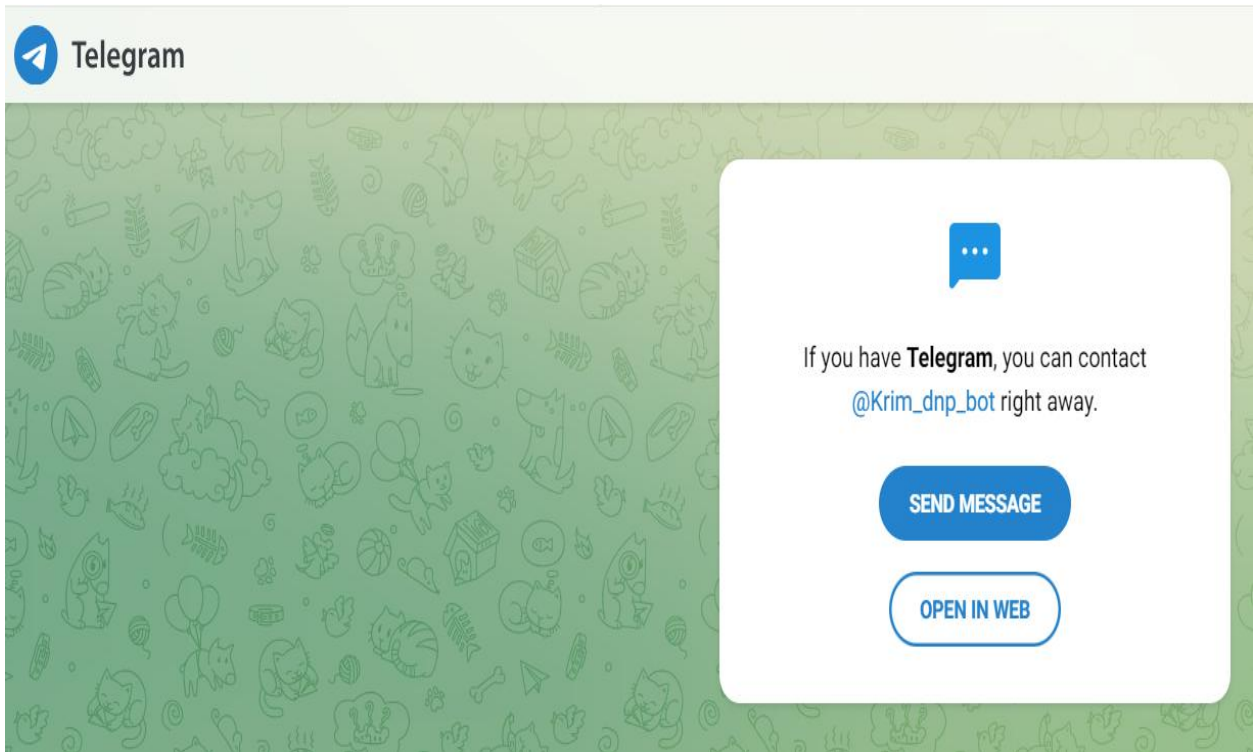


Рисунок 1.5 – Початкова сторінка чатбота для боротьби зі зрадниками

Bard Експеримент Поширені запитан


[Перегляньте останні новини Центру конфіданційності](#)

Bard

Інструмент на основі розмовної моделі штучного інтелекту від Google

Bard допоможе вам відшукати нові ідеї, підвищити продуктивність, а також розкрити творчий потенціал.

[Спробувати Bard](#)



♦ Ось кілька конкретних рекомендацій про те, як почати їсти більше здорової їжі:

- **Почніть з малого.** Не намагайтеся змінити своє харчування кардинально в один день. Почніть з невеликих кроків, таких як додавання однієї здорової страви в свій

Рисунок 1.6 – Початкова сторінка чатбота Google Bard

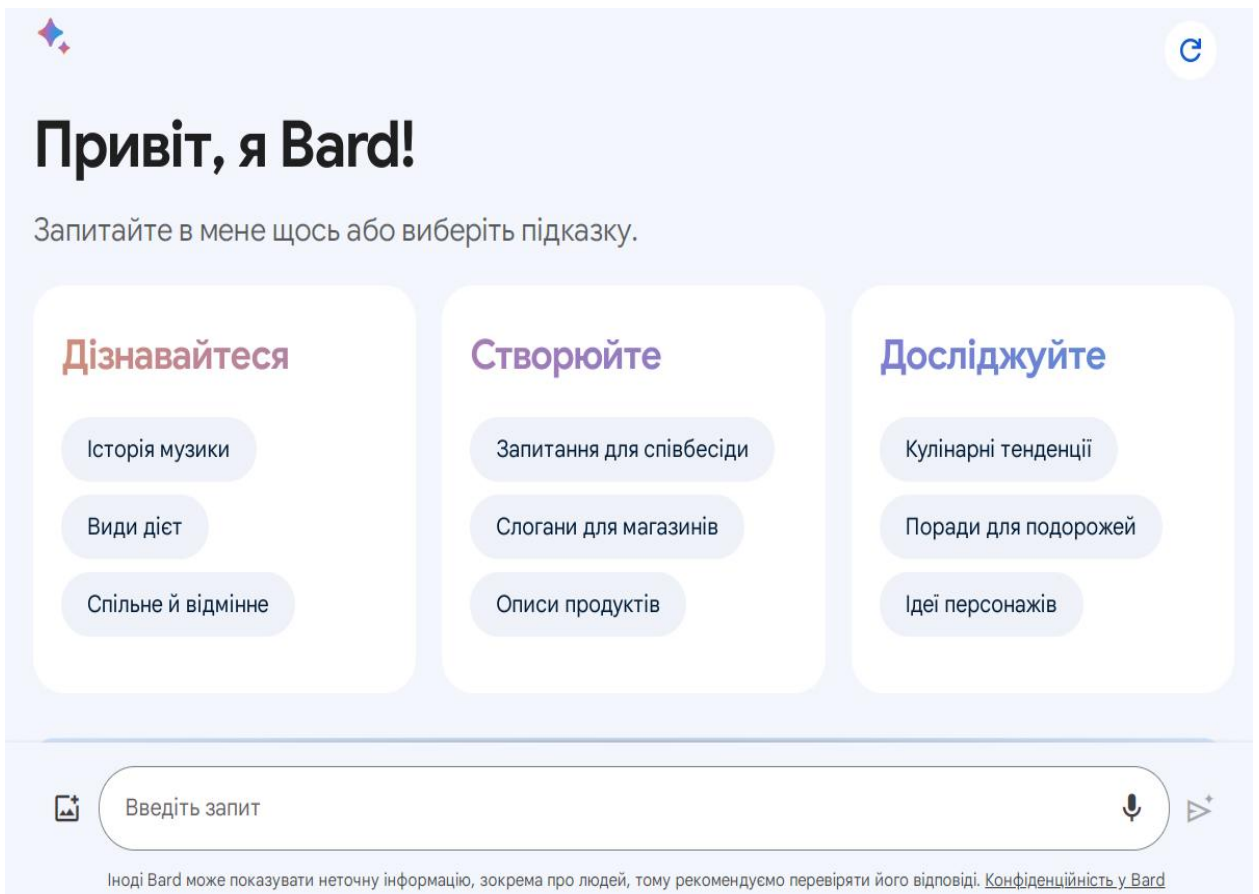


Рисунок 1.7 – Напрями допомоги чатбота Google Bard

1.4 Постановка завдання

Основним завданням кваліфікаційної роботи є обґрунтування структури комп'ютерної системи Центру надання адміністративних послуг міста Дніпро з інтегрованим інтелектуальним чатботом.

Комп'ютерна система ЦНАП міста Дніпра з чатботом дозволить надавати адміністративні послуги мешканцям Дніпра завдяки онлайн-сервісу за графіком 24/7. Це значно підвищить рівень сервісу, але потребує відповідно і такого режиму комп'ютерної системи.

Введення інтелектуального чатбота в комп'ютерну систему підвищить доступність для певних категорій населення до сервісів ЦНАП. Це надасть можливість працівникам Центру більше часу приділяти населенню і надавати індивідуальну допомогу, ніж чатбот, а чатботи можуть обробляти стандартні запити.

Для забезпечення роботи комп'ютерної мережі ЦНАП необхідно використовувати стандарт 1000BASE-T. Комп'ютерна система повинна складатися з 5 підсистем.

При поєднанні комп'ютерної системи ЦНАП з ботом можна отримати ряд наступних переваг:

- чатбот може виконувати рутинну роботу, наприклад, надавати інформацію про адміністративні послуги, допомагати у заповненні документів та звернень.

- звільнення працівників ЦНАП для надання більш індивідуальної допомоги та вирішення складніших питань.

Таким чином комп'ютерна система ЦНАП з інтелектуальним чатботом може стати ефективним інструментом для надання адміністративних послуг населенню. Вона підвищить доступність послуг та допоможе розвантажити працівників ЦНАМ міста Дніпро.

2 ТЕОРЕТИЧНА ЧАСТИНА

2.1 Аналіз чатботів месенджера Телеграм

Одним з найвідоміших месенджерів являється Telegram. Він являється першим, який запусив чатботи і забезпечує передачу текстових і відео повідомлень.

Розглянемо найбільш відомі чатботи в месенджері Telegram.

Бот Railway – дозволяє виконувати швидкий пошук квитків на потяги та автобуси по Україні та Європі. Railway це: придбання квитка за секунди – без черг та очікувань, миттєва оплата і квиток одразу у Telegram, обрання найзручнішого маршруту серед сотень варіантів.

Щоб почати пошук квитків треба натиснути кнопку внизу екрану, потім вказати місце відправки та місце призначення і дату. На екран висвітлюються наявні. Якщо квитки не влаштовують користувача, можна зробити заявку і бот присилає повідомлення коли з'явиться необхідний квиток (рисунки 2.1 – 2.3).

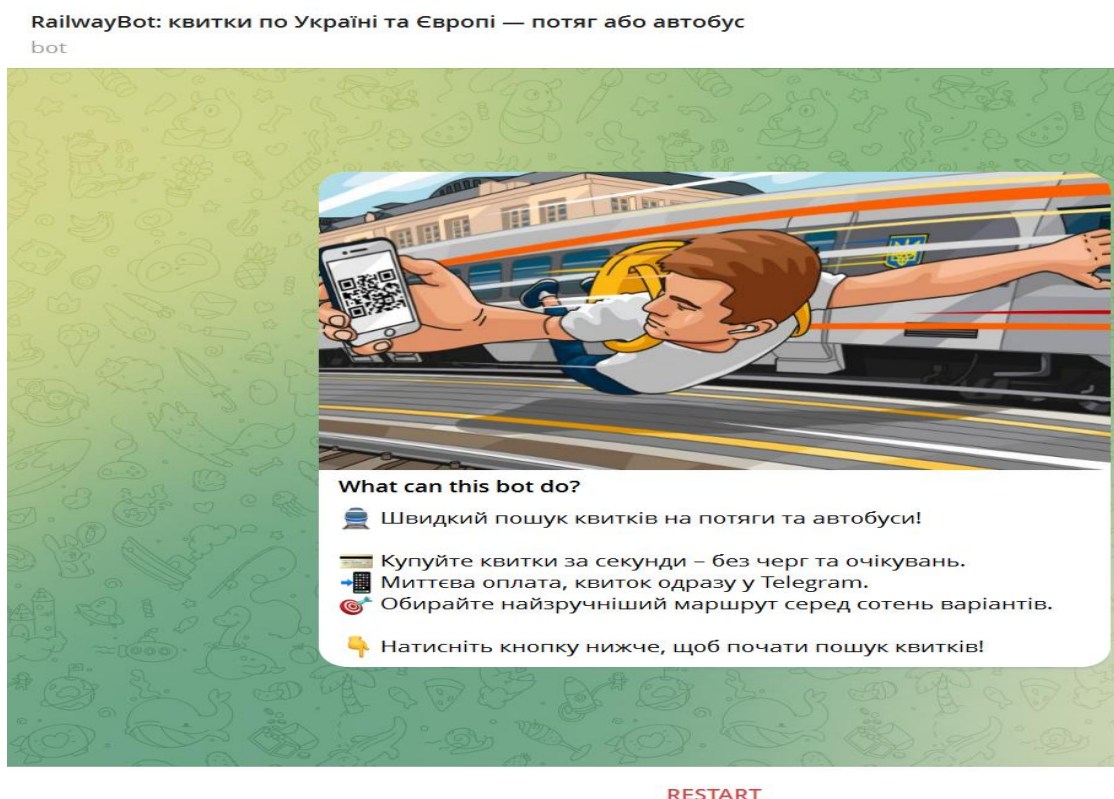


Рисунок 2.1 – Початкова сторінка чатбота Railway



Рисунок 2.2 – Початок роботи з чатботом Railway

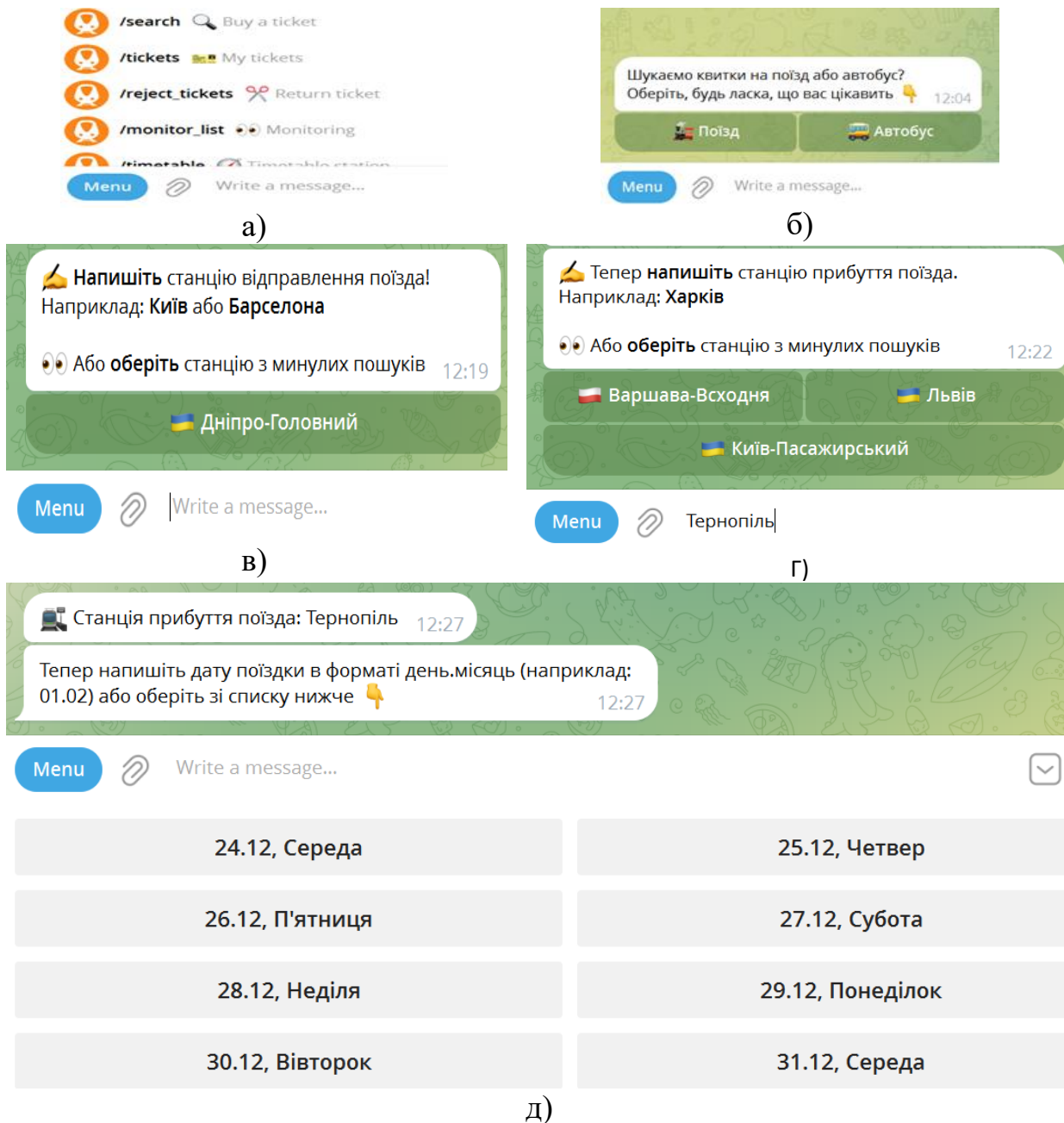


Рисунок 2.3 – Пошук квитків в чатботом Railway: а) початок пошуку квитків; б) вибір виду транспорту; в) визначення станції відправлення; г) визначення станції відправлення; д) визначення дати поїздки

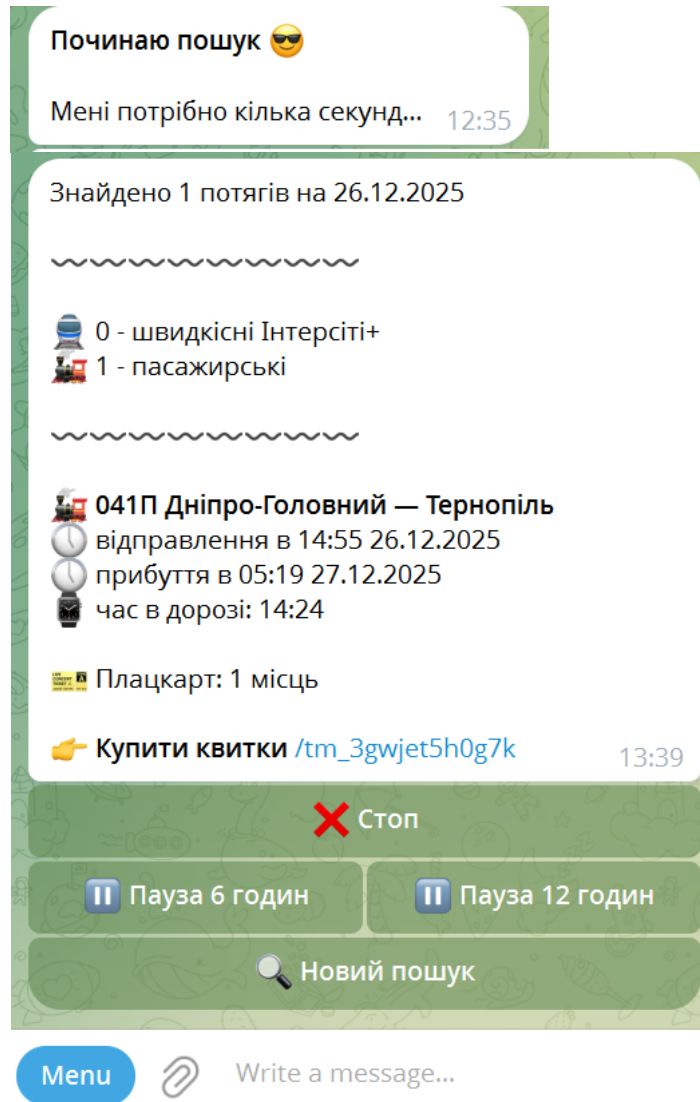


Рисунок 2.4 – Приклад вибору квитка в чатботі Railway

Бот PrivatBankBot дозволяє клієнтам відправляти і отримувати гроші через месенджер за спрощеною процедурою. Бот підключається у обидвох клієнтів з вказанням телефону, дозвільних 4 цифр банківської картки та реєструється через sms (рисунок 2.5).

За допомогою чатбота NPBot можна знайти найближче відділення Нової пошти, відслідковувати відправлення і повідомити про стан посилки (рисунок 2.6).

ПриватБанкБот

bot



Рисунок 2.5 – Приклад початку роботи чатбота PrivatBankBot

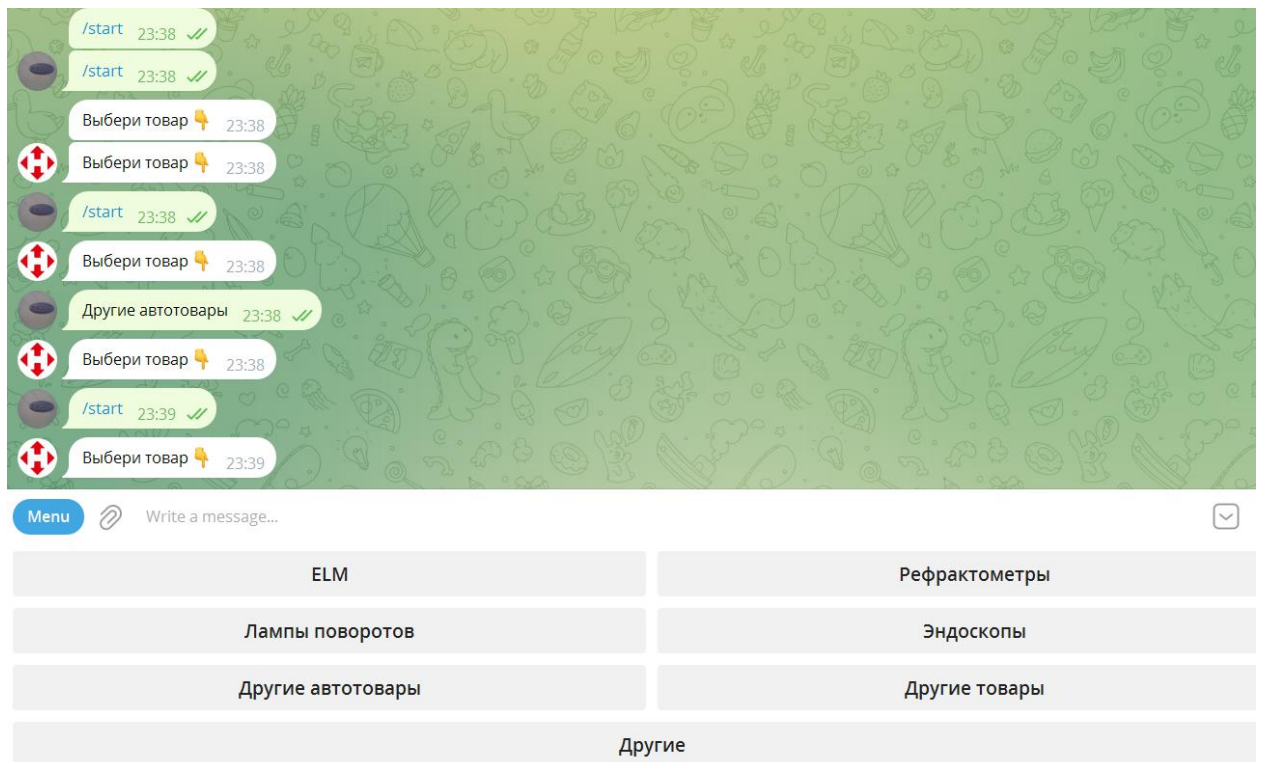


Рисунок 2.6 – Приклад початку роботи чатбота NPBot

Відкриті дані можна дізнатися за допомогою OpenDataBot. Бот допоможе знайти українську команду, якщо відомі її назва, код ОКПО чи прізвище директора (рисунок 2.7).

Бот при звичайній реєстрації отримує дані і судовий реєстр один раз на сім днів, а при професійній – щоденно.

Опендатамедіа

26 703 subscribers



Pinned message

Привіт, друзі! Якщо ви ще не почали працювати з Opendatabot — ця інструкція для вас. Тут є

Опендатамедіа

Кількість осіб у Реєстрі волонтерів

Станом на 7 грудня 2023 року

Місяць	Кількість осіб
Січ '22	~100
Фев '22	~100
Мар '22	~100
Тра '22	~100
Трав '22	~100
Чер '22	~100
Лип '22	~100
Серп '22	~100
Вер '22	~100
Жовт '22	~100
Листопад '22	~100
Грудень '22	~100
Січ '23	~200
Фев '23	~300
Мар '23	~400
Тра '23	~500
Трав '23	~600
Чер '23	~700
Серп '23	~800
Вер '23	~900
Жовт '23	~1000
Листопад '23	~1100
Грудень '23	6 734

● Кількість осіб

● Опендатабот

© 2023 Опендатабот — відкриті дані

6734 особи зареєстровано у Реєстрі волонтерів на початок грудня 2023 року. Офіційних волонтерів побільшало у 2,5 рази — за неповний 2023 рік додалось 4018 осіб.

Загалом від початку повномасштабної війни реєстр виріс у 21 раз. Пік реєстрацій припав на грудень 2022 року. Тоді до переліку додалось 1653 волонтери за місяць. Наразі кількість українців, які б хотіли б приєднатись до реєстру, скорочується квартал до кварталу.

Варто зазначити, що за 11 місяців 2023 року з Реєстру волонтерів було виключено 69 осіб. Для порівняння, за весь 2022 рік з переліку було виключено лише 4 людини.

Реєстрація в Реєстрі Податкової потрібна, аби на значні суми донорських коштів не нараховувались податки, як на доходи.

Деталі: <https://opendatabot.ua/analytics/real-volunteers-2023-12>

👍 14 ❤️ 4

👁️ 4580 9:00

JOIN CHANNEL

Рисунок 2.7 – Приклад початку роботи чатбота OpenDataBot

2.2 Реєстрація чатбота ЦНАП міста Дніпра в месенджері Telegram

Зареєструємо в месенджері Telegram чатбот для Центру надання адміністративних послуг з ім'ям Dnipro TsNAP Bot.

Для реєстрації чатбота в месенджері запустимо Telegram і в контактах знайдемо **@BotFather** (рисунок 2.8) і введемо команду /newbot для створення нового чатбота TsNAP Dnipro (рисунок 2.9), користувачке його ім'я TsNAP_Dnipro_bot [13, 14].



Рисунок 2.8 – Виклик чатбота **@BotFather**

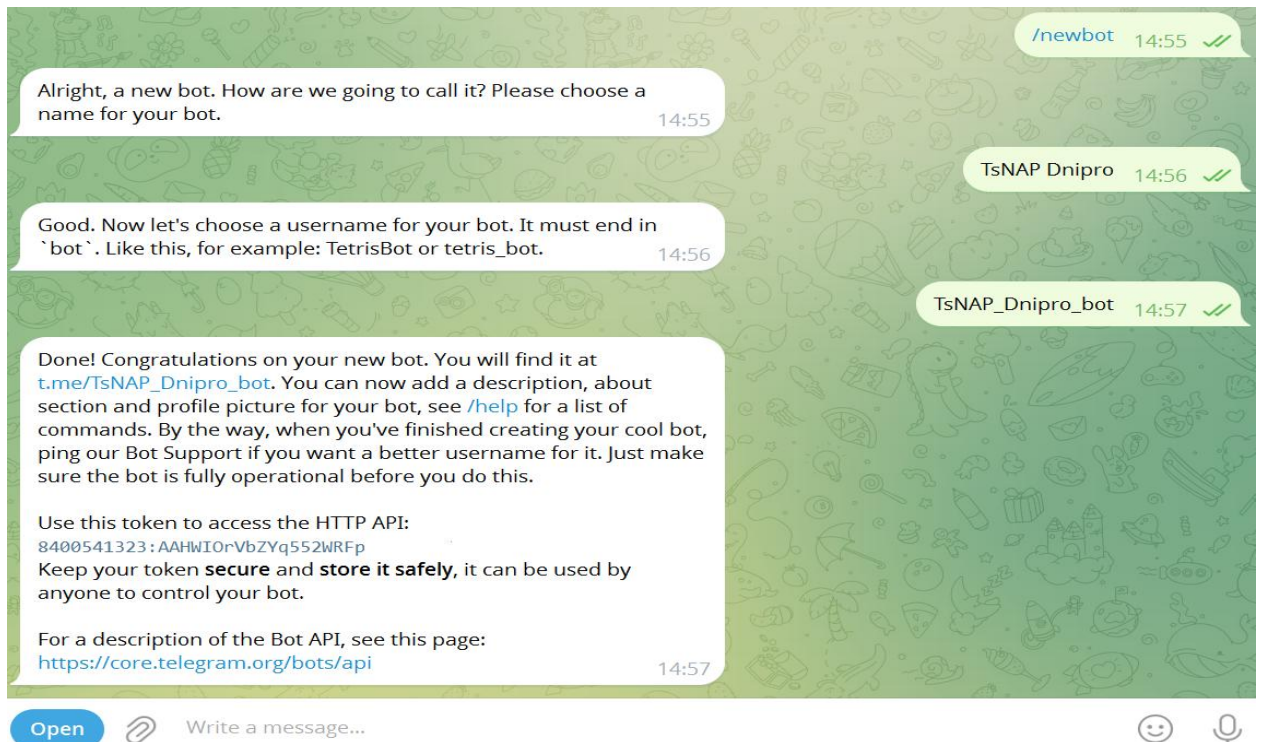


Рисунок 2.9 – Створення чатбота з ім'ям TsNAP Dnipro

Введемо команду `/mybots` і перейдемо до меню для отримання інформації про створений bot з ім'ям TsNAP Dnipro (рисунок 2.10).

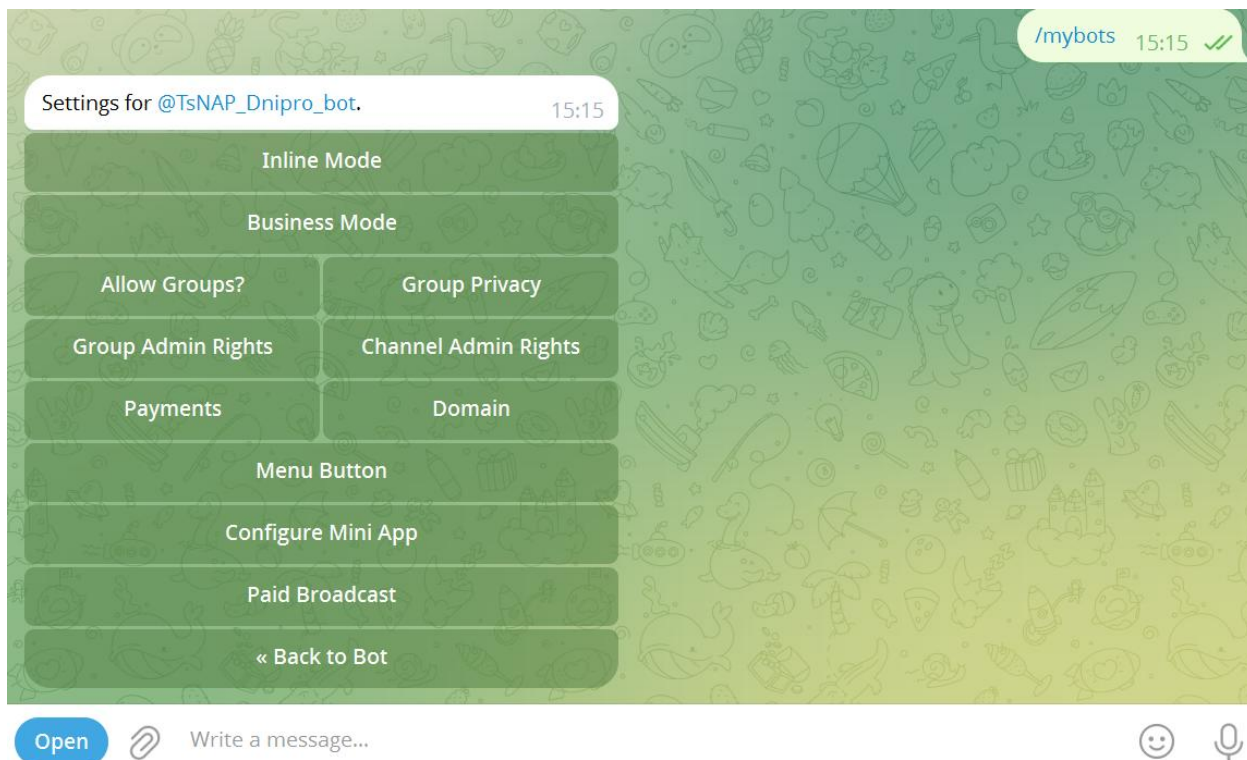


Рисунок 2.10 – Меню для отримання інформації про створений чатбот з ім'ям TsNAP Dnipro

Перелік команд можливо вивести на екран командою `/help` (рисунок 2.11).

Для отримання списку оновлень для створеного бота наберемо в вебпереглядаче команду:

`https://api.telegram.org/bot<Token>/getUpdates`

де: `<Token>` – код доступу, який був отримано при створенні чатбота з ім'ям TsNAP Dnipro.

В результаті отримаємо таку відповідь – рисунок 2.12.

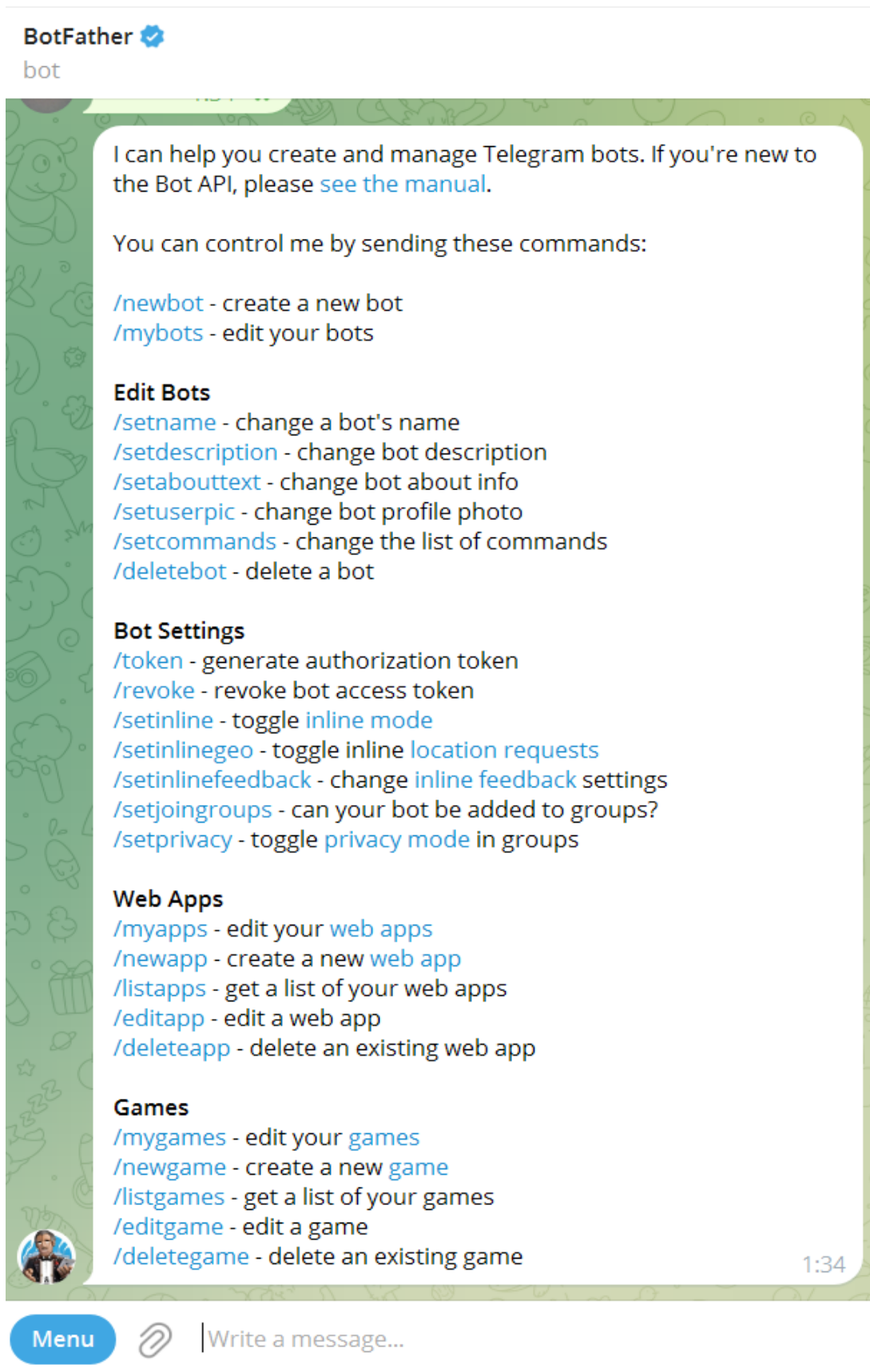


Рисунок 2.11 – Виведення переліку команд командою /help

```
{
  • ok: true,
  • result:
    {
      ◦ message_id: 4,
      ◦ from:
        {
          ▪ id: 8400541323,
          ▪ is_bot: true,
          ▪ first_name: "TsNAP Dnipro",
          ▪ username: "TsNAP_Dnipro_bot"
        },
      ◦ chat:
        {
          ▪ id: -5081672182,
          ▪ title: "ЦНАП міста Дніпра",
          ▪ type: "group",
          ▪ all_members_are_administrators: true,
          ▪ accepted_gift_types:
            {
              ▪ unlimited_gifts: false,
              ▪ limited_gifts: false,
              ▪ unique_gifts: false,
              ▪ premium_subscription: false
            }
        },
      ◦ date: 1766244523,
      ◦ text: "Чатбот ЦНАП міста Дніпра створено!"
    }
}
```

Рисунок 2.12 – Відповідь при перевірці створеного чатбота

Для перевірки створеного чатбота відправимо повідомлення «Бот ЦНАП міста Дніпра створено!» через вебпереглядач в групу «ЦНАП міста Дніпра».

Формат повідомлення наступний:

`https://api.telegram.org/bot<ТОКЕН>/sendMessage?chat_id=<ID_ЧАТА>&text=Бот ЦНАП міста Дніпра створено!`

Де: <ID_ЧАТА> – ідентифікатор групи.

В результаті отримаємо таку відповідь – рисунок 2.13.

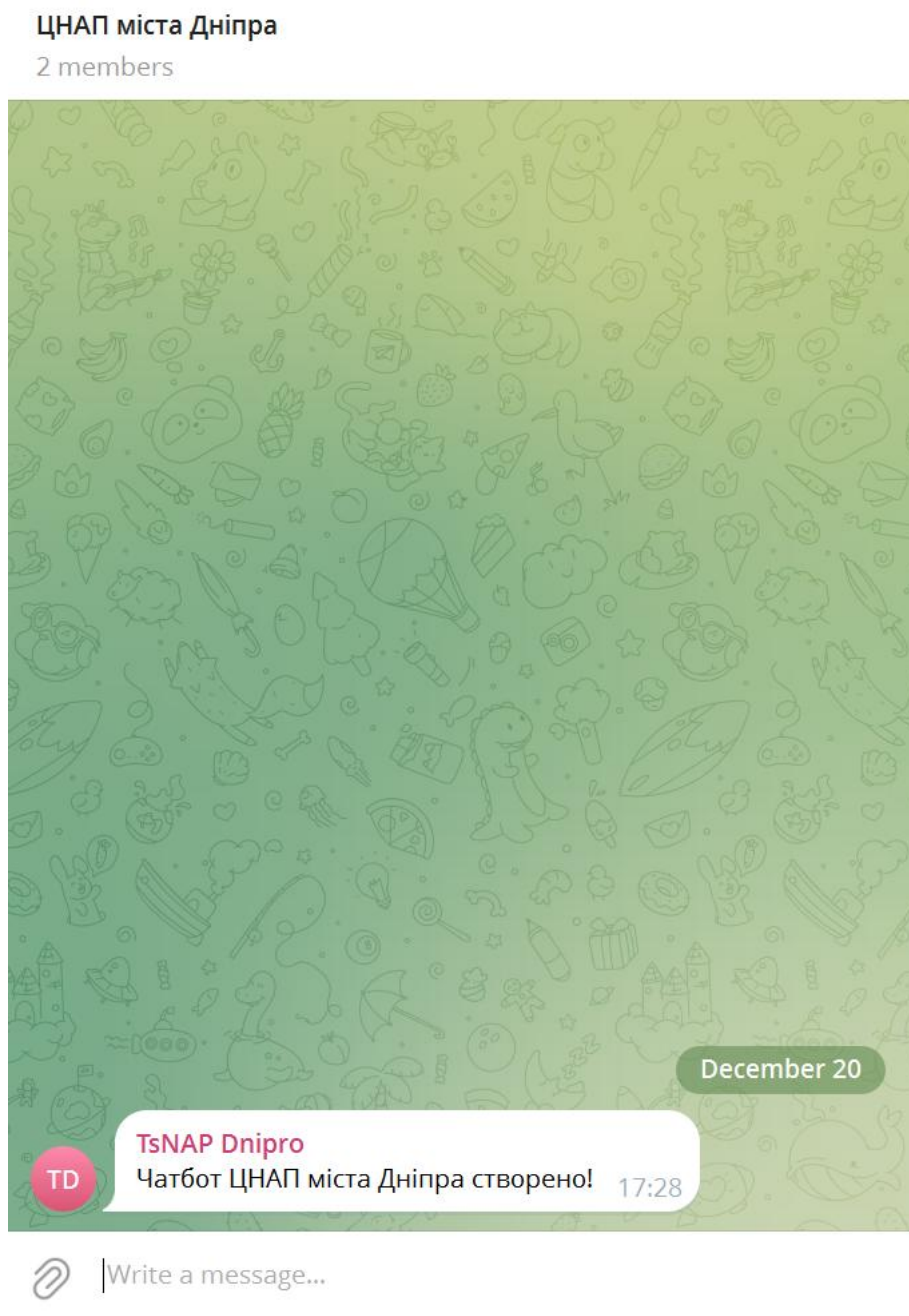


Рисунок 2.13 – Відповідь при перевірці створеного чатбота

2.3 Особливості створення програмного забезпечення ЦНАП міста Дніпра

2.3.1 Особливості створення чатбота

Раніше було визначено, що бот для Центру надання адміністративних послуг розробляється в месенджері Telegram, який частіше застосовується в Україні і є одним з найзахищених [18].

Telegram захищає повідомлення шифруванням end-to-end, має велику кількість видів захисту для забезпечення конфіденціальності користувачів.

Для розробки програми чатбота застосуємо універсальну мову для програмування – Python та безкоштовне інтегроване середовища розробки (IDE) Visual Studio Code [5, 8].

Вибір фреймворка для розробки чатбота на Python залежить від складності проєкту і рівня знань розробника. Відомі прості фреймворки як `pyTelegramBotAPI` або `python-telegram-bot`. Ці фреймворки дозволяють швидко розробляти прості боти, але мають обмежений функціонал.

Тому для більш складного проєкту чатботу TsNAP Dnipro використовуємо більш потужний фреймворк, такий як `aiogram`. Цей фреймворк пропонує функції з асинхронної роботи, підтримка кількох платформ, інтеграції з іншими службами та інструментами.

Розглянемо особливості фреймворк `aiogram` для застосування в чатботі TsNAP Dnipro.

Бот запускається по команді `/start`.

Ідентифікатор чатботу TsNAP Dnipro, який отримано при його створенні, розмістимо в окремому файлі `config.py`.

Створимо клавіатуру для головного меню чатбота (рисунок 2.14). Воно дозволяє користувача ЦНАП ознайомитись, отримати консультацію і записатись на прийом до спеціалістів ЦНАП міста Дніпра для отримання адміністративних послуг. При натисканні на відповідні кнопки користувач отримує наступну інформацію:

1. Відділи (перелік і розклад їх роботи);

2. Послуги (перелік адміністративних послуг ЦНАП міста Дніпра);
3. Мої записи (актуальні записи відвідувача ЦНАП до відділів);
4. Записатись до відділу (запис на прийом до відділу);
5. Задати питання (консультація віртуального спеціаліста ЦНАП).

Клавіатура буде мати два рядки: три кнопки в першому і дві кнопки в другому (приклад 2.1).

В якості клавіатури застосуємо статичну текстову, яка у чаті пропонує користувачу готові кнопки з текстом. Натискання на кнопки надсилає цей текст як звичайне повідомлення боту.

Клавіатуру реалізуємо за допомогою класу ReplyKeyboardMarkup, який використовується для створення об'єкта клавіатури, який потім передається боту для відображення кнопок у чаті.

Для зменшення кнопок до об'єкта клавіатури введемо додатковий параметр `resize_keyboard=True`, щоб кнопки адаптувалися під розмір інтерфейсу та не займали більшу частину екрана.

Повністю цей фрагмент програми представлено в Додатку А.

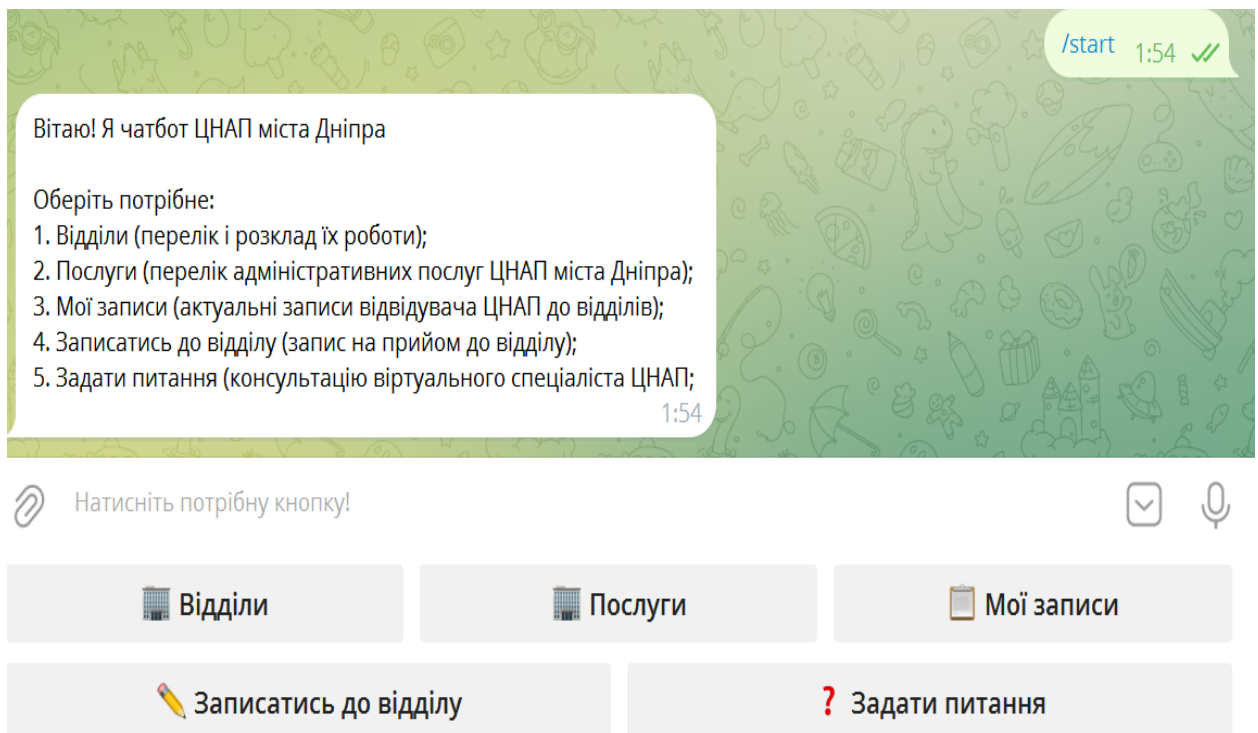


Рисунок 2.14 – Перша сторінка чатбота ЦНАП міста Дніпра

Приклад 2.1

```

# Імпорт конфігурації
from config import (
    BOT_TOKEN, AVAILABLE_DEPARTMENTS, AVAILABLE_SERVICES, )
bot = Bot(token=TOKEN)

# Dispatcher
dp = Dispatcher()

def get_main_menu():
    return ReplyKeyboardMarkup(
        keyboard=[
            [KeyboardButton(text="🏢 Відділи"),
             KeyboardButton(text="🏢 Послуги"),
             KeyboardButton(text="📁 Мої записи)],
            [KeyboardButton(text="👉 Записатись до відділу"),
             KeyboardButton(text=" ? Задати питання")]
        ],
        resize_keyboard=True,
        input_field_placeholder="Виберіть потрібну команду!"
    )

# Обробники
@dp.message(Command("start"))
async def start(message: Message):
    await message.answer(
        "Вітаю! Я чатбот ЦНАП міста Дніпра\n\n"
        "Оберіть потрібне:\n"
        "1. Відділи (перелік і розклад їх роботи);\n"
        "2. Послуги (перелік адміністративних послуг ЦНАП міста Дніпра);\n"
    )

```

"3. Мої записи (актуальні записи відвідувача ЦНАП до відділів);\n"

"4. Записатись до відділу (запис на прийом до відділу);\n"

"5. Задати питання (консультацію віртуального спеціаліста ЦНАП);\n",
reply_markup=get_main_menu())

2.3.2 Особливості підключення бази даних MongoDB до чатбота

Для розробки комп'ютерної системи ЦНАП міста Дніпра важливим є вибір бази даних, яка забезпечить надійне зберігання великих файлів та зручний доступ до них для подальшого аналізу та застосування.

Було розглянуто дві популярні бази даних: MongoDB та PostgreSQL. Головною перевагою MongoDB є підтримка горизонтального масштабування, що дозволяє додавати сервери для збільшення потужностей бази даних без зміни основної архітектури системи [15]. Це являється важливим для розробляємої системи у випадку збільшення обсягів даних або зростання кількості запитів користувачів системи. Тому після аналізу всіх переваг і недоліків було прийнято рішення про застосування MongoDB.

MongoDB є документно-орієнтованою базою даних, яка зберігає дані у форматі JSON-подібних документів. Це забезпечує високу гнучкість при роботі з різними типами даних, що є вагомим аргументом для системи ЦНАП міста Дніпра, де структура даних може змінюватися.

Для інсталяції MongoDB з офіційного сайту за посиланням <https://www.mongodb.com/try/download/community> завантажимо MongoDB Сервер і встановимо 64-розрядну версію для Windows.

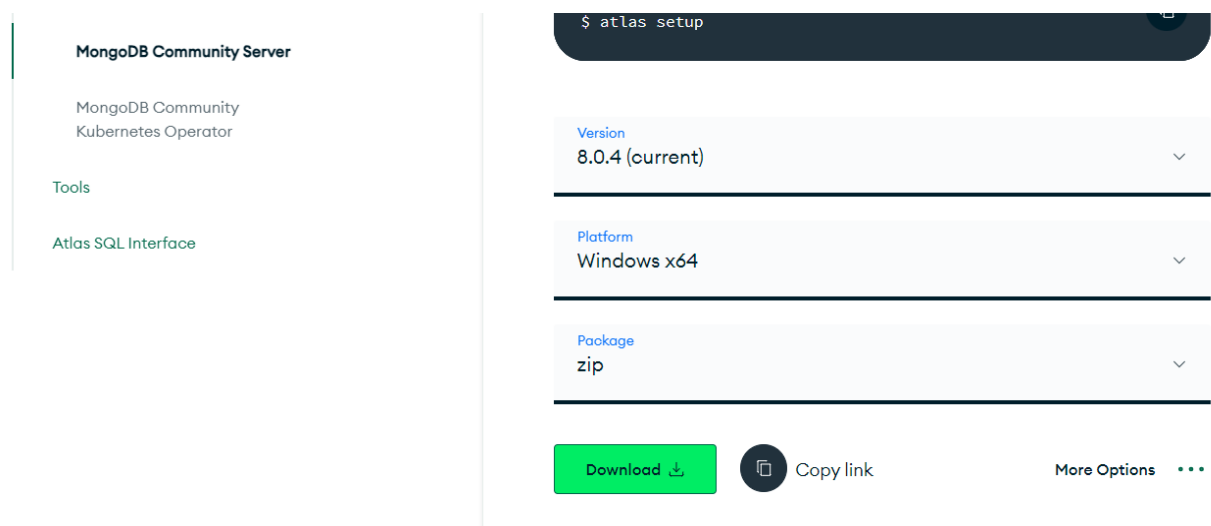


Рисунок 2.15 – Завантаження пакету MongoDB

На рисунку 2.16 наведений перелік файлів архіву.

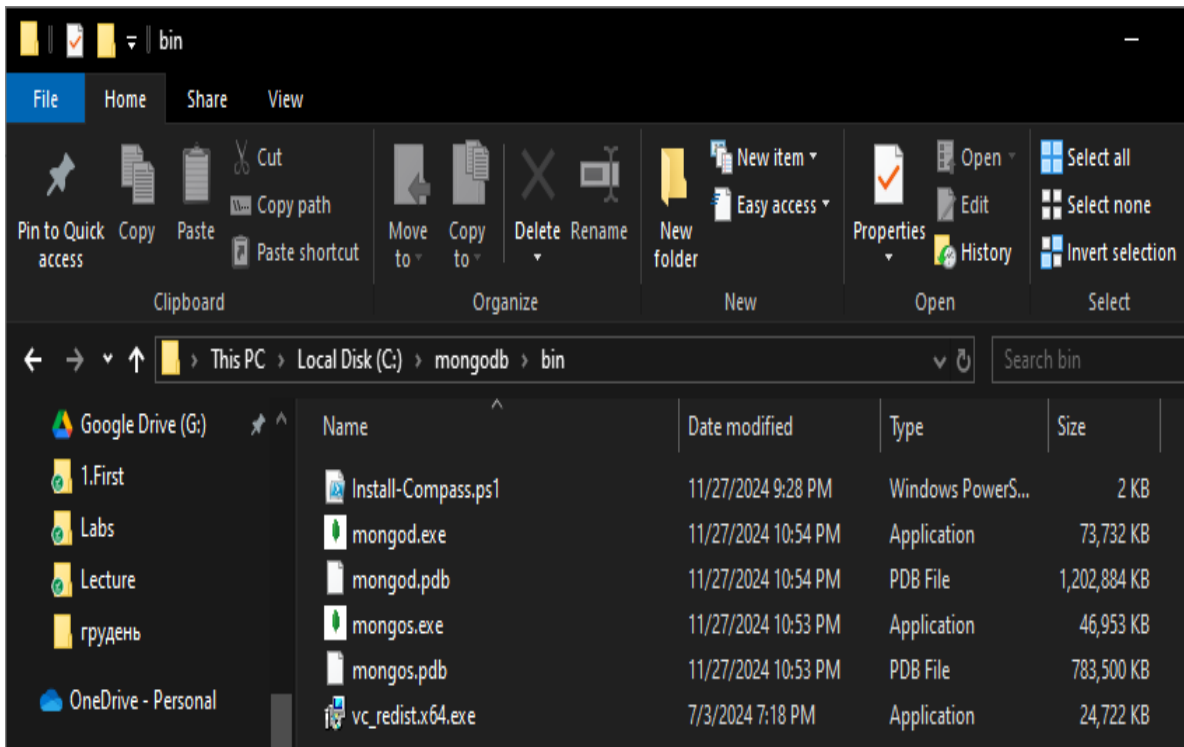


Рисунок 2.16 – Вміст архіву MongoDB

Після розпакування архіву потрібно створити директорію в якій будуть зберігатися БД. В операційній системі Windows, за замовчуванням БД зберігаються по шляху `C:\data\db`.

Після створення каталогу для зберігання БД запусимо сервер MongoDB. Після вдалого запуску серверу можна взаємодіяти з БД.

Для підключення до бази даних MongoDB, використовуємо бібліотеку `pymongo`.

Якщо вона не встановлена, її встановлюють за допомогою `pip install pymongo`.

Далі в приклад 2.2 мовою Python наведено фрагмент програми підключення до бази даних.

Приклад 2.2

```
import pymongo
```

```
# Підключення до бази даних
```

```

connection = pymongo.MongoClient("mongodb://your_host:your_port/")

# Вибір бази даних

db = connection["your_database_name"]

# Вибір колекції (таблиці у термінах реляційних баз даних)

collection = db["your_collection_name"]

...

# Після завершення роботи з базою даних, необхідно закрити з'єднання.

# Закриття з'єднання

connection.close()

```

Для успішного підключення до бази даних ЦНАП міста Дніпра замінимо `your_host`, `your_port`, `your_username`, `your_password`, `your_database_name`, `your_collection_name` та інші параметри на відповідні значення:

```

your_host = "mongodb://localhost:27017/"

your_port = 8080

your_username = "my_username"

your_password = "my_password"

your_database_name = "db_tsnap_dnipro"

your_collection_name = "users"

```

Оновлення даних у MongoDB виконується за допомогою наступних команд: `db.collection.updateOne(<filter>, <update>, <options>)`

`db.collection.updateMany(<filter>, <update>, <options>)`

`db.collection.replaceOne(<filter>, <update>, <options>).`

Основними операціями видалення є `deleteMany()` та `deleteOne()`.

2.3.3 Особливості підключення моделі ChatGPT до бота

При розробці чатбота ЦНАП міста Дніпра виявлено, що частина запитів користувачів має чітко регламентовану структуру (наприклад, оформлення заяви на отримання нового паспорта або його заміну, уточнення реквізитів для оплати адміністративної послуги, ознайомлення з адміністративними послугами тощо), тоді як інша частина є відкритими питаннями типу «який відділ для мого запиту краще вибрати», «як пенсіонеру, який не має смартфона, отримати новорічну допомогу», «які умови додаткової новорічної допомоги».

Для покращення сервісу чатбота запропоновано запити з подібними питаннями передавати на обробку віртуальному спеціалісту ЦНАП, тобто штучному інтелекту.

Аналіз наявних моделей ШІ показав, що найбільш оптимальним вбором з точки зору якості/фінансові витрати є модель ChatGPT від OpenAI версії 4o.

Розглянемо як її підключити для застосування в чатботі.

Для того, щоб чатбот міг звертатися до ChatGPT необхідно спочатку отримати API-ключ від OpenAI. Для цього:

- треба зареєструватись на OpenAI Platform і увійти цю сторінку;
- перейти у розділ API Keys та натиснути Create new secret key;
- скопіювати згенерований ключ, який виглядатиме приблизно так:

```
sk-proj-E26xTfwf0xI1yHНyg68_l1wwSsybTe-
AEaRlIP6HGe2z6I6Mx1EyXAGuHnOcmdj8PhY0HaWT3BlbkGKdUDK7N4_FBO
y1K7x8gGiw833pwyuM4u9C48gVBCrjVC1qmlGZL1i_8mR9v5CRjmHae2bXQA
```
- збережемо цей ключ у безпечному місці, так як він більше повторно не відображатиметься;
- установіть бібліотеки openai виконавши команду `pip install aiogram openai`.

Після цього можна переходити до написання програми чатбота.

Розробимо модуль для перевірки роботи ChatGPT при підключенні до чатбота (Приклад 2.3).

Приклад 2.3

```
import asyncio
from aiogram import Bot, Dispatcher, types
from aiogram.filters import CommandStart
from openai import AsyncOpenAI

# Конфігурація
TG_TOKEN = "TG_TOKEN"
OPENAI_KEY = "OPENAI_KEY"

# Ініціалізація ШІ і бота
client = AsyncOpenAI(api_key=OPENAI_KEY)
bot = Bot(token=TG_TOKEN)
dp = Dispatcher()

# Обробка команди /start
@dp.message(CommandStart())
async def cmd_start(message: types.Message):
    await message.answer("Вітаю! Я віртуальний спеціаліст ЦНАП міста Дніпра. Чекаю Вашого питання!")

# Обробка текстових повідомлень
@dp.message()
async def gpt_answer(message: types.Message):
    # Відправляємо повідомлення, що бот друкує відповідь»
    await bot.send_chat_action(chat_id=message.chat.id, action="typing")

    try:
        # Запрос к API OpenAI (модель gpt-4o як найшвидшу і дешевшу в 2025 році)
```

```
response = await client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "system", "content": "Ти корисний помічник."},
        {"role": "user", "content": message.text}
    ]
)

# відправляємо відповідь користувачу
await message.reply(response.choices[0].message.content)

except Exception as e:
    await message.reply(f"помилка: {e}")

async def main():
    print("Бот почав працювати ...")
    await dp.start_polling(bot)

if __name__ == "__main__":
    asyncio.run(main())
```

2.4 Розробка моделі чатбота Центра надання адміністративних послуг міста Дніпра

Для проектування функціональної моделі чатбота Центру надання адміністративних послуг (ЦНАП) міста Дніпра використовуємо методологію **IDEF0** (Integration DEFinition for Function Modeling). Цей стандарт застосовується для структурного моделювання функцій складних систем та бізнес-процесів, забезпечуючи їхнє інтегроване та прозоре розуміння.

На рисунку 2.12 представлена **контекстна діаграма** процесу (рівень A-0), яка візуалізує зовнішні інтерфейси, вхідні дані, керуючі впливи, механізми виконання та кінцеві результати розробки чат-бота.

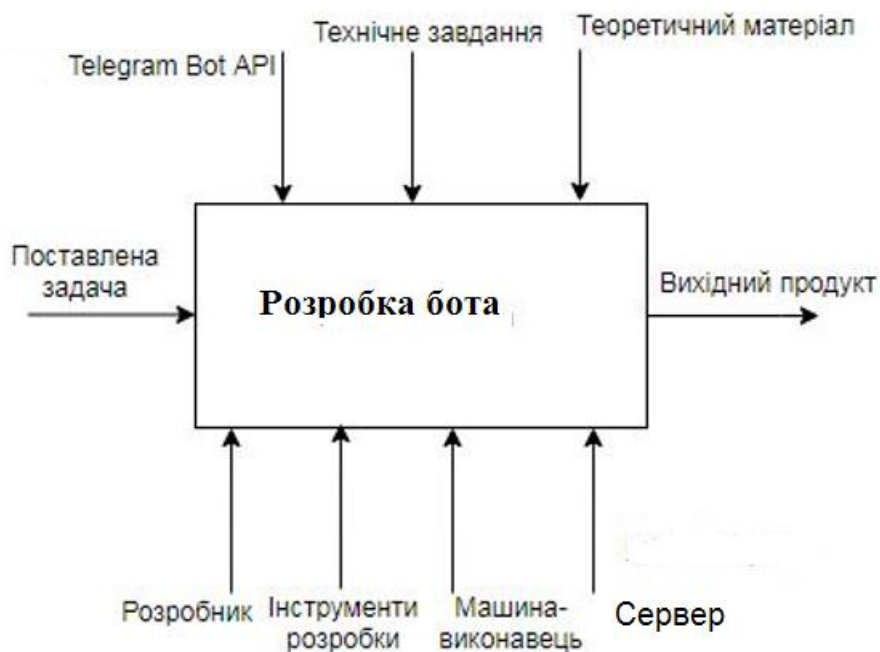


Рисунок 2.12 – Контекстна діаграма розробки чатбота ЦНАП міста Дніпра

Наступним кроком є розробка діаграми декомпозиції (рисунок 2.13), що дозволяє деталізувати основний процес на функціональні підзадачі (рисунок 2-13).

У рамках цієї декомпозиції визначено наступні ключові етапи розробки:

- реєстрація чатбота в системі Telegram: отримання і конфігурування необхідного API-токена;

- реєстрація в системі ШІ: отримання і конфігурування необхідного API-кею від ШІ та інтеграція з чатботом;
- процес розробки програмного забезпечення: реалізація логіки, функціоналу та інтеграції системи;
- тестування чат-бота: валідація коректності роботи всіх функцій та сценаріїв взаємодії.

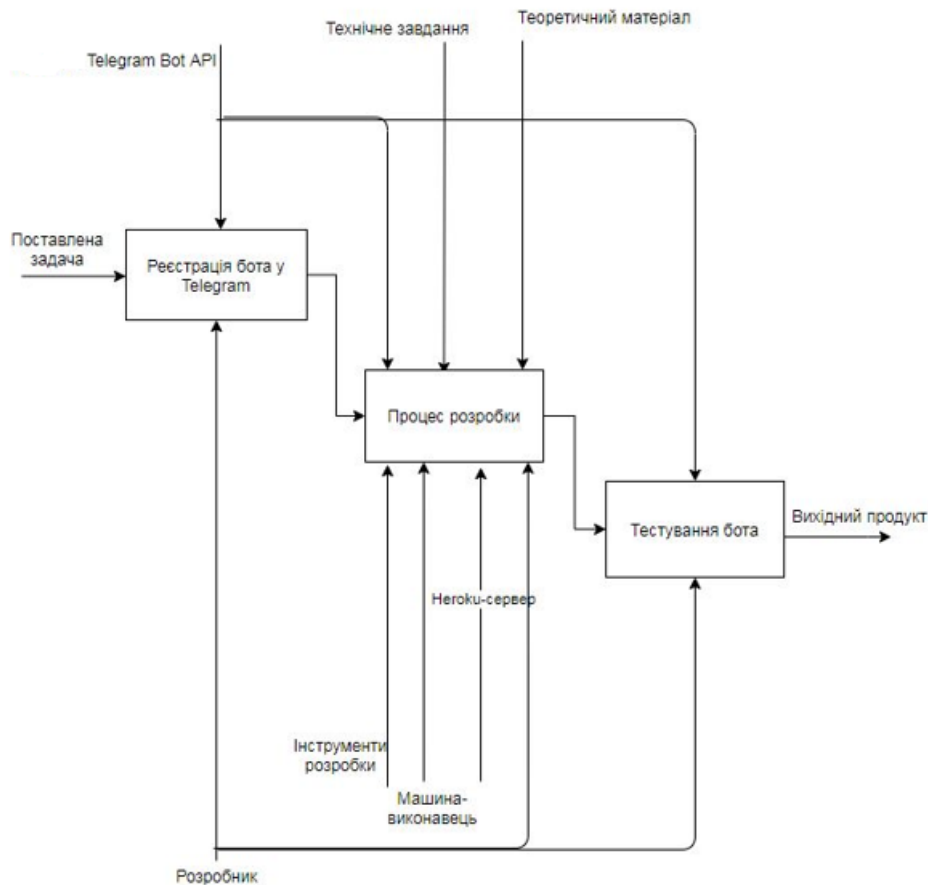


Рисунок 2.13 – Діаграма декомпозиції розробки чатбота ЦНАП міста Дніпра

Після успішного завершення всіх процедур тестування та налагодження завдання з розробки чатбота вважається виконаним.

Цей підхід забезпечує послідовність та контроль якості на кожному етапі проєкту.

Також модель чатбота можна описати як систему масового обслуговування для аналізу процесів, пов'язаних із потоками заявок (клієнтів) та їхнім обслуговуванням (наданням послуг) [12]. Застосування принципів теорії масового обслуговування дозволяє оптимізувати час відгуку системи, зменшуючи час обробки запитів та покращуючи загальну продуктивність.

У контексті чатбота ЦНАП м. Дніпра, модель СМО дозволяє оцінити ефективність роботи системи до її повноцінного впровадження.

Чатбот ЦНАП можна представити як багатоканальну СМО з чергою, що функціонує за певними правилами [12].

В цій моделі джерелом заявок (користувачами) є мешканці міста Дніпра, які потребують адміністративних послуг або консультацій.

Інтенсивність надходження заявок (λ) характеризує це джерело і є випадковою величиною, яка може змінюватися залежно від часу доби (пікові години), дня тижня чи сезону (наприклад, період подачі декларацій).

Заявки (запити) – це конкретні звернення користувачів (наприклад, "дізнатися графік роботи ЦНАП", "записатися в е-чергу", "перелік документів для послуги").

Заявки можуть мати різний пріоритет (наприклад, загальний або терміновий запис в чергу) та різну складність обробки.

Обслуговуючим пристроєм в моделі являється чатбот тобто програмне забезпечення чатбота, розгорнуте в Telegram або Viber. Він характеризується продуктивністю або інтенсивністю обслуговування (μ). Це швидкість, з якою бот обробляє один запит. Для простих запитів швидкість висока, для складних сценаріїв, коли підключається модель штучного інтелекту – нижча.

Черга – це набір запитів, які надійшли, коли чатбот був зайнятий обробкою попередніх запитів. Довжина черги може бути обмеженою або необмеженою. У випадку чатбота черга, як правило, вважається необмеженою, оскільки система може прийняти велику кількість одночасних запитів, хоча це може збільшити час очікування.

За допомогою СМО можна провести аналіз ефективності роботи системи. Моделювання СМО дозволяє розрахувати ключові показники ефективності чатбота:

- середній час очікування в черзі (скільки часу користувач чекає відповіді, якщо бот одразу не вільний);
- ймовірність відмови в обслуговуванні (якщо черга обмежена) або ймовірність того, що користувач не отримає відповіді;
- завантаження каналу обслуговування (який відсоток часу чат-бот активний ($\rho = \lambda / \mu$))

Використання моделі СМО доповнює функціональну модель IDEF0, дозволяючи перейти від якісного опису процесів до кількісної оцінки продуктивності та оптимізації архітектури системи.

Застосування імітаційного моделювання дозволяє визначити оптимальну архітектуру сервера та запобігти затримкам у роботі чатбота під час пікових навантажень?

3 СИНТЕЗ КОМП'ЮТЕРНОЇ СИСТЕМИ ЦНАП МІСТА ДНІПРА

3.1 Вимоги до системи в цілому

Комп'ютерна система Центра надання адміністративних послуг (ЦНАП) міста Дніпра повинна забезпечувати наступне:

- зменшення навантаження на персонал ЦНАП міста Дніпра за рахунок автоматизації обробки та уніфікації сценаріїв відповіді на запити користувачів;
- автоматичну обробку типових запитів користувачів на основі чатботу;
- підтримку обробки не стандартних запитів користувачів з використанням зовнішньої моделі штучного інтелекту;
- зберігання даних про запити користувачів в базі даних ЦНАП міста Дніпра з метою подальшого аналізу та вдосконалення системи;
- забезпечення конфіденційності, цілісності та доступності даних під час обробки й зберігання інформації в системі;
- захист від зловмисного доступу до інформації, яка зберігається в системі ЦНАП;
- надання статистичної інформації, необхідної співробітникам ЦНАП для роботи та вживання своєчасних заходів щодо виявлених недоліків в роботі.

3.2 Технічні вимоги до комп'ютерної системи ЦНАП міста Дніпра

3.2.1 Вимоги до надійності комп'ютерної системи

Комп'ютерна система ЦНАП міста Дніпра повинна забезпечувати надійну і безперебійну роботу.

З метою підвищення надійності роботи системи з урахуванням чатбота його програмне забезпечення розмістити в хмарі на сервері.

Максимальний час для безвідмовної роботи має становити не менше – 50000 ч, мінімальний час для безвідмовної роботи – 20000 ч.

3.2.2 Вимоги до безпеки комп'ютерної системи

Комп'ютерна система повинна із застосуванням технічних і адміністративних засобів забезпечувати високий рівень безпеки даних, які зберігаються в ЦНАП міста Дніпра, а також мати копії цих даних для відновлення внаслідок їх пошкодження або втрати.

У зв'язку з тривалим воєнним станом на території України система повинна бути підготовлена до роботи в умовах можливих перебоїв з електропостачанням та інших надзвичайних ситуацій.

3.2.3 Вимоги до функцій, виконуваних системою

Комп'ютерна система складається з п'яти взаємопов'язаних підсистем: чотири відділи ЦНАП міста Дніпра ("Правобережний", "Лівобережний", "Північний" та "Я-Ветеран") і віддалене робоче місце адміністратора по вул. Воскресенська, 9.

Вона повинна бути обладнана чатботом, з можливістю підключення зовнішньої моделі штучного інтелекту, який має функціонувати безперервно (24/7) і має зручний і зрозумілий інтерфейс.

Бот ЦНАП міста Дніпра повинен мати наступний функціонал:

- поле привітання мешканця Дніпра;
- меню вибору та інформацію про призначення його окремих пунктів;
- перелік відділів і розклад їх роботи;
- перелік адміністративних послуг ЦНАП міста Дніпра;
- актуальні записи відвідувачів ЦНАП до відділів;
- запису на прийом до відділу;
- консультації віртуального спеціаліста ЦНАП
- можливістю анонімно оцінити роботу ЦНАП міста Дніпра.

3.2.4 Вимоги до технічного забезпечення системи

Комп'ютерна система ЦНАП міста Дніпра повинна бути розгорнута та інтегрована в наявне корпоративне середовище відділів ЦНАП міста Дніпра з урахуванням існуючої мережної інфраструктури та засобів захисту інформації.

Технічне забезпечення повинно включати хмарні серверні ресурси та робочі станції достатні для роботи системи відповідно до вимог продуктивності та надійності.

Сервери на яких розгортається чатбот та баз даних повинні мати характеристиками (не нижче): процесор – 1×6 ядер, $\geq 2,6$ ГГц; оперативну пам'ять – ОЗП 16 ГБ DDR4; SSD-сховища обсягом не менше 60 ГГБ.

При виборі обладнання повинно враховуватись можливість масштабування комп'ютерної мережі, збільшення кількості терміналів у кожній локальній мережі та перенос серверів в хмару [3].

3.2.5 Вимоги до програмного забезпечення

На ПК ЦНАП міста Дніпра необхідно встановити наступні програми:

- ОС Linux або Windows 11;
- веб-переглядач Chrome або Safari або Edge.

На серверах необхідно встановити наступні програми:

- операційна система Ubuntu Server 22.04 LTS, або Windows Server 2019;
- інтерпретатор мови програмування Python версії 3.11 або новішої із засоби керування пакетами та віртуальними середовищами (pip, Python-env);
- система керування базами даних MongoDB Community Server версії 8.0 або новішої;
- бібліотеки Python aiogram 3.13.0, pymongo 4.11.3;
- бот ЦНАП міста Дніпра.

3.3 Розробка структурної схеми комп'ютерної системи

ЦНАП міста Дніпра

З урахуванням вимог описаних раніше розглянемо комп'ютерну система ЦНАП міста Дніпра, яка бути розгорнута та інтегрована в наявну систему корпоративного середовища відділів ЦНАП міста Дніпра.

Структурну схему модернізованої комп'ютерної системи ЦНАП міста Дніпра показана на рисунку 3.1. В схемі враховано застосування інтелектуального чатбота який розміщується на сервері в хмарі з метою функціонування безперервно (24/7) та сервера бази даних MongoDB.

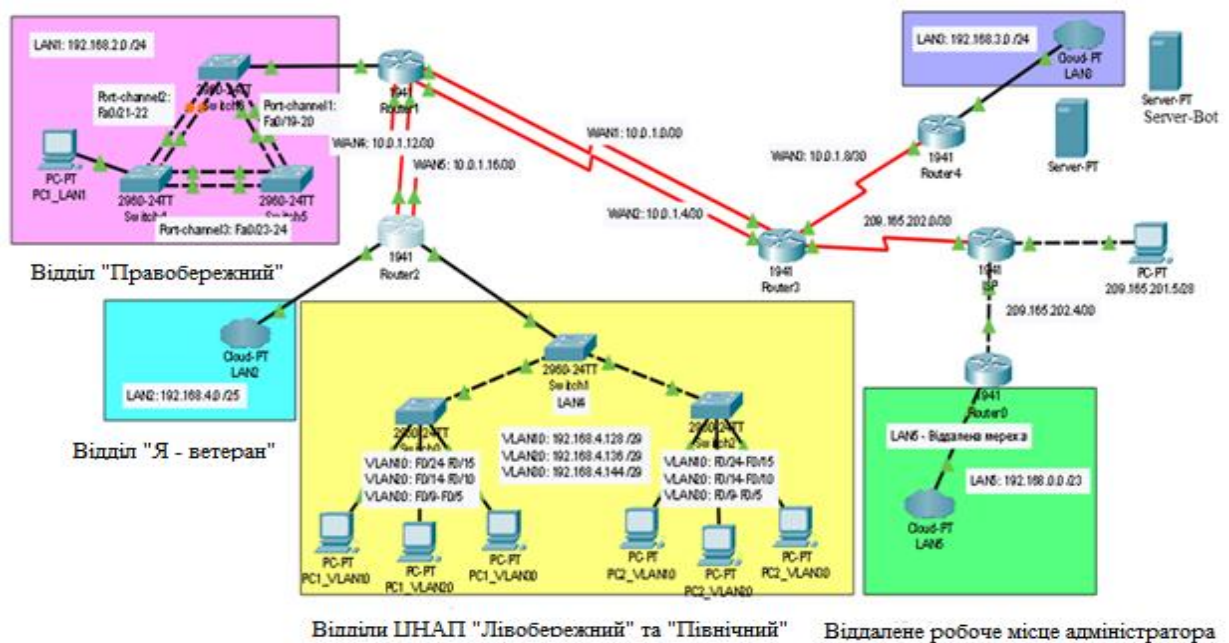


Рисунок 3.1 – Структурна схема комп'ютерної системи ЦНАП міста Дніпра

На цій схемі комп'ютерної системи: LAN 1 – локальна мережа відділу ЦНАП "Правобережний", LAN 2 – локальна мережа відділу ЦНАП "Я-Ветеран", LAN 4 – локальна мережа відділу ЦНАП "Лівобережний" та відділу ЦНАП "Північний", LAN 5 – Віддалене робоче місце адміністратора ЦНАП міста Дніпра.

4 РАЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЦНАП МІСТА ДНІПРА

4.1 Призначення і сфера застосування програми

Основне призначення розробляемого програмного забезпечення комп'ютерної системи ЦНАП міста Дніпра це забезпечення роботи чатбота.

4.2 Обґрунтування технічних характеристик програми

4.2.1 Постановка завдання на розробку програми

Для розробки програмного забезпечення комп'ютерної системи ЦНАП міста Дніпра була вибрана мова програмування Python 3.11 з бібліотеками aiogram в середовищі Visual Studio Code.

4.2.2 Структура і алгоритм функціонування програми

Комп'ютерна система повинна працювати повний робочий день, а бот управління безперервно (24/7).

4.2.3 Вибір складу програмних засобів

Для розробки програми пропонується застосовувати Visual Studio Code або коротко VS Code. Це добрий і популярний редактор коду від компанії Microsoft.

Для скорочення часу для розробки програми пропонуються застосувати бібліотеку aiogram 3.13.

4.3 Опис розробленої програми

4.3.1 Загальні відомості

Розроблена програма має модулі для отримання token ідентифікації з відповідного каталогу, формування кнопок, модулі для підключення до бази даних MongoDB, штучного інтелекту, формування повідомлень і модулі з handlers для формування відповіді на задані питання.

Текст програми реалізований мовою Python версії 3.12 в середовищі Visual Studio Code від компанії Microsoft.

4.3.2 Функціональне призначення

Програма призначена для забезпечення роботи чатбота Центра надання адміністративних послуг міста Дніпра (Додаток А).

4.3.3 Використовувані технічні засоби

Для роботи чатбота ЦНАП міста Дніпра передбачено його розміщення на хмарному сервері (рисунок 3.1).

4.3.4 Опис логічної структури

Цикл роботи чатбота включає отримання token ідентифікації з відповідного каталогу, підключення до бази даних MongoDB, штучного інтелекту, формування кнопок, модулі для формування повідомлень і модулі з handlers для формування відповіді на задані питання.

Для перегляду переліку відділів і розкладу їх роботи сформуємо наступний обробник (Приклад 4.1). Результат роботи показано на рисунку 4.1.

Повністю цей фрагмент програми представлено в Додатку А.

Приклад 4.1

Перелік відділів

```
@dp.message(F.text == "🏢 Відділи")
```

```
async def departments_list(message: Message):
```

```
    text = "🏢 **Доступні відділи:**\n\n"
```

for speciality, times in AVAILABLE_DEPARTMENTS.items():

```
text += f"✓ **{speciality}**\n Години: {' '.join(times)}\n\n"
```

```
await message.answer(text, parse_mode="Markdown")
```

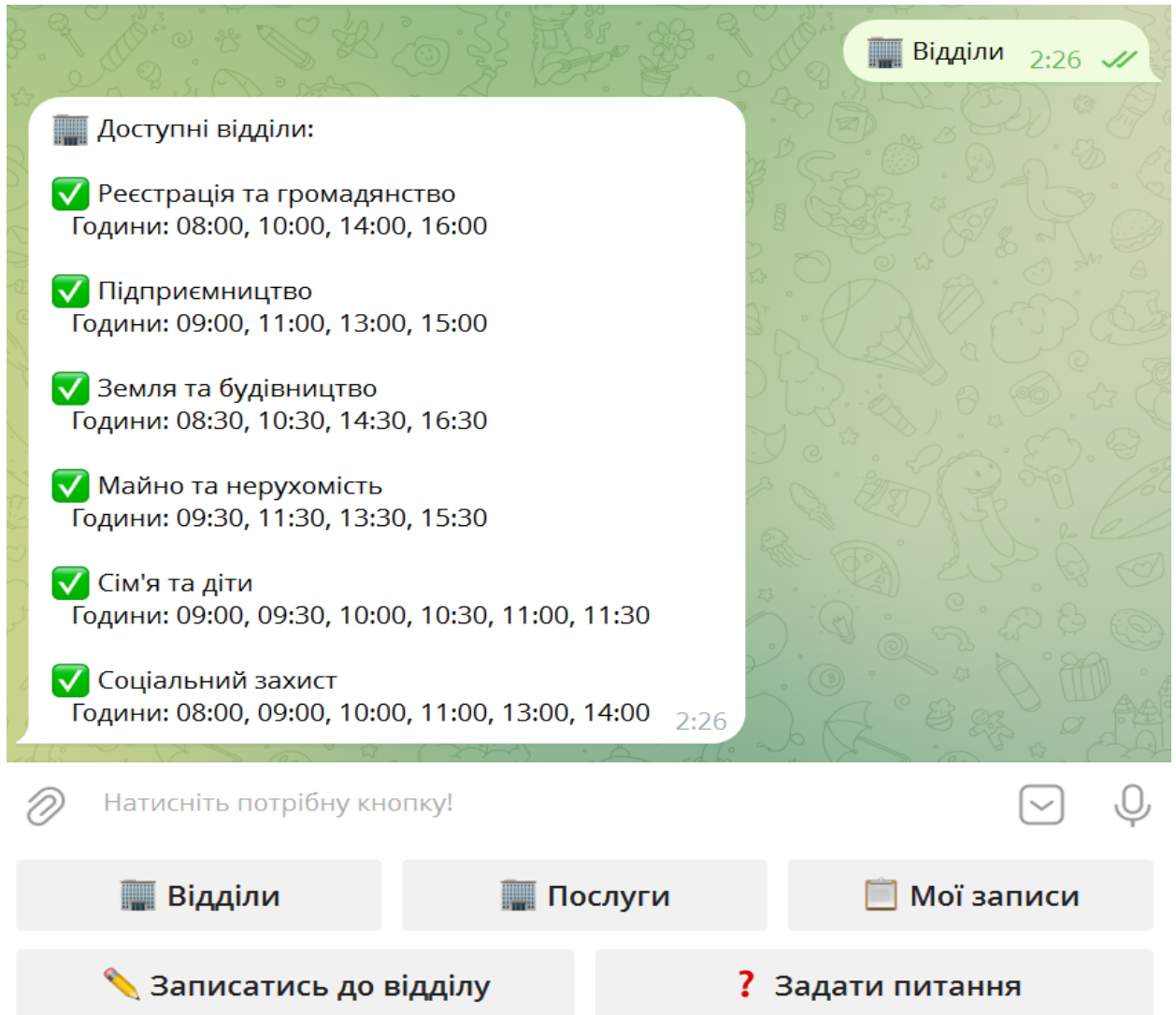


Рисунок 4.1 – Виконання команди «Відділи»

Для перегляду переліку адміністративних послуг і відділів в яких вони формуються їх роботи сформуємо наступний обробник (Приклад 4.2). Результат роботи показано на рисунку 4.2.

Повністю цей фрагмент програми представлено в Додатку А.

Приклад 4.2

Перелік адміністративних послуг

```
@dp.message(F.text == "🏢 Послуги")
```

```
async def services_list(message: Message):
```

```
    text = "🏢 **Адміністративні послуги відділів:**\n\n"
```

```
    for speciality, service in AVAILABLE_SERVICES.items():
```

```
        text += f"✅ **{speciality}**\n  Послуги: {' '.join(service)}\n\n"
```

```
    await message.answer(text, parse_mode="Markdown")
```

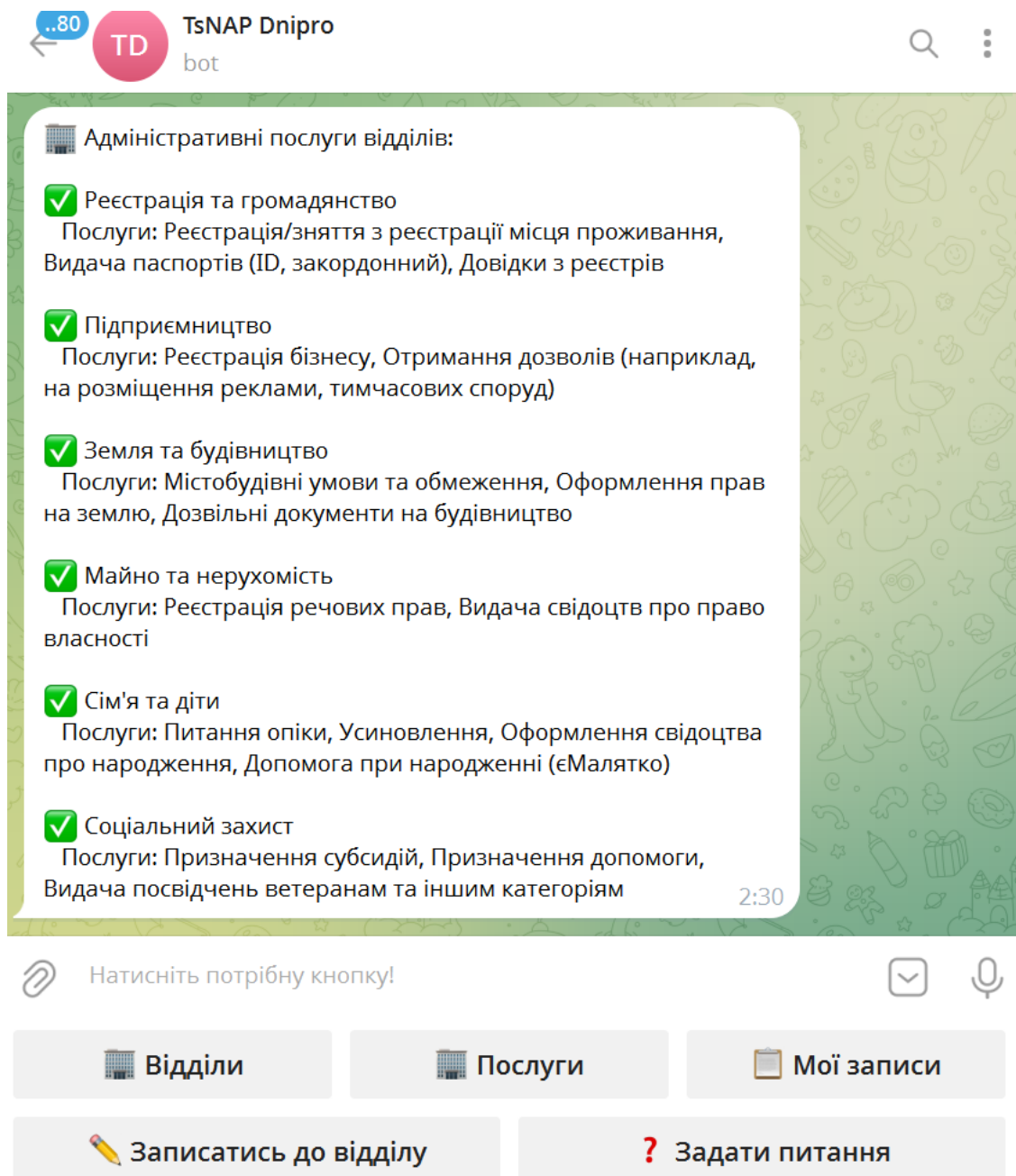


Рисунок 4.2 – Виконання команди «Послуги»

Для перегляду переліку активних записів користувача сформуємо наступний обробник (Приклад 4.3). Результат роботи показано на рисунку 4.3.

Приклад 4.3

Мої записи

@dp.message(F.text == "📅 Мої записи")

async def my_appointments(message: Message):

Спочатку видаляємо старі записи

delete_old_appointments()

rows = list(appointments_col.find({"user_id": message.from_user.id}).sort("date", 1))

if not rows:

await message.answer("📅 У Вас поки немає записів.\n\nНатисніть кнопку нижче, щоб записатись.")

return

text = "📅 **Ваші активні записи:**\n\n"

for r in rows:

text += f"✅ {r['speciality']} | {r['date']} {r['time']}\n Пацієнт: {r['user']}\n\n"

await message.answer(text, parse_mode="Markdown")

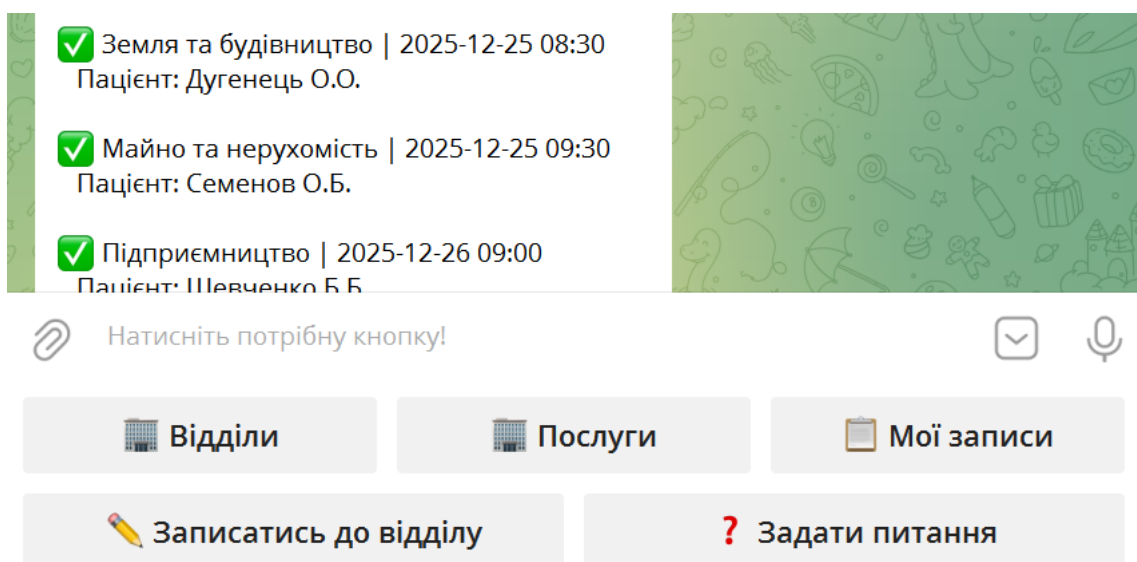


Рисунок 4.3 – Виконання команди «Мої записи»

Для запису користувача до відділу ЦНАП міста Дніпра сформуємо наступний фрагмент програми (Приклад 4.4). Результат роботи показано на рисунку 4.4.

Приклад 4.3

Введення дати

```
@dp.message(lambda msg: msg.from_user.id in user_booking_state and
              user_booking_state[msg.from_user.id].get("step") == "choose_date")
async def choose_date(message: Message):
    user_id = message.from_user.id
    date_str = message.text.strip()

    if not validate_date(date_str):
        return await message.answer("✘ Неправильний формат
дати.\n\nВикористайте: `YYYY-MM-DD`", parse_mode="Markdown")

    try:
        if datetime.strptime(date_str, "%Y-%m-%d").date() < datetime.now().date():
            return await message.answer("✘ Не можна записатись на минулу дату.")
    except:
        pass

    user_booking_state[user_id].update({"date": date_str, "step": "choose_time"})
    speciality = user_booking_state[user_id]["speciality"]
    times = AVAILABLE_DEPARTMENTS[speciality]

    time_buttons = [[KeyboardButton(text=times[i]), KeyboardButton(text=times[i+1])] if
i+1 < len(times)
                    else [KeyboardButton(text=times[i])] for i in range(0, len(times), 2)]
    time_buttons.append([KeyboardButton(text="⏪ ВАСЬ Назад")])

    await message.answer(
        f"✔ Дата: **{date_str}**\n\n🕒 **Крок 3 з 4**\n\nОберіть час:",
        parse_mode="Markdown",
```

```
reply_markup=ReplyKeyboardMarkup(keyboard=time_buttons,
resize_keyboard=True)
)
```

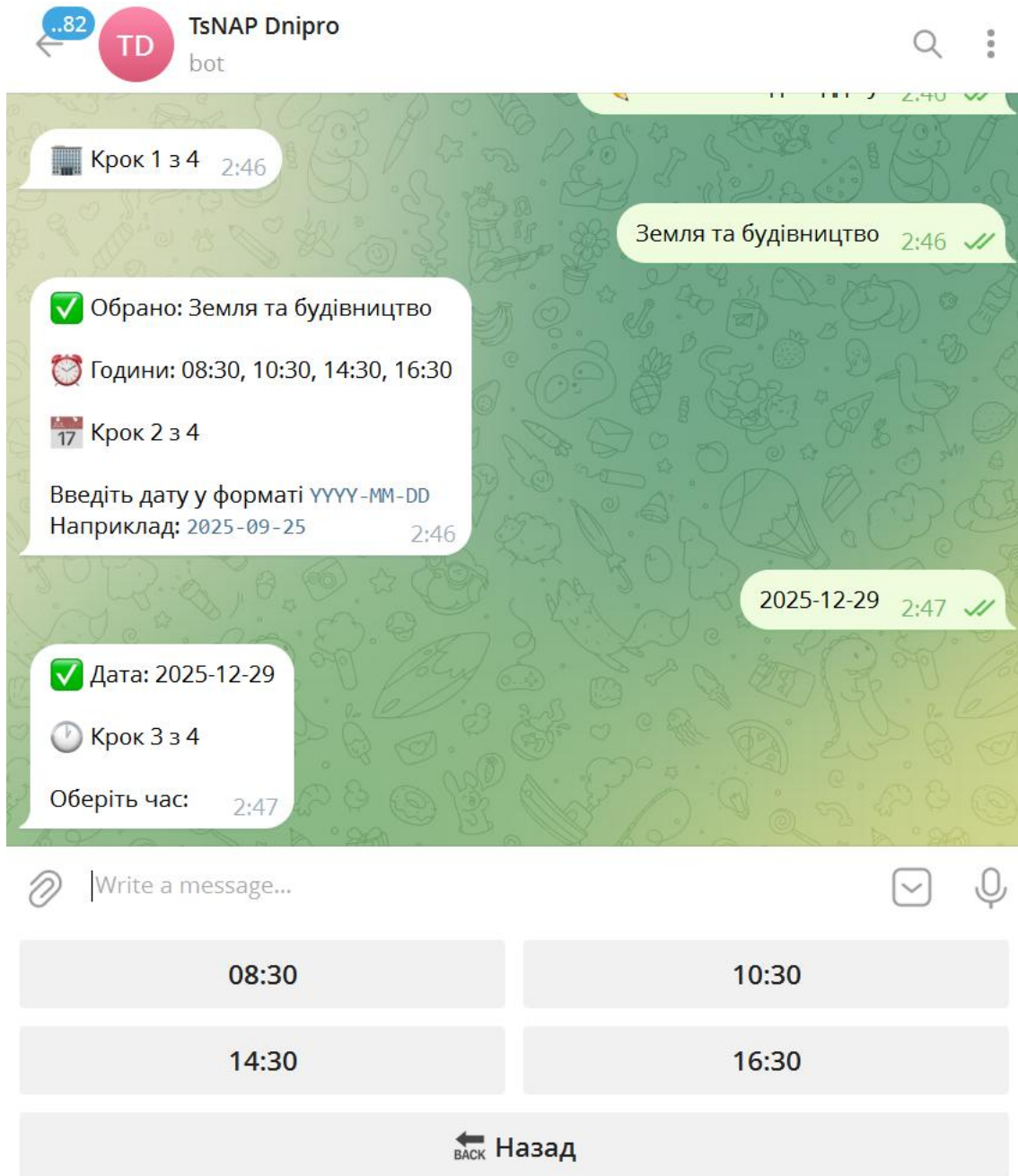


Рисунок 4.4 – Виконання команди «Записатись до відділу», Крок 3

4.3.5 Вхідні та вихідні дані

Вхідними даними являються коди кнопок сформованої клавіатури, а вихідними, повідомлення, які формуються відповідними обробниками.

5 ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ

5.1 Постановка завдання експерименту і обґрунтування методики

Завданням експерименту являється перевірка ефективності роботи програми чатбота ЦНАП міста Дніпра.

Розробимо анкету для опитування мешканців міста Дніпра. За рахунок цього опитування можна визначити думки людей стосовно розробленої програми та порівняти якість обслуговування після введення в дію розробленого чатбота.

Соціологічні опитування – це опитування, яке проводиться з метою вивчення суспільних відносин, громадської думки та поведінки людей. Вони можуть проводитися для певного виборчого округу, району, якоїсь соціальної групи.

За характером взаємодії з респондентом опитування поділяються.

Анкетування – це вид опитування, при якому респондент самостійно заповнює анкету, не взаємодіючи з опитувальником. Анкетування може бути паперовим, електронним або телефонним.

Інтерв'ю – це вид опитування, при якому респондент відповідає на запитання опитувальника в особистій бесіді. Інтерв'ю може бути структурованим, напівструктурованим або неструктурованим.

За обсягом вибірки опитування поділяються.

Масові опитування – це опитування, у якому бере участь велика кількість респондентів. Масові опитування зазвичай проводяться з метою вивчення громадської думки з актуальних питань.

Вибіркові опитування – це опитування, у якому бере участь обмежена кількість респондентів, що репрезентативно представляють цільову аудиторію. Вибіркові опитування зазвичай проводяться з метою вивчення поведінки споживачів, оцінки якості продукції або послуг тощо.

За способом проведення опитування поділяються.

Особисті опитування – це опитування, яке проводиться віч-на-віч з респондентом. Особисті опитування можуть бути ефективними для отримання глибоких і розгорнутих відповідей.

Поштові опитування – це опитування, при якому анкети надсилаються респондентам поштою. Поштові опитування є ефективним способом охоплення великої кількості респондентів, але вони мають низький відсоток повернення.

Телефонні опитування – це опитування, яке проводиться по телефону. Телефонні опитування є відносно недорогим і швидким способом отримання інформації від респондентів, але вони можуть бути обмежені в тривалості та складності питань.

Онлайн-опитування – це опитування, яке проводиться за допомогою Інтернету. Онлайн-опитування є ефективним способом охоплення великої кількості респондентів і отримання швидких результатів.

Головним інструментом виконання соціологічного опитування являється анкета.

Анкету рекомендується складати відповідно трьом станам респондента: налаштування, досягнення поставленої цілі, розслаблення.

Анкета зазвичай включає такі види питань:

- за структурою: закриті, напівзакриті, відкриті;
- за змістом: питання про факти поведінки, питання про факти свідомості, -демографічні характеристики респондента (вік, стать, фах тощо);
- за формою: табличні, графічні, шкальні, прямі та непрямі тощо;
- за функціями: основні та службові.

Часто анкета передбачає наявність вступу, контактної частини, основного блоку питань та заключного блоку. Вступ в анкеті повинен інформувати мешканця про напрям опитування, організацію, що проводить, особливості заповнення анкети, про анонімність та іншу інформацію.

5.2 Вимоги до експерименту

Під час проведення експерименту будемо задавати такі питання:

- щодо задоволення обслуговуванням ЦНАП міста Дніпра;
- думка щодо введення боту у систему комунікацій ЦНАП міста Дніпра;
- проведення опитування щодо задоволення обслуговуванням ЦНАП міста Дніпра після запровадження боту;

Далі проведемо аналіз результатів опитування та розрахунок ефективності обслуговування користувачів чатбота.

5.3 Програма експерименту

Експеримент проводиться методом безпосередньої оцінювання з метою виявлення найкращого показника ефективності.

Під час експерименту виконаємо аналіз результатів опитування як було до введення чатбота ЦНАП міста Дніпра і після його введення.

Оцінка буде бальна, від 1 до 10 балів для однієї людини. Перше опитування буде тривати 24 години. Посилати питання будемо у різні соціальні мережі та пошти.

Перелік питань наступний:

- чи був для Вас зрозумілий інтерфейс чатбота?
- чи був для Вас зручний інтерфейс чатбота?
- чи була вирішено питання при зверненні до ЦНАП міста Дніпра при застосуванні чатбота?
- чи достатньо швидко працює чатбот?

5.4 Результати експерименту

Під час дослідження було сформовано список питань, а також проведені онлайн опитування.

Результати першого опитування стосовно чатбота ЦНАП міста Дніпра, занесемо в таблицю 5.1.

Таблиця 5.1 – Результати першого опитування

№	Запитання	Середня оцінка за одне питання, балів
1.	Чи був для Вас зрозумілий інтерфейс чатбота?	8,2
2.	Чи був для Вас зручний інтерфейс чатбота?	8,1
3.	Чи було вирішено питання при зверненні до ЦНАП міста Дніпра при застосуванні чатбота?	7,5
4.	Чи достатньо швидко працює чатбот?	6,5
	Середня оцінка першого опитування	7,6

За формулою 5.1 розрахуємо середньо арифметичну оцінку.

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{\sum x}{n} \quad 5.1$$

Рівень якості програми розділимо на три ступеня: високий, середній, низький.

Високий ступень оцінюється середнім балом від 9 до 10 оцінкою;

Середній ступень оцінюється середнім балом від 6 до 8 оцінкою;

Низький ступень оцінюється середнім балом від 1 до 5 оцінкою;

В першому опитуванні прийняли участь 56 мешканців Дніпра, з них:

- 14 голосів набрало високий ступень оцінки якості;
- 27 голосів набрало середній ступень оцінки якості;
- 15 голосів набрало низький ступень оцінки якості;

Розрахуємо у відсотках:

- високий ступень = $14 \cdot 100 / 56 = 25,0\%$
- середній ступень = $27 \cdot 100 / 56 = 48,2\%$
- низький ступень = $15 \cdot 100 / 56 = 26,8\%$

Друге опитування проведемо після першого через 2 тижня. Результати другого опитування стосовно чатбота ЦНАП міста Дніпра, занесемо в таблицю 5.2.

Таблиця 5.2 – Результати другого опитування

№	Запитання	Середня оцінка за одне питання, балів
1.	Чи був для Вас зрозумілий інтерфейс чатбота?	8,5
2.	Чи був для Вас зручний інтерфейс чатбота?	8,4
3.	Чи було вирішено питання при зверненні до ЦНАП міста Дніпра при застосуванні чатбота?	7,9
4.	Чи достатньо швидко працює чатбот?	7,5
	Середня оцінка другого опитування	8,1

В другому опитуванні прийняли участь 245 мешканців Дніпра, результати порівняння з першим опитуванням відображені в таблиці 5.3.

Таблиця 5.3 – Порівняння результатів опитування

№	Ступень якості	1-е опитування		2-е опитування		Збільшення порівняно з 1-м %
		голосів	%	голосів	%	
1.	Високий	14	25	102	41.63	16.6
2.	Середній	27	48.2	128	52.25	4.0
3.	Низький	15	26.8	15	6.12	-20.7
	Всього	56	100	245	100	

Кількість голосів при другому опитуванню збільшилась на 189.

Згідно результату експерименту опитування та розрахунків, показник задоволеності обслуговування ЦНАП міста Дніпра виріс.

Оцінка високого та середнього ступеня якості збільшилась на 20,6%. Це означає що мешканцям міста Дніпра сподобався новий вид обслуговування.

ВИСНОВКИ

Кваліфікаційна робота є завершеною науковою роботою, в якій вирішена задача – обґрунтування структури комп'ютерної системи Центра надання адміністративних послуг міста Дніпра з інтегрованим інтелектуальним чатботом.

Основні висновки і результати роботи полягають в наступному:

- проведено аналіз сфери надання адміністративних послуг, умов застосування системи та огляд відповідних підрозділів;
- виконано аналіз розроблених ботів в сфері надання адміністративних послуг;
- вивчено особливості розробки ботів мовою Python;
- розроблено структуру комп'ютерної системи Центра надання адміністративних послуг міста Дніпра;
- розроблено програмне забезпечення системи включаючи і інтелектуальний бот ЦНАП міста Дніпра;
- проведені експериментальні дослідження і обґрунтована структура модернізованої комп'ютерної системи Центра надання адміністративних послуг міста Дніпра.

ПЕРЕЛІК ПОСИЛАНЬ

1. Д ISO 5807:1985 Information processing. Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts. <https://www.iso.org/standard/11955.html>
2. ДСТУ 3008-2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. – К.: Держстандарт, 2015. – 37 с.
3. ДСТУ ISO 5807:2016 Оброблення інформації. Символи та угоди щодо документації стосовно даних, програм та системних блок-схем, схем мережевих програм та схем системних ресурсів (ISO 5807:1985, IDT).
4. Словник термінів ІТ і комп'ютерної інженерії / В.В. Гнатушенко, Г.М. Коротенко, В.І. Олевський [та ін.]; за ред. В.В. Гнатушенка, Г.М. Коротенка, Л.І. Цвіркуна. - Дніпро : НТУ «ДП», 2025. – 709 с.
5. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування: навч. посіб. / А.П. Крєневич. – К.; ВПЦ «Київський університет», 2020. – 152 с.
6. <https://ww.viber.com/erckp-vbot>
7. <https://dniprorada.gov.ua/uk/articles/item/47343/zvernennya-za-kilka-hvilin-u-misti-zaprasuyuvav-chat-bot-kontakt-centru-dnipra>
8. <https://www.python.org/doc/>
9. Журавська І. М. Проектування та монтаж локальних комп'ютерних мереж :[навчальний посібник] / І. М. Журавська. – Миколаїв : Видавництво ЧДУ ім. Петра Могили, 2016. – 396 с.
10. Цвіркун Л.І. Інженерна та комп'ютерна графіка. AutoCAD : навч. посіб. / Л.І. Цвіркун, Л.В. Бешта ; під. заг. ред. Л.І. Цвіркуна ; М-во освіти і науки України, НТУ “Дніпровська політехніка”. – Дніпро: НТУ “ДП” , 2018. – 209 с. – ISBN 978-966-350-663-0.
11. Цвіркун Л.І. Розробка програмного забезпечення комп'ютерних систем. Програмування: навч. посіб. [Електронний ресурс] / Л.І. Цвіркун,

А.А. Євстігнєєва, Я.В. Панферова ; під заг. ред. проф. Л.І. Цвіркун ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – 1 електрон. опт.

12. Теорія систем масового обслуговування : навч. посібник / А. Л. Литвинов ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2018. – 141 с.

13. AI Chatbots for Business: Transforming Your Customer Experience. Build Enterprise-Grade Customer Experience AI Agents | Jio Haptik. URL: https://www.haptik.ai/knowledge-center/ai-chatbot-for-business?utm_source=chatgpt.com (дата звернення: 10.12.2025).

14. Global Market Insights. Chatbot Market Size Report, 2018 – 2024 – [Електронний ресурс] – <https://www.gminsights.com/industry-analysis/chatbot-market> (Дата звернення: 14.11.2025).

15. MongoDB vs PostgreSQL: Which to Choose For Your Database Solutions. EDB. URL: <https://www.enterprisedb.com/choosing-mongodb-postgresql-cloud-database-solutions-guide> (дата звернення: 10.12.2025).

16. Tsvirkun, L., Myronov, Y. Challenges and Specificities of Adopting Continuous Integration within Scalable Cloud Environments. International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2023.

17. Миронов Ю., Цвіркун Л. (2024). Аналіз методів структурної оптимізації процесів неперервної інтеграції. Information Technology: Computer Science, Software Engineering and Cyber Security, 3, 133–139. <https://doi.org/10.32782/IT/2024-3-14>.

18. Цвіркун Л.І. Використання месенджерів як системи оповіщення користувачів локальних систем домашньої автоматизації. / Л.І. Цвіркун, Л.В. Бешта, Ю.А. Миронов // Системні технології. Зб. наук. пр. НМЕТАУ. – 2021. – № 4. – с. 95–101.

Додаток А

Текст програми

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
КОМП'ЮТЕРНОЇ СИСТЕМИ
ЦНАП МІСТА ДНІПРА

Текст програми

804.02070743.25007-01 12 01

Листів 16

2025

АНОТАЦІЯ

Дана програма містить частину програмного коду чатбота комп'ютерної системи ЦНАП міста Дніпра.

Застосована мова програмування Python 3.12 з бібліотекою aiogram 3.2 в середовищі Visual Studio Code.

ЗМІСТ

	Стр.
1. Імпорт бібліотек	4
2. Імпорт конфігурації	4
3. Функції	5
4. Обробники	7
5. Інтелектуальний модуль	15

```

# TsNAP_Dnipro
# Імпорт бібліотек
import asyncio
from datetime import datetime, timezone, timedelta
from aiogram import Bot, Dispatcher, F
from aiogram.filters import Command, CommandObject
from aiogram.types import Message, ReplyKeyboardMarkup, KeyboardButton,
ReplyKeyboardRemove
from pymongo import MongoClient, ASCENDING
from aiohttp import web
import re

# Імпорт конфігурації
from config import (
    BOT_TOKEN, MONGO_URI, DB_NAME,
    ADMIN_IDS, ADMIN_PASSWORD, AVAILABLE_DEPARTMENTS,
    AVAILABLE_SERVICES, DELETE_AFTER_DAYS
)

# MongoDB
User = MongoClient(MONGO_URI)
db = User[DB_NAME]
appointments_col = db["appointments"]
logs_col = db["logs"]
modes_col = db["modes"]
logs_col.create_index([("timestamp", ASCENDING)])

# Стан користувачів
user_booking_state = {}

bot = Bot(BOT_TOKEN)
dp = Dispatcher()

# ===== Helpers =====
def is_admin(uid: int) -> bool:

```

```

return uid in ADMIN_IDS

def get_watch_mode(uid: int) -> bool:
    entry = modes_col.find_one({"admin_id": uid})
    return entry.get("watching", False) if entry else False

def set_watch_mode(uid: int, state: bool):
    modes_col.update_one(
        {"admin_id": uid},
        {"$set": {"admin_id": uid, "watching": state, "since": datetime.now(timezone.utc) if
state else None}},
        upsert=True
    )

def get_main_menu():
    return ReplyKeyboardMarkup(
        keyboard=[
            [KeyboardButton(text="🏢 Відділи"),
            KeyboardButton(text="🏢 Послуги"),
            KeyboardButton(text="📁 Мої записи)],
            [KeyboardButton(text="➡ Записатись до відділу"),
            KeyboardButton(text="❓ Задати питання")]
        ],
        resize_keyboard=True,
        input_field_placeholder="Виберіть потрібну команду!"
    )

def get_departments_menu():
    return ReplyKeyboardMarkup(
        keyboard=[
            [KeyboardButton(text="Реєстрація та громадянство"),
            KeyboardButton(text="Підприємництво"),
            KeyboardButton(text="Земля та будівництво)],
            [KeyboardButton(text="Майно та нерухомість"),

```

```

        KeyboardButton(text="Сім'я та діти"),
        KeyboardButton(text="Соціальний захист"]),
    [KeyboardButton(text="⬅️ Назад")]
    ],
    resize_keyboard=True,
    input_field_placeholder="Виберіть потрібний відділ!"
)

```

```
def validate_date(date_str: str) -> bool:
```

```

    try:
        datetime.strptime(date_str, "%Y-%m-%d")
        return True
    except:
        return False

```

```
def validate_time(time_str: str) -> bool:
```

```

    return bool(re.match(r'^([0-1]?[0-9]|2[0-3]):[0-5][0-9]$', time_str))

```

```
def delete_old_appointments():
```

```

    """Видаляє записи старіші за DELETE_AFTER_DAYS днів"""
    cutoff_date = (datetime.now() -
timedelta(days=DELETE_AFTER_DAYS)).strftime("%Y-%m-%d")
    result = appointments_col.delete_many({"date": {"$lt": cutoff_date}})
    if result.deleted_count > 0:
        print(f"Видалено {result.deleted_count} старих записів (до {cutoff_date})")
    return result.deleted_count

```

```
# ===== Фонове завдання для очистки =====
```

```
async def cleanup_task():
```

```

    """Запускається кожні 24 години для видалення старих записів"""
    while True:
        await asyncio.sleep(86400) # 24 години
        delete_old_appointments()

```

```
# Обробники
```

```
@dp.message(Command("start"))
```

```
async def start(message: Message):
```

```
    await message.answer(
```

```
        "Вітаю! Я чатбот ЦНАП міста Дніпра\n\n"
```

```
        "Оберіть потрібне:\n"
```

```
        "1. Відділи (перелік і розклад їх роботи);\n"
```

```
        "2. Послуги (перелік адміністративних послуг ЦНАП міста Дніпра);\n"
```

```
        "3. Мої записи (актуальні записи відвідувача ЦНАП до відділів);\n"
```

```
        "4. Записатись до відділу (запис на прийом до відділу);\n"
```

```
        "5. Задати питання (консультацію віртуального спеціаліста ЦНАП;\n",
```

```
        reply_markup=get_main_menu()
```

```
    )
```

```
# Перелік відділів
```

```
@dp.message(F.text == "🏢 Відділи")
```

```
async def departments_list(message: Message):
```

```
    text = "🏢 **Доступні відділи:**\n\n"
```

```
    for speciality, times in AVAILABLE_DEPARTMENTS.items():
```

```
        text += f"✔ **{speciality}**\n  Години: {' '.join(times)}\n\n"
```

```
    await message.answer(text, parse_mode="Markdown")
```

```
# Перелік адміністративних послуг
```

```
@dp.message(F.text == "🏢 Послуги")
```

```
async def services_list(message: Message):
```

```
    text = "🏢 **Адміністративні послуги відділів:**\n\n"
```

```
    for speciality, service in AVAILABLE_SERVICES.items():
```

```
        text += f"✔ **{speciality}**\n  Послуги: {' '.join(service)}\n\n"
```

```
    await message.answer(text, parse_mode="Markdown")
```

```
# Мої записи
```

```
@dp.message(F.text == "📅 Мої записи")
```

```
async def my_appointments(message: Message):
```

```
    # Спочатку видаляємо старі записи
```

```
    delete_old_appointments()
```

```

rows = list(appointments_col.find({"user_id": message.from_user.id}).sort("date", 1))

if not rows:
    await message.answer("📅 У Вас поки немає записів.\n\nНатисніть кнопку  
нижче, щоб записатись.")
    return

text = "📅 **Ваші активні записи:**\n\n"
for r in rows:
    text += f"✓ {r['speciality']} | {r['date']} {r['time']}\n  Пацієнт: {r['user']}\n\n"
await message.answer(text, parse_mode="Markdown")

# Початок запису
@dp.message(F.text == "📅 Записатись до відділу")
async def start_booking(message: Message):
    user_booking_state[message.from_user.id] = {"step": "choose_doctor"}
    await message.answer(
        "📅 **Крок 1 з 4**",
        #\n\nОберіть відділ:",
        parse_mode="Markdown",
        reply_markup=get_departments_menu()
    )

# 2 з 4: Вибір відділу
@dp.message(F.text.in_(list(AVAILABLE_DEPARTMENTS.keys())))
async def choose_doctor(message: Message):
    user_id = message.from_user.id
    if user_id not in user_booking_state or user_booking_state[user_id].get("step") !=
"choose_department":
        return

    speciality = message.text

```

```

user_booking_state[user_id].update({"speciality": speciality, "step": "choose_date"})

await message.answer(
    f"✓ Обрано: **{speciality}**\n\n"
    f"□ Години: {'', ' '.join(AVAILABLE_DEPARTMENTS[speciality])}\n\n"
    f"📅 **Крок 2 з 4**\n\n"
    f"Введіть дату у форматі `YYYY-MM-DD`\n"
    f"Наприклад: `2025-09-25`",
    parse_mode="Markdown",
    reply_markup=ReplyKeyboardMarkup(keyboard=[[KeyboardButton(text="⬅️  
Назад")]], resize_keyboard=True)
)

```

Введення дати

```

@dp.message(lambda msg: msg.from_user.id in user_booking_state and
             user_booking_state[msg.from_user.id].get("step") == "choose_date")
async def choose_date(message: Message):
    user_id = message.from_user.id
    date_str = message.text.strip()

    if not validate_date(date_str):
        return await message.answer("✘ Неправильний формат  
дати.\n\nВикористайте: `YYYY-MM-DD`", parse_mode="Markdown")

    try:
        if datetime.strptime(date_str, "%Y-%m-%d").date() < datetime.now().date():
            return await message.answer("✘ Не можна записатись на минулу дату.")
    except:
        pass

    user_booking_state[user_id].update({"date": date_str, "step": "choose_time"})
    speciality = user_booking_state[user_id]["speciality"]
    times = AVAILABLE_DEPARTMENTS[speciality]

```

```

time_buttons = [[KeyboardButton(text=times[i]), KeyboardButton(text=times[i+1])] if
i+1 < len(times)
                else [KeyboardButton(text=times[i])] for i in range(0, len(times), 2)]
time_buttons.append([KeyboardButton(text="⬅️ Назад")])

await message.answer(
    f"✔️ Дата: **{date_str}**\n\n🕒 **Крок 3 з 4**\n\nОберіть час:",
    parse_mode="Markdown",
    reply_markup=ReplyKeyboardMarkup(keyboard=time_buttons,
resize_keyboard=True)
)

# Введення часу
@dp.message(lambda msg: msg.from_user.id in user_booking_state and
             user_booking_state[msg.from_user.id].get("step") == "choose_time")
async def choose_time(message: Message):
    user_id = message.from_user.id
    time_str = message.text.strip()

    if not validate_time(time_str):
        return await message.answer("❌ Неправильний формат часу.\n\nВикористайте:
`HH:MM`", parse_mode="Markdown")

    speciality = user_booking_state[user_id]["speciality"]

    if time_str not in AVAILABLE_DEPARTMENTS[speciality]:
        return await message.answer(
            f"❌ Час {time_str} недоступний.\n\nДоступні: {'
'.join(AVAILABLE_DEPARTMENTS[speciality])}",
            parse_mode="Markdown"
        )

```

```

date = user_booking_state[user_id]["date"]
if appointments_col.find_one({"speciality": speciality, "date": date, "time": time_str}):
    return await message.answer("✘ Цей час вже зайнятий.\n\nОберіть інший.")

user_booking_state[user_id].update({"time": time_str, "step": "enter_name"})

await message.answer(
    f"✔ Час: **{time_str}**\n\n👤 **Крок 4 з 4**\n\nВведіть ПІБ пацієнта:",
    parse_mode="Markdown",
    reply_markup=ReplyKeyboardMarkup(keyboard=[[KeyboardButton(text="⬅️ BACK  
Назад")]], resize_keyboard=True)
)

# Введення ПІБ
@dp.message(lambda msg: msg.from_user.id in user_booking_state and
             user_booking_state[msg.from_user.id].get("step") == "enter_name")
async def enter_name(message: Message):
    user_id = message.from_user.id
    user_name = message.text.strip()

    if len(user_name) < 3:
        return await message.answer("✘ Введіть повне ПІБ (мінімум 3 символи).")

    booking = user_booking_state[user_id]
    appointments_col.insert_one({
        "speciality": booking["speciality"],
        "date": booking["date"],
        "time": booking["time"],
        "user": user_name,
        "created_at": datetime.now(timezone.utc),
        "user_id": user_id
    })

del user_booking_state[user_id]

```

```

await message.answer(
    f"Запис створено!\n\n"
    f"🏥 {booking['speciality']}\n\n"
    f"📅 {booking['date']}"
    f"🕒 {booking['time']}\n\n"
    f"👤 {user_name}",
    parse_mode="Markdown",
    reply_markup=get_main_menu()
)

```

Кнопка «Повернення Назад»

```
@dp.message(F.text == "⬅️ Назад")
```

```
async def back_button(message: Message):
```

```
    user_id = message.from_user.id
```

```
    if user_id not in user_booking_state:
```

```
        return await message.answer("Оберіть дію:", reply_markup=get_main_menu())
```

```
    step = user_booking_state[user_id].get("step")
```

```
    if step == "choose_doctor":
```

```
        del user_booking_state[user_id]
```

```
        await message.answer("Оберіть дію:", reply_markup=get_main_menu())
```

```
    elif step == "choose_date":
```

```
        user_booking_state[user_id]["step"] = "choose_doctor"
```

```
        await message.answer("🏥 Крок 1 з 4\n\nОберіть відділ:",
```

```
        parse_mode="Markdown", reply_markup=get_departments_menu())
```

```
    elif step == "choose_time":
```

```
        user_booking_state[user_id]["step"] = "choose_date"
```

```
        await message.answer(
```

```
            f"📅 Крок 2 з 4\n\nВведіть дату: `YYYY-MM-DD`",
```

```
            parse_mode="Markdown",
```

```

        reply_markup=ReplyKeyboardMarkup(keyboard=[[KeyboardButton(text="⬅️
Назад")]], resize_keyboard=True)
    )
    elif step == "enter_name":
        user_booking_state[user_id]["step"] = "choose_time"
        speciality = user_booking_state[user_id]["speciality"]
        times = AVAILABLE_DEPARTMENTS[speciality]
        time_buttons = [[KeyboardButton(text=times[i]), KeyboardButton(text=times[i+1])] if
i+1 < len(times)
                        else [KeyboardButton(text=times[i])] for i in range(0, len(times), 2)]
        time_buttons.append([KeyboardButton(text="⬅️ Назад")])
        await message.answer("🕒 **Крок 3 з 4**\n\nОберіть час:",
parse_mode="Markdown",
                        reply_markup=ReplyKeyboardMarkup(keyboard=time_buttons,
resize_keyboard=True))

#
@dp.message(Command("admin_mode"))
async def admin_mode_cmd(message: Message, command: CommandObject):
    if not is_admin(message.from_user.id):
        return await message.answer("Немає доступу.")

# MAIN
async def main():
    print("Bot started.")

    # Перша очистка при запуску
    deleted = delete_old_appointments()
    if deleted > 0:
        print(f"При запуску видалено {deleted} старих записів")

    # Запускаємо фонову задачу очистки
    asyncio.create_task(cleanup_task())

    # Запускаємо бота

```

```

await bot.delete_webhook(drop_pending_updates=True)
await dp.start_polling(bot)

if __name__ == '__main__':
    asyncio.run(main())

# Інтелектуальний модуль
import asyncio
from aiogram import Bot, Dispatcher, types
from aiogram.filters import CommandStart
from openai import AsyncOpenAI

# Конфігурація
TG_TOKEN = "ТГ_ТОКЕН"
OPENAI_KEY = "OPENAI_KEY"

# Ініціалізація ШІ і бота
client = AsyncOpenAI(api_key=OPENAI_KEY)
bot = Bot(token=TG_TOKEN)
dp = Dispatcher()

# Обробка команди /start
@dp.message(CommandStart())
async def cmd_start(message: types.Message):
    await message.answer("Вітаю! Я віртуальний спеціаліст ЦНАП міста Дніпра.  
Чекаю Вашого питання!")

# Обробка текстових повідомлень
@dp.message()
async def gpt_answer(message: types.Message):
    # Відправляємо повідомлення, що бот друкує відповідь»
    await bot.send_chat_action(chat_id=message.chat.id, action="typing")

try:

```

```
# Запрос к API OpenAI (модель gpt-4o як найшвидшу і дешевшу в 2025 році)
response = await client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "system", "content": "Ти корисний помічник."},
        {"role": "user", "content": message.text}
    ]
)

# відправляємо відповідь користувачу
await message.reply(response.choices[0].message.content)

except Exception as e:
    await message.reply(f"помилка: {e}")

async def main():
    print("Бот почав працювати ...")
    await dp.start_polling(bot)

if __name__ == "__main__":
    asyncio.run(main())
```