

Міністерство освіти і науки України  
Харківський національний університет імені В. Н. Каразіна  
Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки

«Затверджую»  
Зав. кафедри теоретичної та  
прикладної системотехніки  
д.т.н., проф. С. І. Шматков  
«\_\_» \_\_\_\_\_ 2024 р.


## Пояснювальна записка

до кваліфікаційної роботи  
бакалавра


на тему: «**МОДЕЛЬ КОМП'ЮТЕРНОЇ СИСТЕМИ НАДАННЯ  
АДМІНІСТРАТИВНИХ ПОСЛУГ ІЗ ВИКОРИСТАННЯМ ЧАТ-БОТУ**»

Захищено на засіданні  
Атестаційної комісії № 42  
протокол № \_\_ від \_\_.06.2024 р.  
Оцінка \_\_\_\_ / \_\_\_\_  
Голова Атестаційної комісії  
\_\_\_\_\_ **СКОБ Ю. О.**

(підпис)

**Виконав:**  
студент 4 курсу, групи КІ – 41  
Галузь знань: 12 – Інформаційні  
технології  
Спеціальність: 123 – «Комп'ютерна  
інженерія»  
**НОВІКОВ Олексій Едуардович** 

**Керівник:**  
к.т.н., доцент ЗВО кафедри ТПС  
**СТРІЛЕЦЬ Вікторія Євгенівна** 

**Рецензент:**  
к.т.н., доцент ЗВО, в.о. завідувача  
кафедри теоретичної та прикладної  
інформатики  
**МЕНЯЙЛОВ Євген Сергійович** 

Харків – 2024

## АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і чотирьох додатків. Загальний обсяг роботи складає 87 сторінок, із яких 50 сторінок основної частини, 13 рисунків, 4 таблиці, 19 найменувань використаних джерел та п'ять додатків.

Метою кваліфікаційної роботи є спрощення доступу та вдосконалення якості отримання адміністративних послуг громадянам, через створення телеграм чат-боту для надання адміністративних послуг.

Об'єкт дослідження – процес надання адміністративних послуг громадянам.

Предмет дослідження – моделі і інформаційні системи надання адміністративних послуг.

Проблема яка вирішується у кваліфікаційній роботі полягає в тому щоб скориставшись існуючими програмними засобами створити телеграм чат-бот для надання адміністративних послуг, який дозволить спростити доступ та вдосконалити якість отримання адміністративних послуг. Телеграм чат-бот може стати ефективним рішенням для подолання цих проблем завдяки своїй доступності, зручності у використанні та можливостям для автоматизації процесів.

Область застосування – сфера надання адміністративних послуг населенню.

Ключові слова: телеграм бот, чат-бот, інформаційні системи, адміністративні послуги, бібліотека Telebot, Telegram Bot API.

## ABSTRACT

The explanatory note for the bachelor's qualification work consists of an introduction, three chapters, conclusions, a list of references, and four appendices. The total volume of the work is 87 pages, including 50 pages of the main part, 13 figures, 4 tables, 19 references, and four appendices.

The purpose of the qualification work is to simplify access and improve the quality of obtaining administrative services for citizens through the creation of a Telegram chatbot for providing administrative services.

The object of research is the process of providing administrative services to citizens.

The subject of research is the models and information systems for providing administrative services.

The problem addressed in the qualification work is to use existing software tools to create a Telegram chat bot for providing administrative services, which will simplify access and improve the quality of receiving administrative services. A Telegram chat bot can become an effective solution to these problems due to its accessibility, ease of use, and potential for process automation.

The scope is the sphere of providing administrative services to the population.

Keywords: Telegram bot, chatbot, information systems, administrative services, Telebot library, Telegram Bot API.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП .....	7
РОЗДІЛ 1. АНАЛІЗ ОСНОВ РОБОТИ ТА ФУНКЦІЙ ТЕЛЕГРАМ БОТІВ .....	9
1.1 Можливості телеграм ботів.....	9
1.2 Інструменти та технології для створення телеграм чат-ботів.....	15
1.3 Телеграм боти як ефективний інструмент у галузі надання адміністративних послуг .....	18
Висновки за розділом 1 .....	21
РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ КОМП'ЮТЕРНОЇ СИСТЕМИ ЦНАП.....	22
2.1 Аналіз системи роботи та функцій ЦНАП .....	22
2.2 Дослідження існуючих аналогів телеграм ботів .....	27
2.3 Опис реалізації та функцій телеграм чат-боту адміністративних послуг .	32
2.4 Модель комп'ютерної інформаційної системи надання адміністративних послуг .....	37
Висновки за розділом 2 .....	38
РОЗДІЛ 3. РОЗРОБКА ТЕЛЕГРАМ ЧАТ-БОТА АДМІНІСТРАТИВНИХ ПОСЛУГ.....	40
3.1 Інструменти використані для розробки телеграм бота.....	40
3.2 Аналіз та демонстрація роботи телеграм чат-боту адміністративних послуг .....	43
3.3 Загальна характеристика чат-боту .....	49
3.4 Актуальність застосування .....	50
Висновки за розділом 3 .....	51
ВИСНОВОК .....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	53
ДОДАТКИ .....	55
Додаток А .....	55
Додаток Б .....	57

Додаток В .....	60
Додаток Г .....	66
Додаток Д .....	70

## **ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ**

API – Application Programming Interface;

CRM – Customer Relationship Management;

CMS – Content management system;

FAQ – Frequently asked questions;

IDE – Integrated Development Environment;

JS – JavaScript;

SQL – Structured Query Language;

ЦНАП – Центр надання адміністративних послуг;

## ВСТУП

З впровадженням цифрових технологій у сучасну адміністративну практику постає необхідність пошуку нових ефективних інструментів для взаємодії між державними органами та громадянами. Одним із засобів полегшення цього процесу є застосування чат-ботів, які поєднують у собі швидкість, зручність та доступність для користувачів.

У цьому контексті центри надання адміністративних послуг виконують ключову роль у спрощенні взаємодії громадян з державними структурами. Кваліфікаційна робота спрямована на вивчення можливостей впровадження телеграм чат-ботів у роботу ЦНАП з метою оптимізації процесу надання адміністративних послуг та підвищення задоволеності громадян обслуговуванням.

Використання месенджера Telegram для реалізації чат-бота в контексті ЦНАП є обґрунтованим з точки зору його популярності серед користувачів та широкого функціоналу. Telegram забезпечує зручний інтерфейс для взаємодії з чат-ботом, а також підтримує різноманітні функції, такі як відправлення повідомлень, передача файлів, можливість інтеграції з іншими сервісами та багато іншого. Крім того, Telegram відомий своєю високою захищеністю даних, що є важливим фактором у сфері обробки особистої інформації та конфіденційності в адміністративних процедурах.

Об'єкт дослідження – процес надання адміністративних послуг громадянам.

Предмет дослідження – моделі і інформаційні системи надання адміністративних послуг.

**Метою** даної кваліфікаційної роботи є спрощення доступу та вдосконалення доступності отримання адміністративних послуг громадянам, через створення інформаційної системи для надання адміністративних послуг на основі телеграм чат-боту. Ця система спрямована на оптимізацію процесу

взаємодії між державними структурами та громадянами, забезпечуючи швидкий та зручний доступ до необхідної інформації та послуг через інтуїтивний інтерфейс чат-бота. Розробка такої системи передбачає вивчення потреб користувачів, аналіз їхніх переваг та проблем, а також створення ефективного механізму обробки запитів та надання відповідей з мінімальними затримками.

Для досягнення мети необхідно:

1. Аналіз існуючих моделей інформаційних систем із використанням чат-ботів, принципів їх роботи та структури.
2. Дослідження особливостей інформаційних систем надання адміністративних послуг.
3. Побудова моделі інформаційної системи надання адміністративних послуг із використанням чат-боту.
4. Програмна реалізація моделі інформаційної системи на основі чат-боту для надання адміністративних послуг та її тестування.

**Актуальність теми** "Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту" проявляється в нагальній потребі у зручному та ефективному отриманні державних послуг громадянами. Суспільство постійно розвивається та відчуває необхідність у зручному та оперативному доступі до державних послуг. Застосування чат-ботів у цій сфері є не лише сучасним трендом, але й необхідністю в умовах зростаючого обсягу адміністративних процедур та збільшення потоку користувачів.

Подібні системи мають потенціал значно полегшити взаємодію між державними органами та громадянами. Вони допомагають зменшити бюрократичні бар'єри, сприяють швидкому обміну інформацією та забезпечують постійний доступ до необхідних ресурсів. У контексті сучасних чинників, таких як пандемія та дистанційна робота, модель комп'ютерної системи з чат-ботом набуває ще більшої актуальності.

## РОЗДІЛ 1

### АНАЛІЗ ОСНОВ РОБОТИ ТА ФУНКЦІЙ ТЕЛЕГРАМ БОТІВ

#### 1.1 Можливості телеграм ботів

У світі, де технології швидко розвиваються, телеграм боти стали невід'ємною частиною цифрового життя. Телеграм боти – це автоматизовані програми, здатні взаємодіяти з користувачами через популярну месенджерську платформу Telegram. Їхня роль у сучасному світі надзвичайно важлива, а їх переваги і зручності у використанні стають очевидними з кожним днем.

У словах Віктора Гюго звучить: "Найсильніша з усіх сил – це ідея, час якої прийшов" [15]. Ця цитата ідеально підходить до визначення ролі телеграм ботів у сучасному житті. Вони створюють нові ідеї взаємодії з технологіями, що дають змогу користувачам отримувати доступ до різноманітних сервісів та інформації швидко та легко. Телеграм боти – це не просто програми, це ідея, яка втілюється в життя, змінюючи повсякденне існування людини.

Перш за все, телеграм боти забезпечують швидкий та зручний доступ до інформації та послуг. Вони можуть виконувати різноманітні функції, починаючи від надсилання новин і погоди до нагадування про події та розкладу. Користувачі можуть отримати необхідну інформацію або виконати певні завдання за кілька натискань безпосередньо у месенджері, що заощаджує їхній час і зусилля.

Крім того, телеграм боти можуть функціонувати у будь-який час доби, як вдень так і вночі, що робить їх дуже зручними для користувачів. Незалежно від того, коли людина потребує допомоги або інформації, вона може отримати відповіді від ботів в будь-який момент, не чекаючи на сервісну підтримку або робочий час консультанта.

Останнім часом, розвиток штучного інтелекту дозволяє телеграм ботам ставати все більш інтелектуальними та вдосконаленими. Вони можуть навчатися на основі даних, аналізувати взаємодію з користувачами та постійно покращуватися, щоб надавати кращий сервіс.

Можна сказати, що телеграм-боти стали невід'ємною частиною цифрового життя, допомагаючи людям отримувати інформацію, здійснювати покупки, отримувати рекомендації та вирішувати різні завдання за допомогою месенджера. Їхня зручність, швидкість та доступність роблять їх незамінними помічниками в сучасному світі.

### **1.1.1 Комунікація з користувачем**

Коли користувач відкриває чат з телеграм-ботом, першим кроком є запуск бота шляхом введення команди `"/start"` або натискання на кнопку "Старт". Після цього бот вітає користувача та представляється, розкриваючи свої можливості або запропонувавши набір дій, які можна здійснити. Користувач може відповісти на вітання, задати питання або виразити свої потреби.

Під час початкового етапу взаємодії бот може також пояснити свої можливості або навести приклади команд, які він розуміє. Він може звернути увагу користувача на функції, які можуть бути корисні для нього.

Початковий етап спілкування з телеграм-ботом може також включати в себе етап аутентифікації. Це особливо актуально для ботів, які надають доступ до конфіденційної інформації або функціоналу, що потребує ідентифікації користувача. У такому разі бот може запросити користувача увійти в свій обліковий запис або підтвердити свою особу, використовуючи різноманітні методи аутентифікації, такі як введення паролю, використання двофакторної аутентифікації або навіть відбитку пальця у випадку мобільних пристроїв з підтримкою цієї функції.

Подальше спілкування з ботом відбувається шляхом введення текстових повідомлень, які користувачі відправляють боту через чат. Цей простий, але ефективний спосіб комунікації дозволяє користувачам легко взаємодіяти з ботом, відправляючи команди, запити або звичайні повідомлення. Команди можуть бути у вигляді певних ключових слів або фраз, які бот може розпізнати і виконати відповідні дії.

У Telegram також існує можливість використання клавіатур для спрощення взаємодії з ботом. У боті можна створювати кнопки з певними діями або

варіантами відповідей, які користувач може вибрати за допомогою миші або торкання екрану, не потрібно вводити текст вручну.

Крім текстових повідомлень та кнопок, боти в Telegram можуть використовувати інші елементи інтерфейсу для взаємодії з користувачем, такі як зображення, відео або аудіофайли. Наприклад, бот може надіслати зображення з інформацією, схемами, графіками або ілюстраціями, щоб краще пояснити певні питання або концепції. Це дозволяє ботам надавати більш комплексні та зрозумілі відповіді, покращуючи якість комунікації з користувачем.

У табл. 1.1 наведено порівняння видів взаємодії користувачів і телеграм ботів, показані приклади запитів користувачів, а також сформульовані переваги кожного виду спілкування.

Таблиця 1.1

## Порівняння видів спілкування телеграм бота й користувача

Вид спілкування	Приклади запитів	Переваги
Текстове	Користувач задає боту питання "Яка погода на завтра?" Бот відповідає з інформацією про погоду.	Швидке та ефективне спілкування.
	Користувач відправляє замовлення в інтернет-магазині через бота. Бот підтверджує отримання та обробку замовлення.	Доступність 24/7 без необхідності очікування на оператора.
Мультимедійне	Користувач відправляє боту фотографію своєї проблеми. Бот надає інструкції щодо подальших дій або пересилає фото до фахівця для консультації.	Можливість передачі мультимедійного контенту для кращого розуміння проблеми.
	Бот надсилає користувачеві відеоінструкції з виконання певних завдань або користування сервісом.	Візуальне навчання та інструктаж.
Інтерактивне	Бот надає користувачеві можливість вибору опцій або дій в інтерактивному меню.	Простота та інтуїтивне користування.
	Бот виконує опитування користувачів, дозволяючи їм вибирати варіанти відповідей або вводити свої.	Залучення користувача до діалогу та збір відповідей для аналізу.

### 1.1.2 Сповіщення за допомогою телеграм ботів

Сповіщення через телеграм ботів є ключовим елементом сучасної комунікації, який надає низку переваг у порівнянні з іншими засобами сповіщення. Цей метод комунікації дозволяє надсилати користувачам інформацію в режимі реального часу, безпосередньо через месенджер Telegram. Використання ботів дозволяє автоматизувати процес надсилання повідомлень, що робить його більш ефективним та економічно вигідним для підприємств та користувачів.

Такі сповіщення можуть бути налаштовані для різних цілей, включаючи повідомлення про нові події, акції, оновлення програмного забезпечення, статус замовлення та багато іншого. Це дозволяє користувачам отримувати важливу інформацію безпосередньо на свої смартфони, що забезпечує миттєвий доступ до неї в будь-який час і в будь-якому місці.

Однією з ключових переваг використання телеграм ботів для сповіщень є їхня гнучкість і можливість персоналізації. Користувачі можуть налаштовувати отримання повідомлень згідно з власними уподобаннями і потребами, що робить процес комунікації більш ефективним та зручним.

Сповіщення в телеграм ботах можуть бути реалізовані за допомогою різних механізмів і технологій а саме:

- **Push-сповіщення.** Цей механізм дозволяє телеграм боту надсилати користувачам сповіщення навіть у випадках, коли вони не активно використовують чат з ботом. Це може бути використано для нагадувань про важливі події або повідомлення про акції.

- **Розсилка повідомлень.** Цей функціонал дозволяє боту надсилати однакові або різні повідомлення великій кількості користувачів або груп. Розсилка може бути спрямована на певні цільові аудиторії в залежності від їхніх інтересів або інших параметрів.

– **Заплановані повідомлення.** Ця функція дозволяє боту налаштувати час і дату надсилання сповіщень. Наприклад, це може бути використано для надсилання нагадувань про події або важливі терміни.

– **Інтерактивні сповіщення.** Цей тип сповіщень дозволяє боту надсилати користувачам повідомлення, з якими вони можуть взаємодіяти. Наприклад, це може бути кнопки або шаблони, які дозволяють користувачам вибирати варіанти відповідей або виконувати певні дії безпосередньо з повідомлення.

### **1.1.3 Інтеграція з іншими сервісами**

Інтеграція телеграм ботів з іншими сервісами є ключовим етапом в розвитку чат-ботів і створенні високоефективних рішень для користувачів. Цей процес дозволяє об'єднати функціональність телеграм ботів з іншими інтернет-сервісами або програмами, створюючи комплексні та гнучкі рішення для різноманітних сфер діяльності. Такими прикладами є інтеграції з соціальними медіа, інтеграції з платіжними системами, інтеграції CRM-систем або API сторонніх сервісів.

Одним із основних переваг інтеграцій є покращення користувацького досвіду. Користувачі можуть здійснювати широкий спектр дій безпосередньо з месенджера, не виходячи на зовнішні платформи або веб-сайти. Наприклад, користувач може отримати нагадування про свій розклад або замовити продукти безпосередньо через телеграм бота, взаємодіючи з іншими сервісами, що відповідають за ці функції.

Крім того, інтеграція телеграм ботів з іншими сервісами сприяє оптимізації бізнес-процесів. Наприклад, автоматизація взаємодії з CRM-системами дозволяє підвищити ефективність роботи з клієнтами та обробку замовлень. Також це дозволяє організаціям здійснювати обмін даними між різними системами у реальному часі, що сприяє оперативному та точному прийняттю управлінських рішень.

Основні аспекти інтеграції телеграм ботів з іншими сервісами:

– **Веб-сайти і системи керування контентом.** Інтеграція телеграм-ботів з веб-сайтами дозволяє користувачам спілкуватися з ботом безпосередньо на веб-сайті. Це може бути реалізовано через вбудовані кнопки чат-бота або віджети. Щодо систем керування контентом, таких як CMS (системи управління контентом), інтеграція може забезпечити автоматичне оновлення контенту в боті на основі змін на веб-сайті.

– **Платіжні системи.** Інтеграція з платіжними системами дозволяє користувачам здійснювати оплату через телеграм-бота. Це може бути корисно для магазинів, сервісів доставки їжі, онлайн-послуг та інших бізнесів, які пропонують товари або послуги через цю платформу.

– **Інші соціальні мережі.** Інтеграція з соціальними мережами дозволяє користувачам використовувати функціонал телеграм-бота через інші месенджери або платформи, такі як Facebook Messenger або WhatsApp. Це може полегшити комунікацію з користувачами, які використовують різні месенджери.

– **Бази даних та CRM.** Інтеграція з базами даних дозволяє зберігати та обробляти дані, що надходять через телеграм-бота, в існуючих базах даних або хмарних сервісах. Це може бути корисно для зберігання історії чатів, персоналізації взаємодії з користувачами, аналізу даних тощо.

У табл. 1.2 наведені приклади інтеграції телеграм ботів з різними сервісами.

Таблиця 1.2

#### Приклади інтеграції

Тип інтеграції	Приклад
API інтеграція	Використання API для зв'язку бота з зовнішніми сервісами, наприклад, Google Calendar API.
Веб-сайти	Додавання чат-бота на веб-сайт для прямого спілкування з відвідувачами.
Платіжні системи	Можливість оплати товарів або послуг безпосередньо через чат за допомогою Stripe або PayPal.

Продовження табл. 1.2

Тип інтеграції	Приклад
Зовнішні сервіси	Використання Google Maps API для надання користувачам інформації про місцезнаходження.
Соціальні мережі	Можливість спілкування з ботом через Facebook Messenger або Viber.
Бази даних та CRM	Збереження та оновлення даних про користувачів у CRM-системі через API.

#### 1.1.4 Автоматизація процесів

У контексті автоматизації процесів телеграм боти виявляються потужним інструментом для забезпечення ефективної роботи з користувачами. Основна мета використання телеграм ботів у контексті автоматизації полягає в тому, щоб спростити та прискорити виконання рутинних операцій і завдань, а також знизити витрати ресурсів та часу.

Телеграм боти дозволяють автоматизувати широкий спектр процесів, від обробки замовлень та надання інформації користувачам до моніторингу подій і нагадувань. Наприклад, вони можуть автоматично відповідати на запити користувачів, обробляти замовлення, надсилати сповіщення про статуси доставки товарів або про події, які цікавлять користувача.

Однією з важливих переваг телеграм ботів є їхня здатність взаємодіяти з користувачами у реальному часі через текстові повідомлення, що робить взаємодію з ними інтуїтивно зрозумілою та зручною. Крім того, боти можуть бути налаштовані для виконання конкретних завдань відповідно до потреб користувачів або бізнесу.

#### 1.2 Інструменти та технології для створення телеграм чат-ботів

Створення чат-ботів – це складний та багатоаспектний процес, який вимагає використання різноманітних інструментів та технологій. Розробники повинні ретельно планувати та вибирати інструменти для створення, навчання та впровадження чат-ботів з огляду на специфіку проекту та потреби аудиторії. Процес розробки може включати в себе роботу над алгоритмами обробки

природної мови, створення візуального інтерфейсу для чат-бота, інтеграцію зі сторонніми сервісами та системами.

Одним із ключових аспектів при розробці чат-ботів є вибір мови програмування. Вибір мови визначає не лише технічні можливості, а й швидкість розробки, ефективність та простоту супроводження проекту в майбутньому. Відповідно до поставлених завдань та потреб проекту, розробники обирають мову, яка найкраще підходить для реалізації поставлених завдань, забезпечуючи оптимальне поєднання продуктивності та якості коду.

Важливу роль у розробці чат-ботів відіграє управління даними. Чат-боти зазвичай використовують бази даних для зберігання та організації інформації, необхідної для їхньої роботи. Ці дані можуть включати профілі користувачів, історію повідомлень, налаштування та іншу контекстну інформацію. Важливо забезпечити надійність, ефективність та безпеку управління даними, оскільки це впливає на якість роботи чат-бота та задоволення користувачів.

Крім цього невід'ємними компонентами у розробці чат-ботів є фреймворки та бібліотеки, оскільки вони надають розробникам готові компоненти та функціонал, що спрощує процес створення та полегшує роботу з різноманітними аспектами розробки. Вони дозволяють швидше та ефективніше створювати та розгортати чат-ботів, забезпечуючи гнучкість та розширюваність функціоналу. Розробники можуть використовувати готові модулі та компоненти, які пропонуються фреймворками та бібліотеками, для реалізації різноманітних функціональних можливостей чат-бота без необхідності писати код з нуля.

### **1.2.1 Мови програмування, їх бібліотеки та фреймворки**

При розробці телеграм чат-ботів можна використовувати різні мови програмування. Кожен обирає собі мову до вподоби, але найпоширенішими та популярними є:

1. **Python** є дуже популярною мовою програмування у сфері розробки чат-ботів, включаючи телеграм. Вона відома своєю простотою вивчення, ефективним синтаксисом та багатими бібліотеками, що роблять розробку швидкою та зручною. Для роботи з телеграм API доступні різноманітні бібліотеки, такі як

Telebot, python-telegram-bot та aiogram, які надають зручний інтерфейс для взаємодії з ботом. Python також широко використовується у сферах штучного інтелекту та машинного навчання, що може бути корисним для створення інтелектуальних чат-ботів [2, 6, 7].

2. **Node.js (JavaScript)** є однією з найпопулярніших мов програмування для серверної розробки, а його можливості дозволяють розробникам легко створювати чат-ботів для телеграм. Завдяки бібліотеці Telegraf, розробники можуть швидко реалізувати різноманітні функції чат-бота, включаючи обробку повідомлень, реакції на події та інші аспекти. JavaScript також є дуже популярним у веб-розробці, що дає змогу використовувати знання та інструменти для обох напрямків розробки.

3. **Java** залишається однією з найбільш розповсюджених мов програмування у світі, і багато розробників використовують її для створення телеграм чат-ботів. Фреймворк TelegramBots API для Java забезпечує простий інтерфейс для роботи з телеграм API, дозволяючи швидко розробляти функціональні боти. Java також відома своєю надійністю та масштабованістю, що робить її привабливим вибором для великих проєктів.

4. **Ruby**, спільно з фреймворком Telegram Bot для Ruby, надає можливості для розробки чат-ботів у зручному середовищі. Ruby відома своєю простотою та елегантністю, що робить її привабливим вибором для багатьох розробників. Фреймворк Telegram Bot для Ruby дозволяє швидко створювати ботів з різноманітними функціями та можливостями.

У таблиці Г.1 сформульовані основні переваги і недоліки наведених мов програмування з точки зору їх застосування для створення чат-ботів.

### 1.2.2 Мови баз даних

**SQLite** – це легка, вбудована, база даних SQL, яка ідеально підходить для малих до середніх проєктів, включаючи телеграм чат-боти. Однією з його головних переваг є те, що для роботи з ним не потрібно окремого сервера баз даних, оскільки весь функціонал реалізований вбудовано в додаток. Це дозволяє

легко і швидко розгорнути інсталяції SQLite там, де потрібна локальна база даних [9].

**MySQL** – одна з найпоширеніших реляційних баз даних у світі, вона використовується для розробки багатьох веб-проектів, включаючи телеграм чат-боти. MySQL має широкий набір функцій, високу швидкодію та досить простий у використанні. Його можна масштабувати для обробки великих обсягів даних і володіє великою спільнотою користувачів, що забезпечує підтримку та розвиток [4, 8].

**PostgreSQL** – це потужна реляційна база даних з великим набором функцій та підтримкою розширень. Вона забезпечує високу надійність, масштабованість та можливості, які можуть бути корисні для складних телеграм чат-ботів. PostgreSQL підтримує різноманітні типи даних, операції індексування та інші функції, які полегшують розробку та оптимізацію ботів.

**MongoDB** – це документо-орієнтована NoSQL база даних, яка використовує схему даних JSON-подібного формату. Вона дозволяє зберігати структуровані дані у вигляді документів, що робить її ідеальним вибором для телеграм чат-ботів, де потрібна гнучкість схеми даних та швидкий доступ до них. MongoDB також підтримує горизонтальне масштабування, що дозволяє розподілити базу даних на кілька серверів для підтримки великих обсягів даних та високої навантаженості.

У таблиці Г.2 перелічені основні переваги і недоліки наведених мов баз даних для створення чат-ботів.

### **1.3 Телеграм боти як ефективний інструмент у галузі надання адміністративних послуг**

Телеграм боти в сфері надання адміністративних послуг неабияк вразили своєю ефективністю, розширюючи можливості для користувачів та державних установ. Вони не лише прискорюють та оптимізують процеси, а й надають додатковий рівень зручності та доступності.

Звичайно, першим та очевидним аспектом їх ефективності є швидкий та легкий доступ до необхідної інформації чи послуг просто через месенджер

Telegram. Це робить процес отримання допомоги або інформації максимально зручним для кожного користувача. Замість того, щоб шукати веб-сайти чи дзвонити, можна просто написати боту та отримати потрібну інформацію відразу.

Також важливу роль відіграє автоматизація багатьох процесів, що робить використання телеграм ботів ще більш ефективним. Наприклад, заповнення форм або реєстрація на прийом до спеціаліста тепер можуть здійснюватися миттєво, без необхідності заповнювати складні паперові документи чи чекати в чергах. Це звільняє користувачів від зайвих турбот та робить взаємодію з державними установами більш приємною та ефективною.

Далі наведені деякі *функції центру надання адміністративних послуг*, які можна реалізувати за допомогою телеграм чат-боту, й коротко описано, яким чином це буде працювати.

*Функція «Запис на прийом»:*

- Користувач може звернутися до телеграм чат-бота, використовуючи команду або кнопку "Записатися на прийом".

- Чат-бот може запросити від користувача додаткову інформацію, таку як тип послуги, яка потрібна, або причину візиту.

- Після цього чат-бот може показати доступні дати та часи для прийому, або надати користувачеві можливість вибрати з розкладу роботи співробітників ЦНАП.

- Користувач може обрати зручну дату та час, і після підтвердження запису, отримає підтвердження про прийом через телеграм.

*Функція «Інформаційне обслуговування»:*

- Чат-бот може мати базу знань або доступ до системи, де зберігається інформація про всі послуги, які надаються ЦНАП.

- Користувач може поставити запитання про будь-яку послугу, і чат-бот зможе надати детальну інформацію про неї, включаючи вимоги до документів, терміни отримання та інше.

Функція «Подача документів»:

- Чат-бот може запропонувати користувачу відправити необхідні документи безпосередньо через чат.
- Користувач може завантажити скан або фотографію необхідних документів, або заповнити форму онлайн, відповідаючи на питання чат-бота.
- Після завершення процесу подачі документів, чат-бот може надіслати підтвердження про їхню отримання та інформацію про подальші кроки.

Функція «Статус заявки»:

- Користувач може запитати про статус своєї заявки або документів через телеграм чат-бот, надавши необхідну інформацію для ідентифікації.
- Чат-бот може автоматично сповіщати користувача про зміни статусу заявки або процесу розгляду, щоб забезпечити актуальну інформацію.

Функція «Оплата послуг»:

- Чат-бот може надати користувачу список доступних методів оплати за послуги.
- Після вибору методу оплати, чат-бот може надати інструкції щодо подальших кроків для завершення оплати.
- Користувач може здійснити оплату безпосередньо через чат-бота, використовуючи зручний для нього метод. Після успішної оплати, чат-бот може надіслати підтвердження оплати та інформацію про отримання послуги.

Використання телеграм ботів як засобу автоматизації надання адміністративних послуг має свої переваги (табл. 1.3).

Таблиця 1.3

Переваги ЦНАП телеграм ботів

Критерій	Телеграм чат-бот	Відділення ЦНАП
Доступність	Доступний цілодобово та без вихідних через телеграм месенджер	Обмежений робочий час, зазвичай з 9:00 до 18:00 в будні, з вихідними

Продовження табл. 1.3

Критерій	Телеграм чат-бот	Відділення ЦНАП
Місцезнаходження	Відсутність фізичної локації, доступний з будь-якого місця	Фізичне відділення, потрібно відвідувати особисто
Зручність	Зручний спосіб здійснення запису на прийом та отримання інформації, можливість взаємодії у будь-який час	Потрібно витратити час на поїздки до відділення та очікування у черзі
Інформаційна підтримка	Надає детальну інформацію про послуги та статуси заявок у реальному часі	Інформаційна підтримка обмежена, зазвичай доступна лише у робочий час
Подача документів	Можливість зручного завантаження документів безпосередньо через чат-бот	Необхідність особисто принести документи до відділення
Оплата послуг	Здійснення оплати за послуги безпосередньо через телеграм чат-бот	Оплата послуг можлива лише у відділенні або через банк
Взаємодія з користувачем	Автоматизований процес взаємодії з користувачем через бота	Взаємодія особисто від співробітників ЦНАП

## Висновки за розділом 1

У розділі 1 детально проаналізовано потенціал та переваги використання телеграм ботів. Зокрема, було висвітлено їхню роль у взаємодії з користувачем, забезпечення сповіщень та інтеграцію з різними сервісами. Крім того, було розглянуто широкий спектр інструментів та технологій, які можна використовувати для створення та підтримки телеграм ботів, у тому числі мови програмування та бази даних.

Не менш важливою є роль телеграм ботів у сфері адміністративних послуг, де вони можуть значно спростити процедури та покращити обслуговування користувачів, забезпечуючи зручний та ефективний інтерфейс для взаємодії з громадянами. Був проведений аналіз який дозволяє краще зрозуміти потенціал та можливості використання телеграм ботів у різних сферах діяльності та їхню значущість для сучасного суспільства.

## РОЗДІЛ 2.

### РОЗРОБКА МОДЕЛІ КОМП'ЮТЕРНОЇ СИСТЕМИ ЦНАП

#### 2.1 Аналіз системи роботи та функцій ЦНАП

Адміністративні послуги – це різноманітні послуги, що надаються державними органами або муніципальними установами громадянам, бізнесу та іншим суб'єктам з метою забезпечення їхніх прав, виконання законних обов'язків або здійснення певних дій чи операцій. Ці послуги спрямовані на забезпечення ефективного функціонування суспільства, дотримання законів та регуляцій, а також на задоволення потреб громадян та бізнесу.

Адміністративні послуги можуть бути різного характеру і охоплювати широкий спектр сфер діяльності.

##### 2.1.1 Реєстраційні послуги

Ці послуги спрямовані на реєстрацію різних об'єктів, подій або фактів, що відображаються в офіційних документах та реєстраційних системах. Реєстраційні послуги можуть мати різноманітний характер і охоплювати різні сфери життя суспільства.

Основними видами реєстраційних послуг є:

1. *Реєстрація осіб.* Ця функція може включати реєстрацію громадян, бізнес-суб'єктів, неприбуткових організацій, політичних партій, соціальних об'єднань та інших юридичних чи фізичних осіб.
2. *Реєстрація власності.* Включає реєстрацію прав власності на рухоме та нерухоме майно, земельні ділянки та інші об'єкти.
3. *Реєстрація бізнесу та юридичних осіб.* Включає реєстрацію підприємств, компаній, корпорацій, та інших комерційних та некомерційних організацій.
4. *Реєстрація інтелектуальної власності.* Включає реєстрацію авторських прав, патентів, торгових марок, дизайнів та інших об'єктів інтелектуальної власності.

5. *Реєстрація шлюбу, розлучення, народження та смерті.* ЦНАП може обробляти документи, пов'язані зі шлюбом, розлученням, народженням та смертю, та видачу відповідних свідоцтв.

### **2.1.2 Ліцензійні та дозвільні послуги**

Ці послуги регулюються законодавством та спрямовані на регулювання та контроль за певними видами діяльності або використанням ресурсів з метою забезпечення безпеки, здоров'я, порядку, екології та інших аспектів суспільного життя.

Основні характеристики ліцензійних та дозвільних послуг:

1. *Видача ліцензій на певні види діяльності.* Надання послуг по видачі ліцензій на медичну практику, фармацевтичну діяльність, освітню діяльність, будівництво та інші види професійної або комерційної діяльності.

2. *Видача дозволів на проведення різних заходів.* Оформлення й видача дозволів на проведення публічних заходів, виставок, концертів, спортивних змагань, торгівельних ярмарків тощо.

3. *Дозвільні послуги в галузі будівництва.* Включає видачу дозволів на будівництво, реконструкцію, демонтаж будівель та споруд, а також на видалення дерев або встановлення рекламних конструкцій.

4. *Ліцензування та дозволи на транспортні засоби.* Отримання дозволів на водіння автомобілів, мотоциклів, вантажних автівок тощо, а також ліцензій на пасажирський та вантажний транспорт.

5. *Ліцензії на використання природних ресурсів.* Включає отримання ліцензій на видобуток та використання природних ресурсів, таких як вода, ліси, мінерали тощо.

### **2.1.3 Соціальні послуги**

Соціальні послуги – це різноманітні види допомоги та підтримки, які надаються державними органами, благодійними фондами та іншими установами з метою забезпечення добробуту, безпеки, самостійності та якості життя різних категорій населення.

Основні призначення соціальних послуг:

1. *Консультації з соціальних питань.* Можуть включати надання інформації про соціальні програми, права та пільги громадян, поради з питань зайнятості, житлової сфери, медичного обслуговування та інше.
2. *Реєстрація на отримання соціальних послуг.* Допомога громадянам у процесі реєстрації на отримання соціальних пільг, допомога з оплати комунальних послуг, медичного обслуговування та інших програм.
3. *Допомога у здійсненні пенсійних та соціальних виплат.* Надання послуг з отримання пенсій, соціальних виплат, матеріальної допомоги та інших видів соціальної підтримки, а також консультації з питань отримання цих виплат.

#### **2.1.4 Фінансові послуги**

Фінансові послуги – це різноманітні продукти та послуги, що надаються фінансовими установами, такими як банки, страхові компанії, інвестиційні фонди, пенсійні фонди та інші фінансові інститути. Вони мають на меті забезпечення фінансової безпеки, зручності та ефективності для клієнтів у їхній фінансовій діяльності та управлінні грошовими потоками.

Основні види фінансових послуг включають:

1. *Прийом платежів.* Прийняття платежів за різними видами послуг, такими як комунальні послуги (електроенергія, газ, вода), податки, штрафи, адміністративні та інші платежі.
2. *Видача довідок та рахунків.* Надання громадянам довідки про внесення платежів, рахунки на оплату послуг, а також інші фінансові документи, необхідні для подання звітності або здійснення операцій.
3. *Допомога у вирішенні фінансових питань.* Фахівці ЦНАП можуть надавати консультації з питань сплати податків, правильного оформлення платежів, а також інші фінансові поради та допомогу.
4. *Реєстрація та видача фінансових документів.* Здійснення реєстрації та видача різних фінансових документів, таких як податкові відомості, довідки про сплату податків, рахунки на оплату, довідки про нарахування штрафів та інші.

### **2.1.5 Юридичні послуги**

Юридичні послуги – це різноманітні види допомоги та консультацій, що надаються юристами, адвокатами та іншими юридичними професіоналами з метою захисту прав та інтересів клієнтів, вирішення юридичних питань та ведення справ у судах або перед іншими правоохоронними органами.

Основними видами юридичних послуг є:

1. *Консультації з правових питань.* Юридичні консультації можуть стосуватися різних аспектів права, таких як сімейне право, спадщина, нерухомість, трудові відносини, бізнес та інші. Фахівці можуть надати клієнтам інформацію про їхні права та обов'язки, а також поради щодо подальших дій у вирішенні юридичних питань.

2. *Нотаріальні послуги.* Забезпечення нотаріального посвідчення документів, таких як договори, довіреності, спадкові акти та інші юридичні документи. Нотаріуси виконують функцію свідків укладення договорів та забезпечують їх легальність та вірогідність.

3. *Реєстраційні послуги.* Надання послуг з реєстрації різних подій та фактів, таких як шлюб, розлучення, народження, смерть, а також реєстрація прав на нерухомість, бізнес та інші види реєстрації.

### **2.1.6 Транспортні послуги**

Транспортні послуги – це різноманітні види послуг, пов'язаних з переміщенням товарів, пасажирів та інших вантажів з одного місця до іншого. Ці послуги забезпечуються різними видами транспорту, такими як автомобільний, залізничний, повітряний, водний та міський громадський транспорт.

Основні види транспортних послуг включають:

1. *Реєстрація транспортних засобів.* Це включає реєстрацію автомобілів, мотоциклів, скутерів, велосипедів тощо. Також видача документів такі як технічний паспорт, страховий поліс, реєстраційний номер, які необхідні для користування транспортними засобами на дорозі.

2. *Оформлення водійських посвідчень.* Надання послуг з отримання водійських посвідчень, в тому числі інформування про процедуру отримання водійського посвідчення.

3. *Видача технічних паспортів транспортних засобів.* Здійснення процедури видачі технічних паспортів на транспортні засоби, які необхідні для реєстрації та експлуатації транспортних засобів.

4. *Надання інформаційно-консультаційної підтримки.* Надання консультацій та інформаційної підтримки громадянам щодо правил дорожнього руху, проведення процедур реєстрації транспортних засобів, видачі водійських посвідчень та інших питань, пов'язаних з транспортом.

### **2.1.7 Житлові та комунальні послуги**

Житлові та комунальні послуги включають в себе широкий спектр сервісів, пов'язаних з житловими умовами та комунальними потребами населення. Центри надання адміністративних послуг можуть виконувати ряд функцій у сфері надання житлових та комунальних послуг:

1. *Реєстрація мешканців та адреси проживання.* Надання послуг з реєстрації мешканців, а також зміни адреси проживання. Це важлива послуга, яка стосується багатьох аспектів життя громадян, включаючи отримання документів, користування комунальними послугами та отримання соціальної допомоги.

2. *Надання інформації про житлові умови.* Працівники ЦНАП можуть надавати консультації та інформацію про різні аспекти житлових умов, включаючи права та обов'язки орендарів та власників житла, порядок користування комунальними послугами, питання енергоефективності тощо.

3. *Підключення до комунальних мереж.* Надання допомоги громадянам та підприємствам у підключенні до комунальних мереж, таких як водопостачання, газопостачання, електропостачання тощо. Це включає в себе подання заяв, оформлення договорів та інші адміністративні процедури.

### **2.2 Дослідження існуючих аналогів телеграм ботів**

Цей розділ роботи присвячений детальному огляду різних аналогів ЦНАП телеграм-ботів, а також їхнього порівняння. Дослідимо широкий спектр

можливостей, які надають ці боти, починаючи від обробки запитів користувачів до надання конкретних послуг та інформаційної підтримки. Аналізуючи ці аспекти, варто зосередитись на виявленні переваг та недоліків кожного з розглянутих аналогів [1].

Ретельне дослідження даного питання дозволить знайти ті функції, які допоможуть забезпечити користувачів найвищою якістю обслуговування та оптимізувати процеси надання адміністративних послуг через телеграм-ботів.

У рамках дослідження зосередимось на розгляді чотирьох аналогів телеграм-ботів центрів надання адміністративних послуг у різних містах України. Докладно проаналізуємо та порівняємо функціональні можливості ботів у містах Чернігів, Черкаси, Харків та Київ. Розглянемо їхні переваги та недоліки, оцінимо рівень зручності та ефективності взаємодії з користувачами, а також виявимо ті аспекти, які можуть бути ключовими для забезпечення найвищої якості обслуговування громадян та вдосконалення процесів надання адміністративних послуг через телеграм-ботів у відповідних містах.

### **2.2.1 Телеграм чат-бот міста Чернігів**

Телеграм бот ЦНАП міста Чернігів пропонує користувачам широкий спектр послуг та інформації, але дизайн його меню може бути покращений для забезпечення кращої зручності та читабельності [16]. Головне меню бота відзначається непоганим дизайном, проте є деякі аспекти, які потребують уваги. Наприклад, хоча воно має привабливий зовнішній вигляд, відсутність розподілу та підписів може ускладнювати навігацію для користувачів. Важливо враховувати, що зручний та легко читабельний інтерфейс є ключовим елементом для успішної взаємодії з користувачами.

Щодо функцій, бот пропонує різноманітні можливості, такі як доступ до послуг, контактної інформації, питань та відповідей, графіку прийому громадян. Хоча функціонал є широким, деякі аспекти можуть вимагати поліпшення, наприклад, більше вичерпних відповідей у розділі питань та відповідей, а також більш структурований та зрозумілий інтерфейс для графіку прийому громадян.

На рис. 2.1 продемонстровано інтерфейс головного меню телеграм боту міста Чернігова.

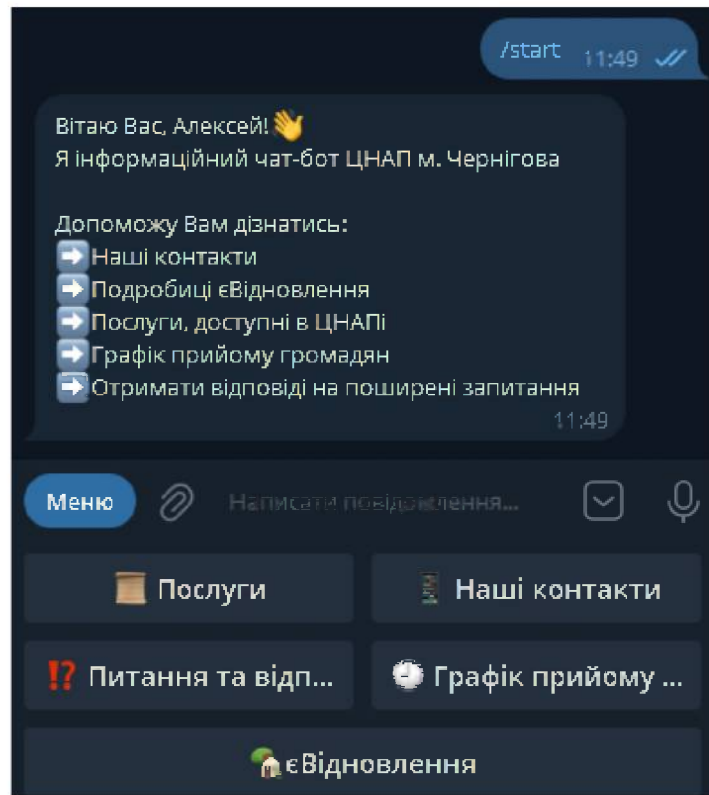


Рисунок 2.1 – Головне меню чат-боту міста Чернігів

Чат-бот Чернігова представляє собою суто інформаційний чат-бот і виявляється досить поверхневим у своєму функціоналі. Незважаючи на наявність деяких базових функцій, загалом можна відзначити відсутність широкого спектру сервісів та інтерактивних можливостей для користувачів. Крім того, місцями поганий дизайн та ієрархія меню ускладнюють взаємодію з ботом та можуть призводити до заплутаності для користувачів. Узагальнення переваг та недоліків чат-боту наведено в таблиці Г.3.

### 2.2.2 Телеграм чат-бот міста Черкаси

Телеграм бот центру надання адміністративних послуг у місті Черкаси призначений для забезпечення громадян зручної взаємодії з міськими службами та послугами [19]. Хоча його головне меню може здатися трохи простим і дизайн

не дуже привабливим, бот пропонує деякі корисні функції, які допомагають мешканцям міста отримувати необхідну інформацію та послуги.

Одна з основних функцій бота – це реєстрація в черзі на обслуговування. Хоча ця функція може викликати деякі незручності через відсутність inline клавіатури, яка дозволяє швидко повертатися до головного меню, все ж таки ця функція надає можливість користувачам уникнути довгих черг та зайвого очікування.

Крім того, бот також пропонує можливість відстежувати етапи виконання певних послуг та оцінювати якість обслуговування. Функція відстеження дозволяє користувачам швидко отримувати інформацію про замовлені ними послуги в минулому. Оцінювання якості обслуговування надає змогу отримувати зворотній зв'язок від користувачів, що сприяє постійному вдосконаленню чат-бота.

На рисунку 2.2 продемонстровано інтерфейс головного меню телеграм боту міста Черкаси.

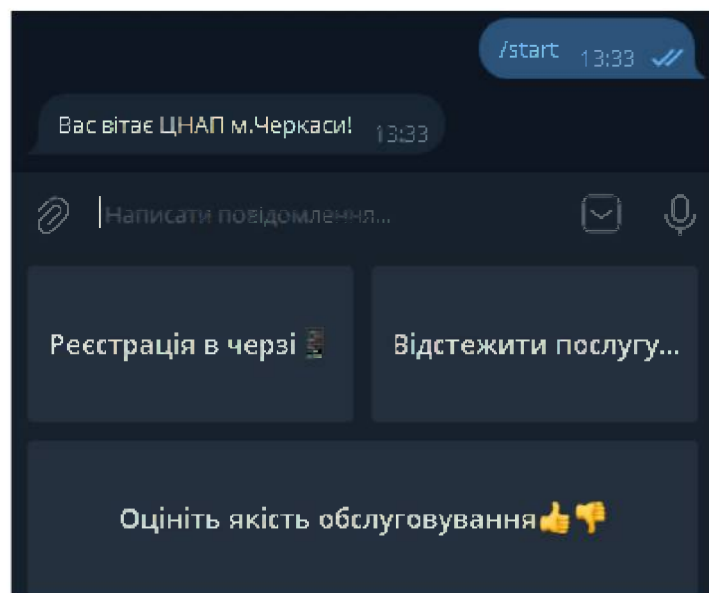


Рисунок 2.2 – Головне меню чат-боту міста Черкаси

Чат-бот Чернігова має деякі переваги, такі як наявність базових функцій і здатність надавати користувачам певну інформацію. Однак, він може виглядати

дещо обмеженим через малу кількість реалізованих у ньому функцій. Зокрема, відсутність широкого спектру сервісів та інтерактивних можливостей може зменшувати його привабливість для користувачів. Треба відзначити, що деякі аспекти дизайну та ієрархії меню можуть бути поліпшені, щоб зробити взаємодію з чат-ботом більш зручною. Узагальнення переваг та недоліків чат-боту наведено в таблиці Г.4.

### 2.2.3 Телеграм чат-бот міста Харків

Телеграм чат-бот міста Харків є платформою, яка надає користувачам можливість отримати широкий спектр адміністративних послуг у зручному форматі [18]. Він, на жаль, відзначається обмеженим функціоналом та відсутністю клавіатури, що може ускладнити навігацію для користувачів. Проте, серед доступних функцій варто відзначити можливість отримати відповіді на питання, що є досить зручною та ефективною функцією, яка виглядає досить гарно оформленою. Крім того, користувачі можуть скористатися можливістю зареєструватися в черзі, що також працює надійно.

Загалом, хоча всі реалізовані функції працюють належним чином і мають привабливий дизайн, але натомість має дуже мало функціоналу, що є певним недоліком цього чат-бота.

На рисунку 2.3 продемонстровано інтерфейс головного меню телеграм боту міста Харків.

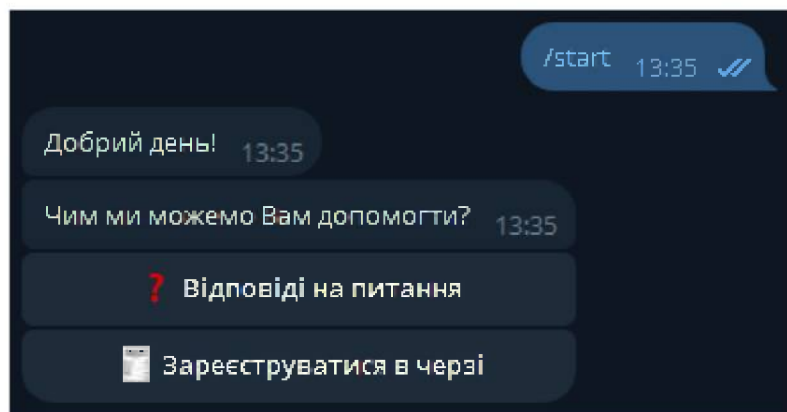


Рисунок 2.3 – Головне меню чат-боту міста Харків

Загальні враження від чат-боту міста Харків досить позитивні. Реалізовані функції працюють ефективно і без нарікань, що є важливим критерієм для користувачів. Однак, головним недоліком є мала кількість функцій, що обмежує можливості користувачів, що впливає на загальну користь цього застосунку. Узагальнення переваг та недоліків чат-боту наведено в таблиці Г.5.

#### **2.2.4 Телеграм чат-бот міста Київ**

Телеграм чат-бот міста Київ створений з метою спростити процес отримання адміністративних послуг для мешканців. Попри його потенціал, він має деякі недоліки, які варто врахувати. Головне меню, хоч і виглядає привабливо, проте, виявляється не досконалим через відсутність inline клавіатури. Це може призвести до необхідності постійного прокручування вгору для доступу до головного меню або повторного введення команди "start".

Наслідком цього є ускладнення процесу взаємодії з ботом та можливе зниження задоволення користувачів від його використання. Тим не менш, серед доступних функцій варто відзначити можливість перегляду адрес, що становить велику користь для мешканців при пошуку різних міських установ та організацій.

Деякі функції, такі як реєстрація в електронній черзі та інформація про стан розгляду справи, доступні лише через посилання на веб-сайт, що може викликати незручності у деяких користувачів через необхідність переходу між різними платформами.

Бот також надає можливість скористатися переліком послуг, доступних в особистому кабінеті, а також залишити відгук про надані послуги. Деякі інші функції, такі як Call центр ЦНАП та графік роботи, представлені досить маломісткими способами, що може ускладнити взаємодію користувачів з ботом.

Загалом, бот міста Київ має свої переваги та недоліки. Хоча він є корисним інструментом для отримання адміністративних послуг, він може бути покращений для забезпечення більшої зручності та доступності користувачам.

На рис. 2.3 продемонстровано інтерфейс головного меню телеграм боту міста Київ.

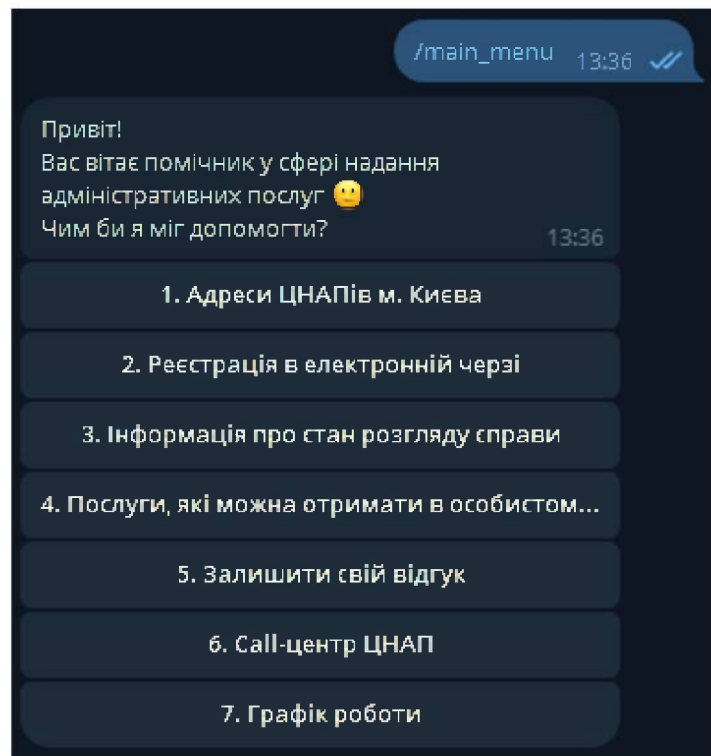


Рисунок 2.4 – Головне меню чат-боту міста Київ

Хоча бот пропонує широкий спектр корисних функцій, деякі з них можуть бути неідеально реалізовані або відсутні у самому боті (табл. 2.4). Особливо це стосується обмеженої зручності головного меню та відсутності дуже важливих функцій у самому боті, що може ускладнити користування. Узагальнення переваг та недоліків чат-боту наведено в таблиці Г.6.

### **2.3 Опис реалізації та функцій телеграм чат-боту адміністративних послуг**

Ретельний аналіз існуючих ЦНАП телеграм чат-ботів дав можливість оцінити їхні переваги та недоліки з точки зору функціональності, ефективності та користувацького досвіду. Цей аналіз став важливим етапом у процесі формування концепції для власного продукту, надаючи чітке уявлення про те, які можливості та функції має мати новий телеграм бот [14].

Дослідження відзначило необхідність створення боту, який буде не лише функціональним, а й дружнім до користувача. Це вимагає врахування простоти взаємодії, зрозумілого дизайну та лаконічності у поданні інформації. Такий

підхід сприятиме залученню широкого кола аудиторії та забезпечить позитивні враження від користування ботом.

Окремою складовою успішного розроблення телеграм боту є аналіз помилок та недоліків, які були виявлені у аналогах. Врахування цих аспектів дозволяє уникнути повторення схожих помилок та покращити якість продукту.

Таким чином, на підставі аналізу існуючих рішень та виявлених недоліків, було розроблено телеграм бот, який буде відповідати потребам користувачів, забезпечуючи зручність, ефективність та задоволення від використання.

Далі будуть наведені основні функції телеграм бота та їх опис.

### **2.3.1 Головне меню**

При відкритті головної сторінки телеграм боту вас вітає привітальний текст, який ознайомлює з родом діяльності даного чат-боту. Тут ви зможете дізнатися про всі можливості та сервіси, які пропонує цей бот для вас.

Для початку спілкування з ним достатньо натиснути кнопку «Розпочати» або ввести команду «/start». Це відкриє доступ до основних функцій та послуг, які може надати бот.

Наступним кроком бот демонструє свої можливості через inline клавіатуру, що є дуже зручним та зрозумілим способом взаємодії. Ця клавіатура забезпечує простий та швидкий доступ до різноманітних функцій боту без необхідності пошуку головного меню у стрічці повідомлень під час спілкування.

Усі наступні функції та можливості також будуть реалізовані за допомогою цієї клавіатури, що робить користування ботом ще більш зручним та ефективним. Такий підхід сприяє покращенню вражень користувача та забезпечує максимальну доступність сервісів боту.

### **2.3.2 Загальна інформація**

Пункт меню "Загальна інформація" бота надає користувачам доступ до основних даних про центр надання адміністративних послуг. Обравши цей пункт, користувач отримує доступ до ключової інформації, такої як графік роботи та місцезнаходження центру.

Бот також надає можливість переглянути місце розташування центру на карті, що дозволяє зручно та швидко знайти його географічне положення та прокласти маршрут до нього.

### **2.3.3 Інформація по послугам**

Функція "Інформація по послугам" є важливим складовим елементом цього бота, яка надає користувачам доступ до обширної бази даних про найголовніші послуги, що надаються ЦНАП. Обравши цей пункт, користувач потрапляє на наступне меню, де послуги розділені за різними напрямками, такими як сімейні, транспортні, сільськогосподарські, спортивні тощо. Кожен напрям містить перелік доступних послуг.

У цій функції зберігається уся необхідна інформація, така як категорія заявників, необхідні документи для отримання послуги, а також вартість її надання.

### **2.3.4 Електронна черга**

Функція "Електронна черга" у головному меню дозволяє користувачам легко записатися на прийом до спеціаліста з будь-якого місця. При необхідності запису у чергу користувач спочатку обирає напрям послуги, на якому він бажає отримати обслуговування або консультацію з фахівцем.

Далі з бази даних обирається доступний час для реєстрації, щоб користувач міг обрати зручний для себе час та день візиту до центру надання адміністративних послуг. Виконавши усі необхідні дії, користувач отримує чек, в якому буде вказана вся інформація про зайняту чергу: вид послуги, дата та час запису, а також невелика інструкція щодо користування цим чеком.

### **2.3.5 Заповіти документ**

Пункт "Заповіти документ" у головному меню надає користувачам можливість віддалено замовити необхідні документи. Обравши цей пункт, користувач обирає потрібну довідку або документ серед запропонованих.

Після вибору документа користувач отримує покрокову інструкцію щодо того, як правильно зробити замовлення. Це включає в себе вказівки щодо

заповнення необхідної інформації та вимоги до документації, яка може бути потрібна для обробки замовлення.

Після завершення всіх необхідних кроків, запит надсилається на обробку. Користувач отримує повідомлення про те, що його запит прийнято, а також надається приблизний час очікування на відповідь.

### **2.3.6 Статус заявки**

Дана функція дозволяє користувачам відстежувати процес обробки їхніх заявок. Після обрання цього пункту меню, користувачу надається можливість вибрати конкретну заявку, яку він раніше подавав. Після вибору заявки користувач отримує інформацію про її поточний стан.

Ця функція дозволяє користувачам бути в курсі стану їхніх заявок без необхідності дзвінків або відвідування центру надання послуг. Вона спрощує процес комунікації між користувачами та організацією, що обробляє заявки, та підвищує рівень задоволення користувачів від надання послуг.

### **2.3.7 Контактна інформація**

Цей пункт меню надає користувачам зручний доступ до необхідних контактів та ресурсів. Обравши цей пункт, користувачам буде доступно всі необхідні посилання на популярні джерела, такі як Telegram, Facebook, YouTube, а також електронну пошту та веб-сайт.

Ця функція спрощує процес знаходження контактів та дозволяє користувачам легко зв'язатися зі службою підтримки або отримати доступ до додаткової інформації про продукт або послугу. Будь-які необхідні посилання і контактні дані знаходяться в одному місці, що робить процес комунікації більш зручним та ефективним для користувачів.

### **2.3.8 Структура ЦНАП**

Обираючи цей пункт меню, користувачам буде доступна інформація про робітників ЦНАП, включаючи їх повне ім'я, посаду та номер телефону. Це дає можливість трохи познайомитися з фахівцями перед візитом у ЦНАП для більш зручного та приємного подальшого спілкування з ними.

Також це допоможе зв'язатися з ними якщо з'являться якісь конкретні питання.

### **2.3.9 F.A.Q.**

Ця функція дозволяє користувачам отримати відповіді на найчастіші запитання, які зазвичай виникають у громадян. Обираючи цей пункт меню, користувачу надаються теми питань, які можуть його зацікавити.

Після вибору конкретної теми користувачу надаються всі найчастіші запитання цього напрямку, а також повні розгорнуті відповіді на них. Це спрощує процес отримання інформації для користувачів і дозволяє швидко та ефективно отримати відповіді на їхні запитання.

### **2.3.10 Статистика ЦНАП**

Функція "Статистика ЦНАП" у головному меню боту надає користувачам детальну інформацію про роботу центру надання адміністративних послуг. Обравши цей пункт, користувач отримує доступ до різноманітних показників, таких як кількість обслугованих громадян, кількість прийнятих заяв та інші статистичні дані.

Під час використання цієї функції користувач може отримати важливу інформацію щодо робочої потужності та ефективності роботи ЦНАП.

### **2.3.11 Вільні вакансії**

Цей пункт надає користувачам інформацію про наявні робочі місця на даний момент. Обравши цей пункт, користувач має можливість переглянути список вакансій.

Кожна вакансія має докладний опис, включаючи вимоги до кандидата, обов'язки, що покладаються на працівника, та умови праці. Крім того, надаються контактні дані та посилання для зв'язку з роботодавцем для отримання додаткової інформації чи подачі заявки на вакансію.

### **2.3.12 Переглянути петиції**

Функція "Переглянути петиції" у телеграм-боті надає користувачам можливість переглядати існуючі петиції, які створили інші громадяни. Обравши цей пункт меню, відкривається список доступних петицій.

Користувач може обрати петицію, подивитися її зміст та, за необхідності, підписати її. У кожній петиції вказується інформація про автора, дату подання, поточний статус, кількість зібраних підписів, кількість днів до завершення збору підписів, а також сам текст петиції.

Ця функція спрощує процес залучення громадян до підтримки важливих ініціатив та забезпечує зручний доступ до інформації про петиції, що цікавлять користувачів.

## 2.4 Модель комп'ютерної інформаційної системи надання адміністративних послуг

На основі переліку функцій телеграм боту була розроблена модель інформаційної системи надання адміністративних послуг із використанням нотації UML [10].

На діаграмі варіантів використання (Use Case) зображені функціональні вимоги до системи та як користувачі взаємодіятимуть із системою (рис. 2.5).

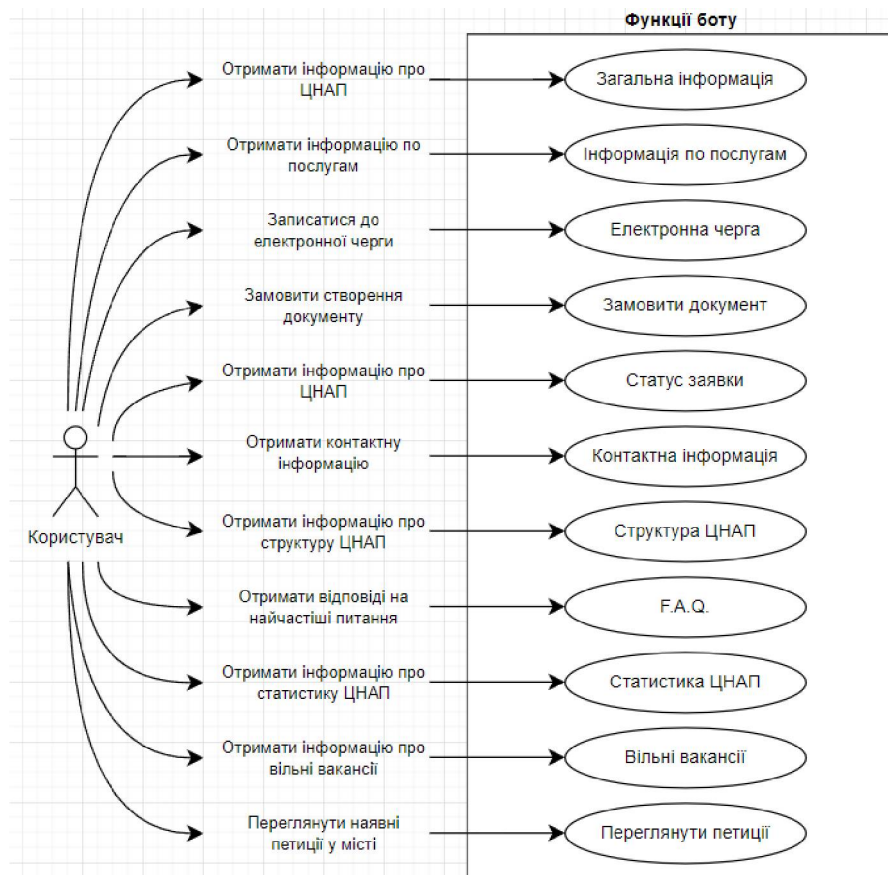


Рисунок 2.5 – Діаграма Use Case функцій чат-боту

На рис. 2.6 зображено Діаграму активності функції "Електронна черга" для віддаленої реєстрації в черзі за допомогою чат-боту, де кожна фаза процесу відображена з відповідними активностями та переходами.

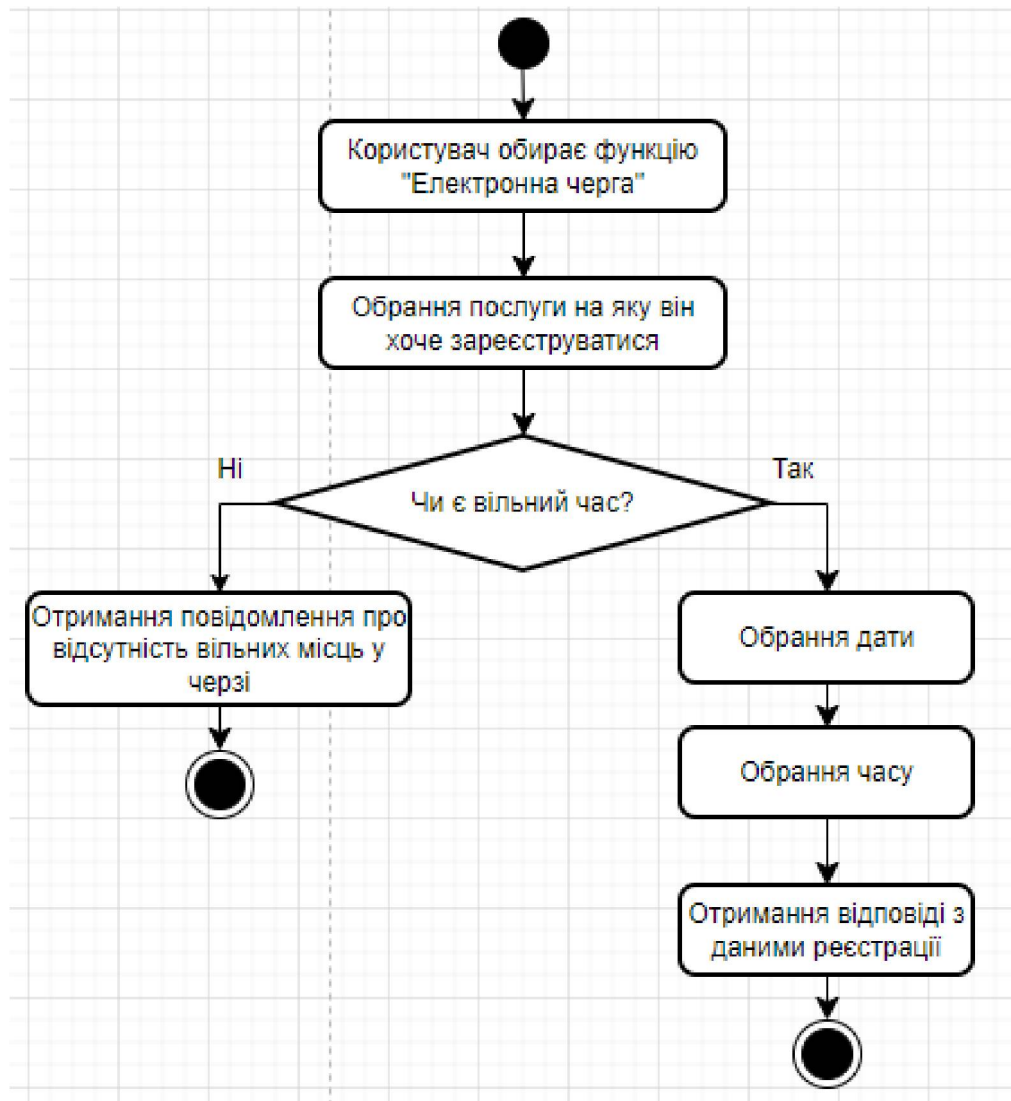


Рисунок 2.6 – Діаграма активності функції «Електронна черга»

## Висновки за розділом 2

У розділі 2 було детально розглянуто роботу ЦНАП (Центр надання адміністративних послуг) та його реалізацію за допомогою телеграм ботів. Проведений аналіз системи роботи та функцій ЦНАП дозволив нам краще зрозуміти особливості їх роботи та ідентифікувати області для покращень. Дослідження існуючих аналогів телеграм ботів дало можливість порівняти їхні

можливості та ефективність у вирішенні завдань, пов'язаних з наданням адміністративних послуг.

Опис реалізації та функцій телеграм чат-боту, який був розроблений в рамках даного дослідження, підкреслив його потенціал у полегшенні процесів отримання інформації та надання адміністративних послуг через інтерфейс месенджера. Новий чат-бот пропонує ряд функцій, спрямованих на оптимізацію взаємодії з ЦНАП та полегшення доступу громадян до послуг, таких як запис на прийом, отримання інформації про стан заявки, а також отримання необхідних документів та інструкцій.

У розділі було проведена оцінка поточного стану роботи ЦНАП та визначені слабкі місця, також запропоновані конкретні рішення для їх вдосконалення шляхом впровадження телеграм ботів, що відкриває перспективи для покращення доступності та ефективності надання адміністративних послуг громадянам.

## РОЗДІЛ 3.

### РОЗРОБКА ТЕЛЕГРАМ ЧАТ-БОТА АДМІНІСТРАТИВНИХ ПОСЛУГ

#### 3.1 Інструменти, використані для розробки телеграм бота

У цьому розділі буде представлений детальний огляд інструментів, мов програмування та бібліотек, які були використані в процесі розробки телеграм чат-боту адміністративних послуг. Кожен вибір був обдуманий та здійснений на основі вимог проекту та специфіки завдань. Описані технології відіграли важливу роль у створенні функціонального та ефективного бота, що забезпечує зручний інтерфейс та корисні можливості для користувачів.

У цьому розділі будуть розглянуті не лише конкретні інструменти та мови програмування, але й їхні особливості, переваги та обмеження у контексті розробки чат-ботів. Також будуть описані технологічні аспекти, які допомогли в створенні рішення з оптимальною швидкістю, надійністю та зручністю використання для кінцевих користувачів. Інструменти, які описані мною далі, дозволили легко взаємодіяти з API телеграму та іншими зовнішніми сервісами для розширення функціональності чат-боту.

##### 3.1.1 Мова програмування Python

Була обрана мова програмування Python для розробки телеграм чат-бота з кількох причин. По-перше, Python – це мова програмування, яка має простий синтаксис, що робить код легким для читання і розуміння. Коли йде робота над проектом, швидкість розробки – це ключовий аспект, і Python допомагає швидко перетворювати ідеї в реальність [2, 6, 7].

По-друге, Python має велику кількість бібліотек і фреймворків, що спрощує роботу з API та іншими інструментами, необхідними для розробки чат-ботів. Наприклад, для роботи з Telegram можна використовувати бібліотеку Telebot, яка забезпечує зручний інтерфейс для взаємодії з Telegram API.

Крім того, Python – це популярна мова серед розробників, тому є багато ресурсів, документації та спільнот, з якими можна звертатися за допомогою і підтримкою.

Основні переваги мови Python для розробки телеграм ботів:

- *Простота використання.* Python має приємний, легкий для вивчення синтаксис, який дозволяє швидко розробляти функціонал для телеграм-чат-ботів. Це особливо корисно для новачків у програмуванні, які швидко можуть отримати результати.
- *Багата кількість бібліотек.* Python має велику кількість бібліотек та фреймворків, призначених для розробки чат-ботів. Наприклад, бібліотека Telebot надає зручний інтерфейс для взаємодії з API Telegram та дозволяє легко створювати ботів та керувати ними.
- *Асинхронне програмування.* Python має підтримку асинхронного програмування, що дозволяє обробляти багато запитів одночасно без блокування виконання інших задач. Це особливо важливо для чат-ботів, які можуть отримувати одночасно багато повідомлень від користувачів.
- *Широкі можливості інтеграції.* Python легко інтегрується з іншими технологіями, такими як бази даних, API сторонніх сервісів, інструменти машинного навчання тощо. Це дозволяє створювати потужні та розширені функції для телеграм-чат-ботів, наприклад, обробку платежів, аналіз тексту або взаємодію з іншими сервісами.

Для розробки застосунку було використане середовище розробки PyCharm.

PyCharm – це інтегроване середовище розробки (IDE) для мови програмування Python, розроблене компанією JetBrains. Воно надає широкий набір інструментів для розробки програм на Python, включаючи підтримку віртуальних середовищ, автоматичне доповнення коду, рефакторинг, відлагодження, підтримку систем контролю версій та інші корисні функції [3].

### **3.1.2 Бібліотека Telebot**

Обираючи задовільну бібліотеку для розробки системи надання адміністративних послуг (ЦНАП) у Telegram, вибір пав на Telebot. Ця потужна бібліотека не лише надає зручний та ефективний інтерфейс для створення ботів, але й спрощує процес їх налаштування та управління. Використання Telebot

значно спростило процес розробки та підтримки чат-боту, забезпечивши швидку інтеграцію з різноманітними функціями та можливостями Telegram API [12]. Також дана бібліотека була обрана завдяки своїй простоті використання, багатому функціоналу та активній спільноті користувачів. Використання Telebot дозволило зосередитися на створенні функціональних та інноваційних рішень для чат-боту, забезпечуючи високу якість та ефективність взаємодії з користувачами у месенджері Telegram [11].

Основні переваги використання Telebot:

- *Підтримка функціональності Telegram Bot API.* Telebot забезпечує доступ до всіх можливостей Telegram Bot API, включаючи роботу з повідомленнями, клавіатурами, медіафайлами тощо.
- *Гнучкість і розширюваність.* Бібліотека дозволяє розширювати функціональність бота за допомогою власних обробників повідомлень та команд.
- *Підтримка асинхронності.* Telebot може працювати в асинхронному режимі, що дозволяє оптимізувати роботу бота та обробляти багато запитів одночасно за допомогою асинхронного програмування в Python.
- *Можливість реалізації складної логіки.* Telebot надає гнучкі можливості для реалізації складної логіки чат-бота, такої як зберігання даних, обробка запитів, взаємодія з базами даних тощо.
- *Підтримка різних типів повідомлень і медіафайлів.* Telebot дозволяє робити ботів, які можуть обробляти різні типи повідомлень (текстові, голосові, фото, відео тощо) і відправляти їх користувачам.
- *Можливість інтеграції з іншими сервісами.* Telebot легко інтегрується з іншими сервісами та API, що дозволяє розширити можливості чат-бота.

### **3.1.3 Telegram Bot API**

Серед великої кількості доступних інструментів за допомогою яких можна реалізувати чат-бот був обраний Telegram Bot API. Telegram Bot API[5, 13] – це інтерфейс програмування додатків, який надає Telegram для створення і взаємодії з чат-ботами на їхній платформі. З його допомогою можна отримати широкий

спектр інструментів і можливостей для створення боту, який взаємодіє з користувачами через текстові повідомлення, медіафайли, кнопки, графічні інтерфейси та інші елементи.

Переваги використання Telegram Bot API для розробки чат-боту [5, 13]:

- *Багатофункціональність.* Telegram Bot API дозволив створити бота з різноманітними функціями, від простих відповідей на текстові повідомлення до складних інтерактивних додатків.

- *Швидкість розробки.* Інтерфейс Telegram Bot API має зрозумілу документацію та простий інтерфейс програмування, що дозволяє швидко створювати та впроваджувати чат-ботів без значних зусиль.

- *Зручний інтерфейс.* Telegram має інтуїтивно зрозумілий інтерфейс для користувачів, що сприяє зручній та комфортній взаємодії з чат-ботами.

- *Підтримка медіафайлів.* Telegram Bot API дозволяє ботам обмінюватися різними типами медіафайлів, такими як зображення, відео, аудіо, документи тощо.

- *Безпека.* Telegram відомий своєю фокусуванням на конфіденційності та безпеці користувачів, що робить його відмінним вибором для розробки чат-ботів, які обробляють конфіденційну інформацію.

- *Можливість інтеграції з іншими сервісами.* Telegram Bot API підтримує інтеграцію з різними сервісами і додатками, що дозволяє розширити функціональність та автоматизувати різноманітні завдання.

### **3.2 Аналіз та демонстрація роботи телеграм чат-боту адміністративних послуг**

У даному розділі буде розглянутий функціонал застосунку, звернена особлива увага на ключові можливості та властивості. Будуть наведені приклади роботи деяких функцій та їх розбір. Також будуть наведені приклади роботи деяких ключових функцій. Зроблений детальний аналіз функцій, щоб зрозуміти принцип їх дії та практичне застосування.

### 3.2.1 Функція «Інформація по послугам»

Дана функція є стартовою точкою роботи зі створеним чат-ботом. На рис. 3.1 продемонстровано меню даної функції.

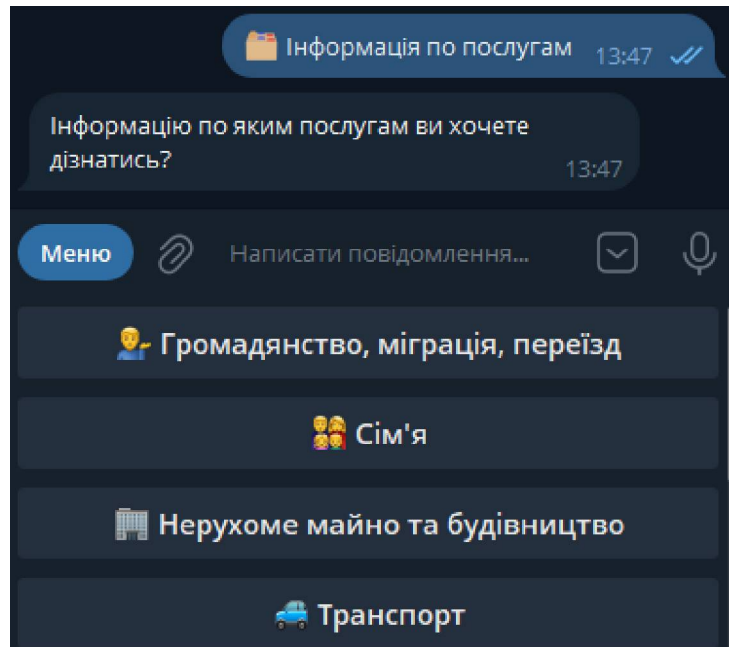


Рисунок 3.1 – Меню вибору виду послуг

Ця корисна функція подібна до центральної бази даних, що містить перелік усіх необхідних послуг для громадян. Вона створена з метою забезпечення легкого доступу до видів послуг, які можуть знадобитися користувачам. Крім того, до кожної послуги докладно додається опис, що містить всю необхідну для громадянина інформацію. Це робить використання чат-боту надзвичайно зручним та ефективним, адже громадяни можуть швидко знайти потрібну їм інформацію, не втрачаючи часу на її пошук у інших джерелах.

Наведений приклад ілюструє систематизацію усіх наданих послуг за їх тематикою, що дозволяє спростити процес навігації користувача в межах даної функції. Після вибору відповідної тематики користувач потрапляє до наступного рівня меню, де вже пропонуються конкретні назви послуг з обраної тематики.

На рис. 3.3 продемонстровано наступний рівень меню даної функції.

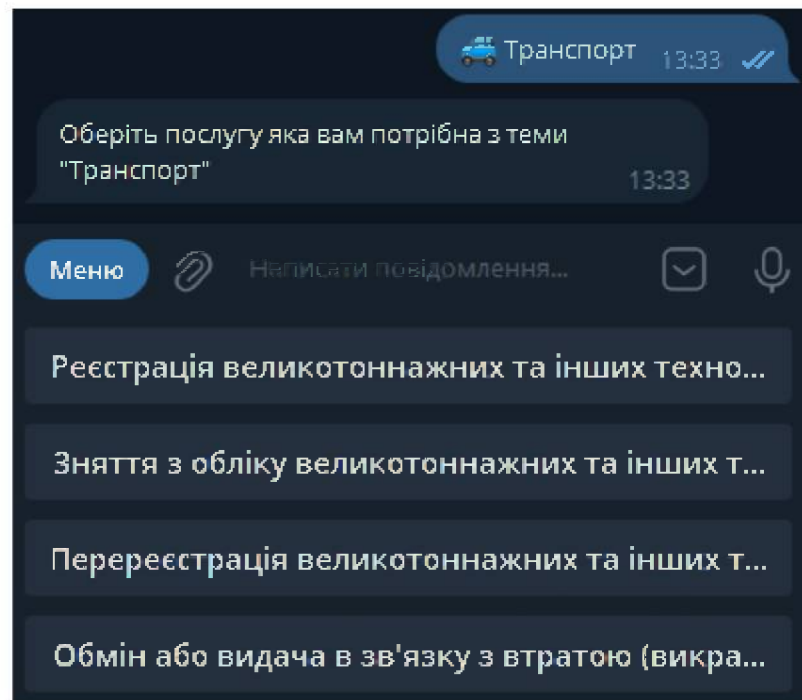


Рисунок 3.2 – Меню послуг з теми «Транспорт»

На рис. 3.2 можна побачити вже конкретні послуги, які може надати державна установа. Нарешті обравши конкретну послугу, користувач отримує більш детальний опис найважливіших аспектів цієї послуги у формі, яка зрозуміла і доступна для нього. Цей опис допомагає громадянам краще розуміти суть та умови надання послуги, сприяючи більш ефективному використанню державних сервісів.

На рис. 3.3 продемонстровано у якому форматі користувач отримає потрібну йому інформацію.

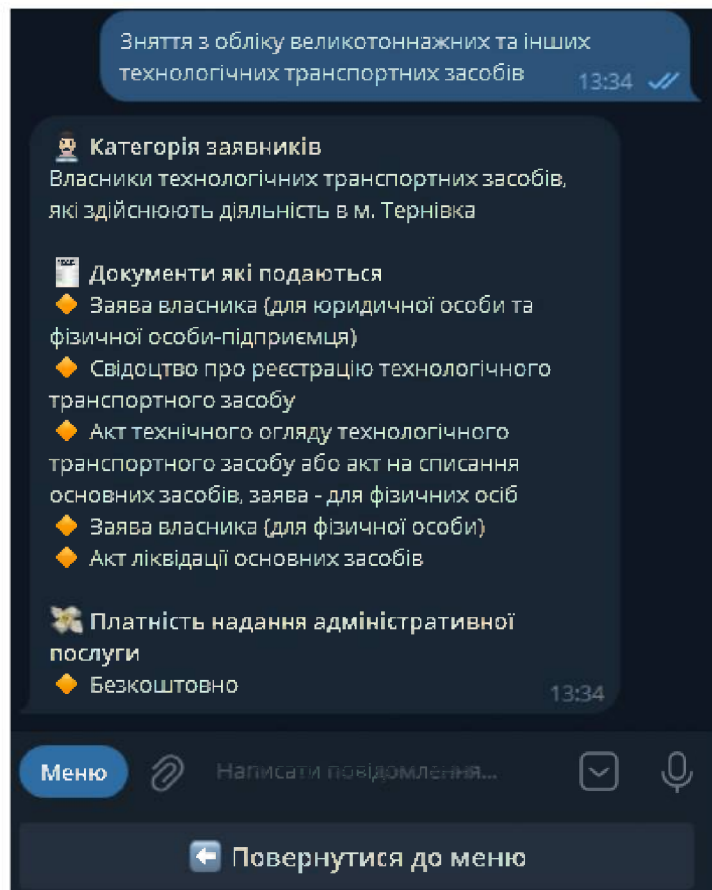


Рисунок 3.3 – Інформація з послуги «Зняття з обліку великотоннажних та інших технологічних транспортних засобів»

### 3.2.2 Функція «Електронна черга»

Далі перейдемо до демонстрації ще однієї надзвичайно важливої функції телеграм чат-боту, яка відкриває перед користувачами можливість реєстрації в чергу для отримання послуг або консультацій з будь-якого місця, де є доступ до Інтернету. Ця функція дозволяє забезпечити максимальну зручність для громадян, дозволяючи їм уникнути витрати часу на особистий візит до офісу чи установи. Вона відкриває можливості для ефективного планування часу та оптимізації процесів надання послуг, сприяючи покращенню якості обслуговування та задоволенню потреб громадян.

На рис. 3.4 продемонстровано меню даної функції.

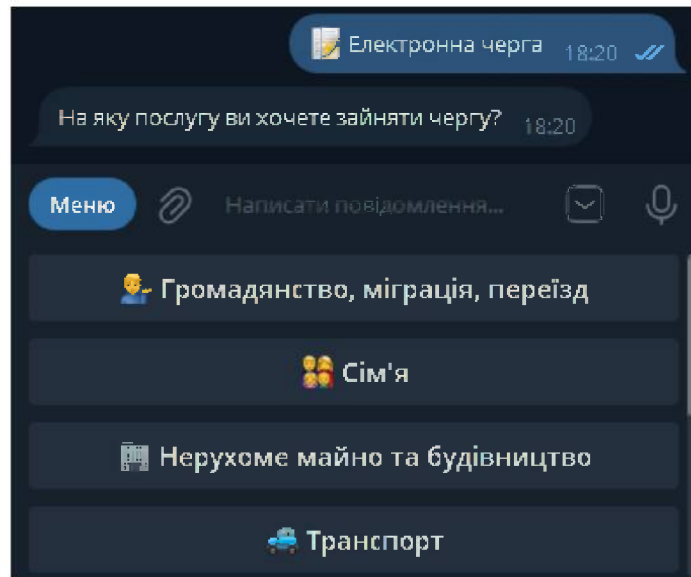


Рисунок 3.4 – Меню вибору виду послуг

Меню цієї функції повністю ідентичне з меню у попередньому прикладі, і це не дивно. Вибір типу послуги, на яку користувач бажає записатися в чергу для отримання допомоги чи консультації, є першим кроком у використанні цієї функції. Така структура меню дозволяє забезпечити зручну та логічну навігацію для користувачів, допомагаючи їм швидко знайти потрібний сервіс та скористатися ним.

На рисунках 3.5 та 3.6 можна побачити наступні кроки для успішного запису до черги.

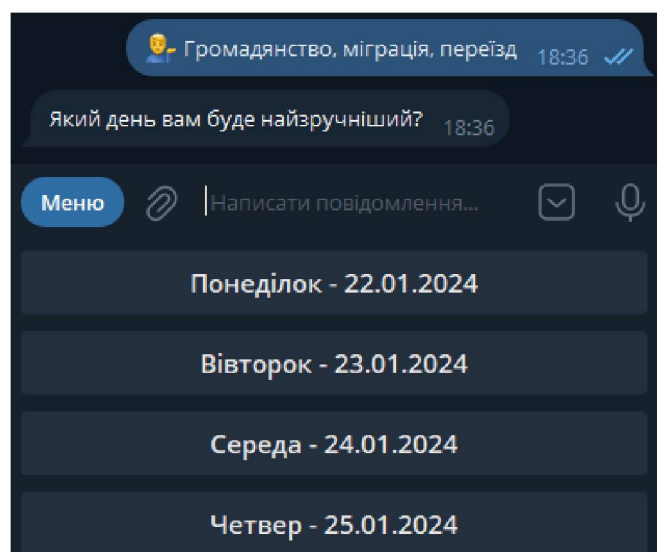


Рисунок 3.5 – Меню вибору дати прийому

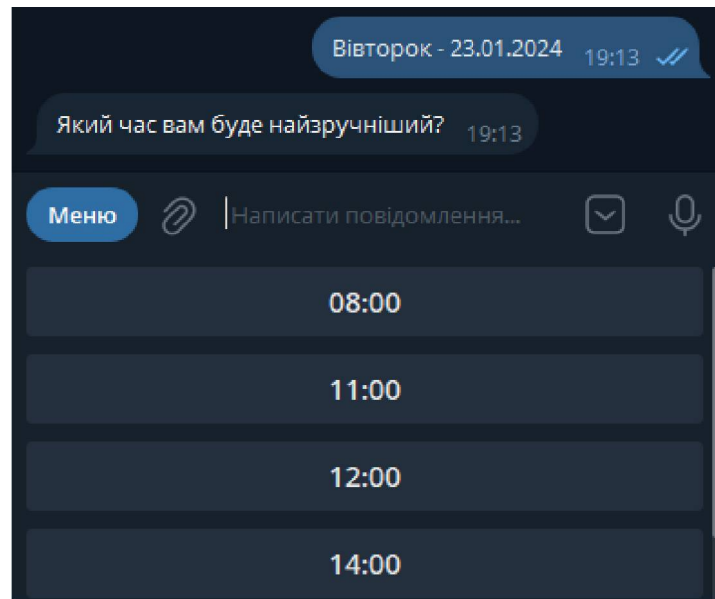


Рисунок 3.6 – Меню вибору часу прийому

Після успішного проходження всіх перерахованих кроків для запису на прийом до державної установи з надання адміністративних послуг, користувач отримає відповідне повідомлення-чек. Цей чек є свідченням про заброньоване місце у черзі, що дозволяє гарантувати право на прийом. Підтвердивши свою присутність та маючи з собою це повідомлення, користувач матиме можливість зачекати свою чергу та отримати необхідну адміністративну допомогу. Цей чек є підставою для потрапляння до приймальні, де фаховий працівник установи буде готовий надати послуги або проконсультувати по питанням за темою, яку користувач обрав у застосунку.

На рис. 3.7 продемонстровано відповідь чат-боту у вигляді чеку який необхідний для дозволу на прийом.

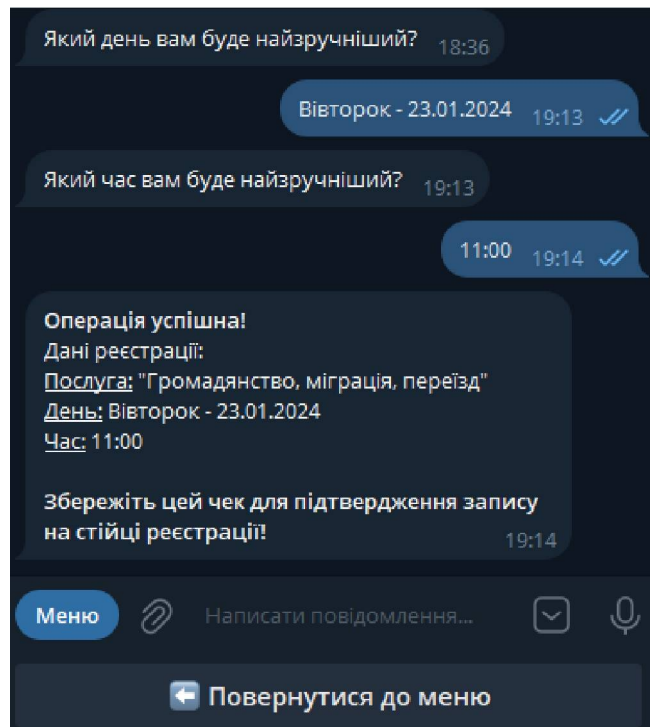


Рисунок 3.7 – Повідомлення-чек

### 3.3 Загальна характеристика чат-боту

Узагальнюючи усі ті теми, які було розкрито у цьому розділі про телеграм чат-бот адміністративних послуг, можна зазначити його ключову роль у забезпеченні доступу до державних послуг та консультацій для громадян. Його широкий функціонал дозволяє користувачам оперативно отримати інформацію про доступні послуги, записатися в чергу для отримання необхідної допомоги, отримувати важливі повідомлення та підтримку та ще багато іншого. Чат-бот є важливим інструментом для поліпшення доступності та ефективності державних сервісів, що дозволяє громадянам зекономити час та зусилля при взаємодії з установами влади. Його спрощена навігація та інтуїтивно зрозумілий інтерфейс, який був спроектований та реалізований для підвищення якості обслуговування та задоволення потреб користувачів, дає можливість громадянам ефективно користуватися всіма функціями чат-боту без зайвих зусиль та запитань.

У таблиці 3.1 буде наведено перелік переваг застосунку відносно аналогів які були проаналізовані у попередньому розділі.

Таблиця 3.1

## Переваги запропонованого рішення

Характеристика	Перевага
Дизайн	Чудовий та сучасний дизайн, що приваблює користувачів своєю естетикою та стильністю. Він створює приємну атмосферу та робить взаємодію з чат-ботом більш привабливою та приємною.
Ієрархія	Чітка та логічна ієрархія, яка допомагає користувачам легко орієнтуватися в системі. Кожен елемент інтуїтивно розташований, що спрощує навігацію та робить користування ботом зручним та ефективним.
Навігація	Інтуїтивно зрозуміла та проста навігація, яка дозволяє користувачам легко переміщатися між різними функціями та розділами. Вона забезпечує зручність взаємодії та допомагає швидко знаходити потрібну інформацію.
Кількість функціоналу	Розширений функціонал, що надає користувачам широкий спектр можливостей та послуг. Велика кількість доступних функцій робить чат-бот універсальним інструментом для вирішення різноманітних завдань та потреб.
Користь функціоналу	Функціонал чат-боту забезпечує не тільки велику кількість а ще й необхідну корисність важливих інформаційних ресурсів та послуг, що сприяє оптимізації їх часу та зусиль. Він робить взаємодію з державними службами більш ефективною та зручною для громадян.

### 3.4 Актуальність застосунку

Актуальність даного телеграм чат-боту привертають увагу завдяки ряду ключових факторів. Він виходить за межі звичайного інтерфейсу, пропонуючи широкий спектр функцій, які відповідають поточним потребам користувачів. Розробка чат-боту була попередньо підготовлена шляхом аналізу та узагальнення усіх можливих аналогів на ринку, що дозволило врахувати як їхні позитивні аспекти, так і недоліки. Кожна функція була відібрана з урахуванням реальних потреб користувачів, щоб забезпечити найбільш ефективний та зручний досвід використання. Такий підхід дозволяє не лише відповісти на сучасні вимоги ринку, але й перевершити їх, забезпечуючи користувачам надзвичайно комфортне та продуктивне взаємодію з чат-ботом.

### Висновки за розділом 3

У цьому розділі, детально розглядалися ключові аспекти вибору технологій. Пояснено, які конкретно інструменти та бібліотеки були використані,

та як вони відповідають вимогам проекту. Надано аргументи, чому саме ці засоби обрані, та як вони сприяють досягненню цілей розробки.

Крім того, опис користувацького інтерфейсу застосунку дав змогу поглиблено розглянути внутрішній механізм взаємодії з користувачем. Визначені основні елементи інтерфейсу, їх функціональність та способи використання.

Аналіз застосунку та демонстрація прикладів роботи відобразили реальний функціонал та можливості чат-боту. Це дозволило зрозуміти, як він працює у практичних умовах та як він може бути корисним для користувачів.

Обґрунтування переваг та актуальності рішення відобразило важливість та конкурентні переваги розробленого чат-боту порівняно з іншими аналогами. Вказано, як саме він вирізняється та чому він є кращим вибором для користувачів. Такий аналіз допомагає чітко визначити переваги та цінність застосунку в порівнянні з конкурентами.

## ВИСНОВОК

Під час виконання кваліфікаційної роботи був проведений ретельний аналіз можливостей телеграм ботів. Також були розглянуті різноманітні інструменти та технології, що дозволяють швидко та ефективно створювати ботів з різними функціональними можливостями. Крім того, було досліджено ефективність телеграм ботів у галузі надання адміністративних послуг. Вони виявилися цінним інструментом, що забезпечує швидкий та зручний доступ до інформації та послуг для користувачів.

Наступним кроком був проведений детальний аналіз системи роботи та функцій ЦНАП який дозволив отримати глибше розуміння процесів, які можна оптимізувати та автоматизувати за допомогою телеграм ботів. Дослідження існуючих аналогів телеграм ботів надало можливість визначити переваги та недоліки існуючих рішень і врахувати їх при розробці чат-боту для адміністративних послуг. Опис реалізації та функцій телеграм чат-боту відобразив можливості та переваги використання такого інструменту у контексті адміністративних послуг. Зазначені функції та можливості дозволили забезпечити швидку та зручну взаємодію з громадянами та підвищити якість надання адміністративних послуг.

Врешті-решт було розроблено телеграм чат-бот для Центру надання адміністративних послуг міста Тернівка. Цей інноваційний інструмент призначений не лише для полегшення робочих процесів у державній установі, але й для спрощення процедур отримання адміністративних послуг громадянами. Шляхом впровадження чат-бота з розширеним функціоналом, який включає в себе різноманітні опції та можливості, відповідальність за деякі обов'язки працівників ЦНАП буде виключена або відчутно зменшена. Зручний та інтуїтивно зрозумілий інтерфейс чат-бота дозволить користувачам отримати доступ до необхідних послуг швидко та ефективно. Крім того, велика кількість функцій, реалізованих у чат-боті, сприятиме автоматизації та оптимізації багатьох процесів, що підвищить продуктивність та ефективність роботи ЦНАП.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. EMC Education Services. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. Wiley 2015. 432с.
2. Eric Matthes. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. No Starch Press 2015. 560с.
3. Getting started | PyCharm documentation [Електронний ресурс] – режим доступу URL: <https://www.jetbrains.com/help/pycharm/getting-started.html> (дата звернення 27.02.2024).
4. MySQL Documentation [Електронний ресурс] – режим доступу URL: <https://dev.mysql.com/doc/> (дата звернення 10.02.2024).
5. Nicolas Modrzyk. Building Telegram Bots: Develop Bots in 12 Programming Languages using the Telegram Bot API. Apress 2018. 319с.
6. Python 3.11.8 documentation [Електронний ресурс] – режим доступу URL: <https://docs.python.org/uk/3.11/> (дата звернення 3.02.2024).
7. Python Developer's Guide [Електронний ресурс] – режим доступу URL: <https://devguide.python.org/> (дата звернення 5.02.2024).
8. Rick Silva. MySQL Crash Course: A Hands-on Introduction to Database Development. No Starch Press 2023. 323с.
9. Sqlite3 — DB-API 2.0 interface for SQLite databases [Електронний ресурс] – режим доступу URL: <https://docs.python.org/3/library/sqlite3.html> (дата звернення 13.02.2024).
10. Suriya Sundaramoorthy. UML Diagramming: A Case Study Approach. Auerbach Publications 2022. 416с.
11. TeleBot version – pyTelegramBotAPI 4.18.1 documentation [Електронний ресурс] – режим доступу URL: [https://pytba.readthedocs.io/en/latest/sync\\_version/index.html](https://pytba.readthedocs.io/en/latest/sync_version/index.html) (дата звернення 18.03.2024).
12. Telegram APIs [Електронний ресурс] – режим доступу URL: <https://core.telegram.org/> (дата звернення 20.02.2024).

13. Telegram Bot API [Електронний ресурс] – режим доступу URL: <https://core.telegram.org/bots/api> (дата звернення 24.02.2024).
14. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. Introduction To Algorithms. MIT Press 2001. 1180с.
15. Victor Hugo. Histoire D'un Crime: Déposition D'un Témoin, Volumes 1-2 (French Edition) Hardcover. Legare Street Press 2023. 344с.
16. Портал адміністративних послуг м. Чернігів [Електронний ресурс]– режим доступу URL: <https://www.admincher.gov.ua/> (дата звернення 17.02.2024).
17. Центр надання адміністративних послуг м. Києва [Електронний ресурс]– режим доступу URL: <https://kyivcnap.gov.ua/> (дата звернення 15.02.2024).
18. Центр надання адміністративних послуг м. Харкова [Електронний ресурс]– режим доступу URL: <https://dozvil.kh.ua/> (дата звернення 16.02.2024).
19. ЦНАП м. Черкаси [Електронний ресурс] – режим доступу URL: <https://cnapck.gov.ua/> (дата звернення 18.02.2024).

**ДОДАТКИ****Додаток А**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки  
Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр  
Галузь знань: 12 – Інформаційні технології  
Спеціальність: 123 – Комп'ютерна інженерія

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри теоретичної  
та прикладної системотехніки  
 д.т.н., проф. Шматков С. І.  
«21» грудня 2024 року

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

**Новікова Олексія Едуардовича**  
(прізвище, ім'я, по батькові студента)

1. Тема роботи **«Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту»**

керівник роботи Стрілець Вікторія Євгенівна, канд. техн. наук, доцент  
(прізвище, ім'я, по батькові, науковий ступінь, місце знання)

затверджені наказом по університету від «03» травня 2024 року № 4101-5/909

2. Строк подання студентом роботи 31 травня 2024 року

3. Перелік питань, які потрібно розробити

- 1) Аналіз існуючих моделей інформаційних систем із використанням чат-ботів, принципів їх роботи та структури.
- 2) Дослідження особливостей інформаційних систем надання адміністративних послуг.
- 3) Побудова моделі інформаційної системи надання адміністративних послуг із використанням чат-боту.
- 4) Програмна реалізація моделі інформаційної системи на основі чат-боту для надання адміністративних послуг та її тестування.

## 4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Визначення особливостей створення інформаційних систем на основі чат-ботів, вибір мови програмування та бібліотек	21.12.2023 – 05.01.2024
2	Аналіз існуючих інформаційних систем надання адміністративних послуг та їх функціоналу	06.01.2024 – 20.01.2024
3	Побудова моделі інформаційних систем надання адміністративних послуг із використанням чат-боту	21.01.2024 – 29.01.2024
4	Реалізація моделі інформаційних систем надання адміністративних послуг	30.01.2024 – 10.03.2024
5	Тестування й аналіз повноти моделі, додавання функціоналу до чат-боту	11.03.2024 – 20.03.2024
6	Оформлення рекомендації та спрощення використання чат-боту	21.03.2024 – 28.03.2024
7	Тестування усіх функцій та виправлення помилок	29.03.2024 – 10.04.2024
8	Оформлення пояснювальної записки	11.04.2024 – 15.05.2024
9	Представлення кваліфікаційної роботи керівнику та рецензенту	16.05.2024 – 21.05.2024
10	Оформлення супроводжувальної документації	22.05.2024 – 27.05.2024
11	Оформлення звіту за результатами переддипломної практики	22.05.2024 – 05.06.2024

## 5. Дата видачі завдання 21.12.2023

Студент

Новіков О. Е.

ініціали, прізвище



підпис

Керівник роботи

Стрілець В. Є.

ініціали, прізвище



підпис

Затверджую

« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

**Технічне завдання**

**на розробку програмного виробу «Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту».**

1.	Введення	<p>1.1 Назва програмного виробу – Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту.</p> <p>1.2. Галузь застосування: Інформаційні технології</p>
2.	Підстава для розробки	<p>2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія</p> <p>2.2. Завдання на кваліфікаційну роботу бакалавра № <u>4101-5/909</u> від «03» <u>травня</u> 2024 року (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).</p>
3.	Призначення розробки	<p>3.1. Мета розробки: створення додатку для надання адміністративних послуг за допомогою телеграм боту.</p> <p>3.2. Призначення розробки: можливість використання додатку громадянами для зручного та доступного надання адміністративних послуг.</p> <p>3.3. Вхідні дані: послуга, тематика послуг, дата та час.</p> <p>3.4. Вихідні дані розробки: доступ до послуг, інформаційна допомога, надсилання й отримання документів, запис у чергу.</p>
4.	Технічні вимоги до програмного виробу	<p>4.1. Вимоги до функціональних характеристик: покращення зручності й доступності надання адміністративних послуг, наявність усіх функцій які можуть знадобитися громадянину.</p> <p>4.2. Вимоги до надійності: забезпечення безперервної роботи телеграм чат-боту при будь-яких вимогах користувача в рамках призначення виробу та обмеження доступу до бази даних.</p> <p>4.3. Вимоги до умов експлуатації: доступ до глобальної мережі Internet.</p> <p>4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із будь-якою операційною системою сімейства Windows, Linux, Mac OS, Android. Крім того, завантажений на пристрій Telegram.</p>

		<p>4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows, підтримка мови програмування та встановлений додаток Telegram.</p> <p>4.6. Вимоги до маркування та упаковки: вимоги до маркування та упакування не представляються.</p> <p>4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються.</p> <p>4.8. Спеціальні вимоги: Збереження персональних даних.</p>	
5.	Вимоги до програмної документації	<p>Програмною документацією до виробу «Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту» вважати:</p> <p>1) Ознайомчу сторінку з правилами користування додатком.</p> <p>2) Програму і методику випробувань розробленої програми (представити як додаток В до пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).</p>	
6.	Вимоги до техніко-економічних показників	<p>Програмною документацією до виробу «Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Опис програмного виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Джерела базової інформації.</p>	
7.	Стадії і етапи розробки	Дата	Назва етапу
		21.12.2023 – 05.01.2024	Визначення особливостей створення інформаційних систем на основі чат-ботів, вибір мови програмування та бібліотек.
		06.01.2024 – 20.01.2024	Аналіз існуючих інформаційних систем надання адміністративних послуг та їх функціоналу
		21.01.2024 – 29.01.2024	Побудова моделі інформаційних систем надання адміністративних послуг із використанням чат-боту

		30.01.2024 – 10.03.2024	Реалізація моделі інформаційних систем надання адміністративних послуг
		11.03.2024 – 20.03.2024	Тестування й аналіз повноти моделі, додавання функціоналу до чат-боту
		21.03.2024 – 28.03.2024	Оформлення рекомендації та спрощення використання чат-боту
		29.03.2024 – 10.04.2024	Тестування усіх функцій та виправлення помилок
		11.04.2024 – 15.05.2024	Оформлення пояснювальної записки
		16.05.2024 – 21.05.2024	Представлення кваліфікаційної роботи керівнику та рецензенту
		22.05.2024 – 27.05.2024	Оформлення супроводжувальної документації
8.	Порядок контролю і приймання програмного продукту (моделі)	1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати в електронному вигляді в 1 примірнику та в роздрукованому вигляді.	

Виконавець  
Студент групи КІ-41  
Новіков О. Е.



Замовник  
канд. техн. наук  
Стрілець В. Є.



## **Програма і методика випробувань програмного виробу**

«Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту»

### **1. Об'єкт випробувань**

1. Назва програмного виробу: «Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту».

2. Галузь застосування: Інформаційні технології

3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

### **2. Мета випробувань**

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

### **3. Загальні положення**

3.1 Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2 Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3.3 Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавці та інших осіб, присутніх на засіданні.

### **4. Вимоги до програми або програмного виробу**

4.1. Вимоги до функціональних характеристик: покращення зручності й доступності надання адміністративних послуг, наявність усіх функцій які можуть знадобитися громадянину.

4.2. Вимоги до надійності: забезпечення безперервної роботи телеграм чат-боту при будь-яких вимогах користувача в рамках призначення виробу та обмеження доступу до бази даних.

4.3. Вимоги до умов експлуатації: доступ до глобальної мережі Internet.

4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із будь-якою операційною системою сімейства Windows, Linux, Mac OS, Android. Крім того, завантажений на пристрій Telegram.

4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows, підтримка мови програмування та встановлений додаток Telegram.

4.6. Вимоги до маркування та упаковки: вимоги до маркування та упакування не представляються.

4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються.

4.8. Спеціальні вимоги: Збереження персональних даних.

## **5. Вимоги до програмної документації**

Програмною документацією до виробу «Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту» вважати:

1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).

2) Опис програмного виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).

3) Джерела базової інформації.

## 6. Засоби і порядок випробувань

### 6.1. Засоби випробувань

Засоби випробувань представлено на ПК на яких встановлено наступні програмні засоби: Telegram.

### 6.2. Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- Ознайомчий (1-й етап);
- Власне випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

- 1) Перевірку комплектності складу програмної документації здійснюються за критерієм наявності зазначеної в ТЗ документації
- 2) Перевірка якості програмної документації. Перевірку здійснювати за критерієм відповідності вимогам ГОСТ 19.301-79 ЕСПД «Програма і методика випробувань».

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

- 1) Перевірку відповідності технічних характеристик програми вимогам технічного завдання.
- 2) Перевірку ступеня виконання функціональних вимог до програми.
- 3) Методику проведення перевірок:
  - а) Запустити Telegram.
  - б) Порядок проведення випробувань:
    - У пошуку Telegram ввести назву бота.
    - Знайти бот, відкрити, почати спілкування з ним.
    - Перевірити чи коректно працює бот чи виконує усі доступні функції.

Якщо перевірки на першому та другому етапах виконано успішно, то виріб вважається таким, що пройшов випробування.

Для проведення випробувань пропонується тест 1, тест 2, тест 3.

#### Тест 1

1. Перевірка виконання програми;
2. Перевірка роботи головного меню;
3. Поява усіх функцій бота й привітального повідомлення.

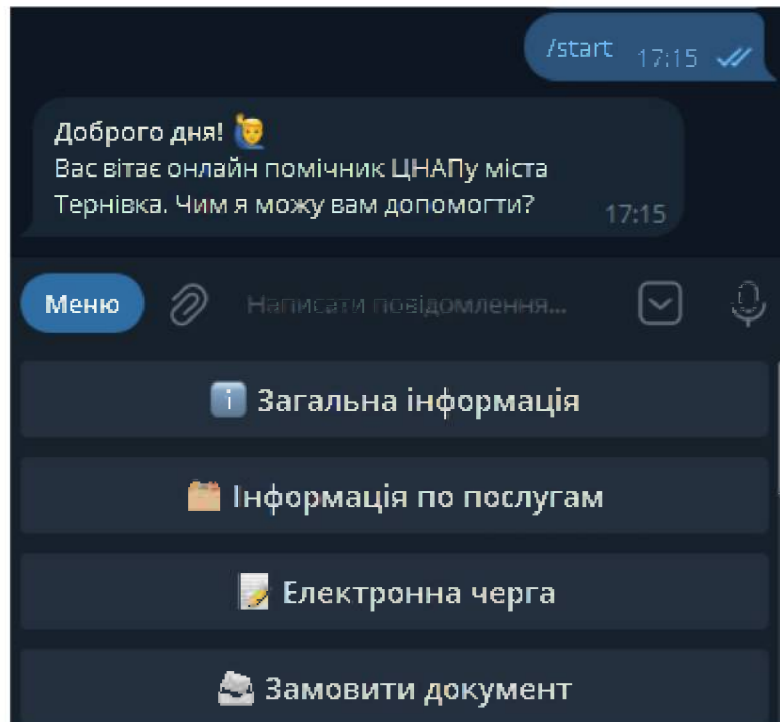


Рис. В.1 Тест 1

Тест вважається успішним якщо після введення початкової команди користувач отримує привітальний текст та меню усіх функцій чат-боту.

#### Тест 2

1. Перевірка виконання програми
2. Перевірка виводу загальної інформації про ЦНАП
3. Отримання результату

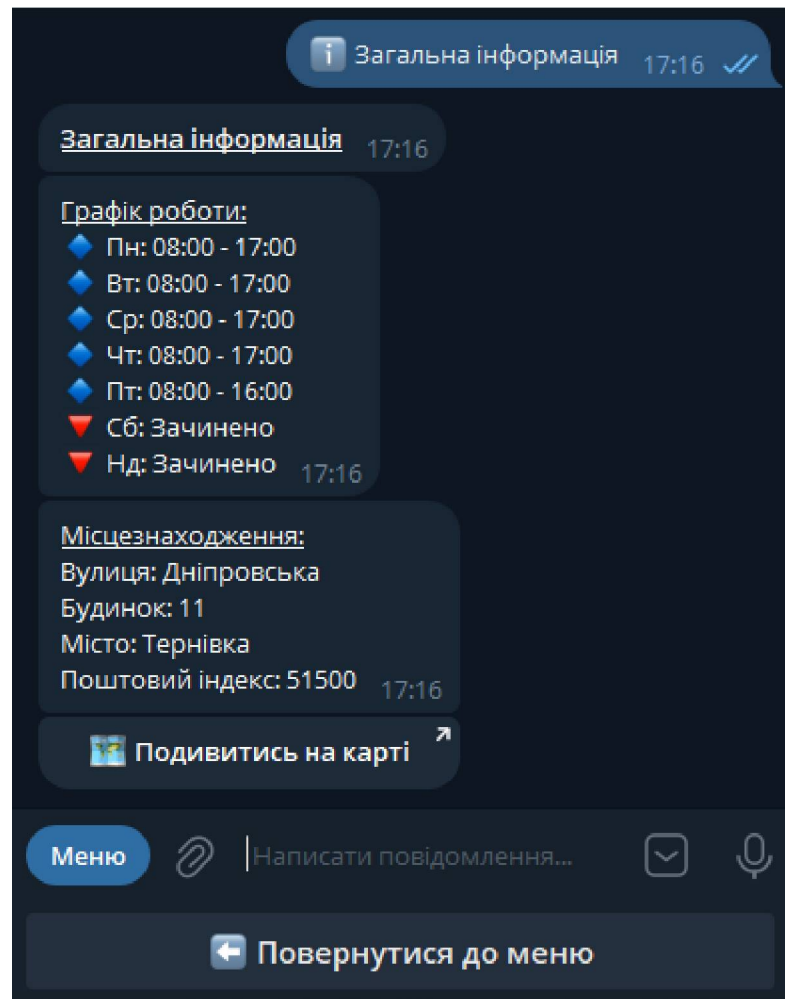


Рис. В.2 Тест 2

Тест вважається успішним якщо після натискання на кнопку меню «Загальна інформація» користувач отримує повідомлення з відповідним змістом.

### Тест 3

1. Перевірка виконання програми
2. Перевірка реєстрації користувача до електронної черги
3. Отримання результату

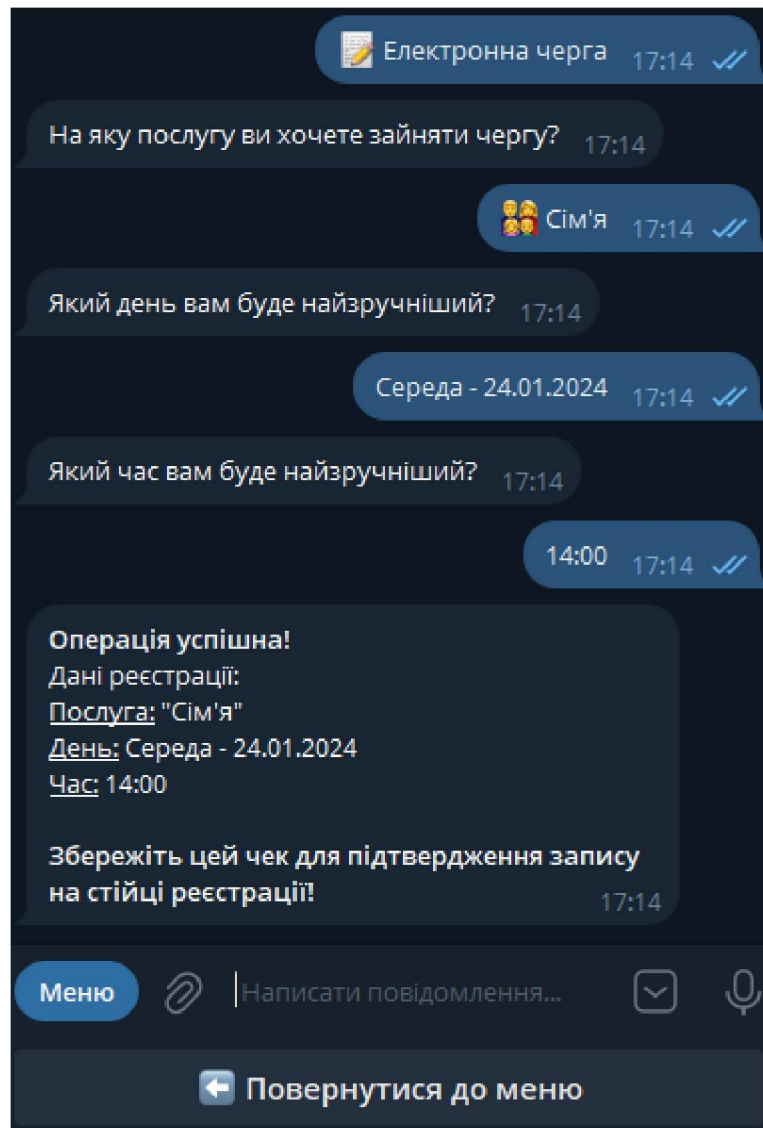


Рис. В.3 Тест 3

Тест вважається успішним якщо після натискання на кнопку меню «Електронна черга», обрання потрібної теми, вибору дати та часу користувач отримує повідомлення щодо успішності проведення операції.

**Висновки:** тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець: студент групи КІ41, Новіков О. Е.

Переваги та недоліки існуючих інструментів та технологій для розробки чат-ботів

Таблиця Г.1

Переваги та недоліки різних мов програмування

Мова програмування	Переваги	Недоліки
Python	Простота вивчення, Велика кількість бібліотек; Висока продуктивність розробки; Широке застосування у сфері штучного інтелекту.	Може бути менш ефективною для великих обсягів даних; Обмежена підтримка мультипотокості.
Node.js (JavaScript)	Популярність у веб-розробці; Багато модулів та пакетів; Можливість використання однієї мови на клієнті та сервері.	Потребує використання більше ресурсів; Менша надійність.
Java	Висока надійність; Масштабованість; Велика кількість готових бібліотек; Об'єктно-орієнтований підхід; Підтримка різних платформ.	Більш громіздка синтаксично; Менш гнучка порівняно з іншими мовами.
PHP	Зручність у веб-розробці; Легкість в освоєнні; Багато готових фреймворків; Підтримка різних баз даних.	Менша швидкодія; Менша надійність; Менш гнучка порівняно з іншими мовами.
Ruby	Простий та елегантний синтаксис; Зручність у веб-розробці; Активна спільнота.	Менша популярність; Менша швидкодія

Таблиця Г.2

## Переваги та недоліки різних мов баз даних

Мова баз даних	Переваги	Недоліки
SQLite	Легкість використання; Вбудована база даних, яка не вимагає окремого сервера; Підтримка SQL-синтаксису.	Менш підходить для великих обсягів даних; Обмежені можливості масштабування.
MySQL	Висока швидкодія; Велика спільнота користувачів; Підходить для великих обсягів даних; Підтримка широкого спектру операцій та функцій.	Може бути складно управляти під великим навантаженням; Може виникати проблеми з резервним копіюванням та відновленням.
PostgreSQL	Потужність та гнучкість; Великий набір функцій та розширень; Висока надійність та стабільність.	Вимагає більше ресурсів; Складніше налаштування та адміністрування.
MongoDB	Гнучкість схеми даних; Швидкість розробки; Підтримка розподіленої бази даних та горизонтального масштабування.	Менш підходить для додатків з потребою вищої консистентності даних; Менша підтримка транзакцій порівняно з реляційними базами даних.

Таблиця Г.3

## Переваги та недоліки чат-боту міста Чернігова

Меню	Переваги	Недоліки
Головне меню	Гарний інтерфейс; Зручна клавіатура; Повний опис усіх функцій.	Погана ієрархія пунктів меню.
Послуги	Широкий спектр функцій; Розподіл функцій по пунктах;	Відсутність форматування тексту
Контакти	Приємний дизайн; Присутні найголовніші посилання.	Не помічено

Продовження табл. Г.3

Меню	Переваги	Недоліки
Питання та відповіді	Зручна функція;	Надано замало інформації; Відсутність форматування тексту
Графік прийому громадян	Наявна уся необхідна інформація.	Поганий дизайн.
«ЄВідновлення»	Гарний дизайн; Необхідні посилання;	Погане розташування у меню.

Таблиця Г.4

## Переваги та недоліки чат-боту міста Харків

Меню	Переваги	Недоліки
Головне меню	Простий та зрозумілий інтерфейс.	Не дуже гарний дизайн; Обмежений функціонал.
Відповіді на питання	Зручна функція; Містить у собі великий об'єм інформації.	Можливість покращення дизайну, для більш привабливого та інтуїтивного інтерфейсу.
Реєстрація в черзі	Зручна ієрархія; Інтуїтивно зрозумілий інтерфейс.	Не помічено.

Таблиця Г.5

## Переваги та недоліки чат-боту міста Київ

Меню	Переваги	Недоліки
Головне меню	Простий доступ до основних опцій.	Відсутність inline клавіатури.
Подивитися адреси	Гарно оформлена, зручна функція.	Деякі меню не працюють.
Реєстрація в електронній черзі	У цілому корисна функція.	Функція відсутня у чат-боті, потрібно переходити на сайт.
Інформація про стан розгляду справи	У цілому корисна функція.	Функція відсутня у чат-боті, потрібно переходити на сайт.

Продовження табл. Г.5

Меню	Переваги	Недоліки
Послуги які можна отримати в особистому кабінеті	Непогано структурований перелік послуг.	Ніякої функціональності, суто інформаційне меню.
Залишити відгук	Добре структурована функція, з непоганою ієрархією.	Немає значних недоліків.
Call центр ЦНАП	Просте надання контактних номерів телефону.	Немає значних недоліків, проте може бути більш функціональним.
Графік роботи	Зручний доступ до графіка роботи.	Немає значних недоліків.

Таблиця Г.6

## Переваги та недоліки чат-боту міста Черкаси

Меню	Переваги	Недоліки
Головне меню	Простий доступ до основних функцій; Зручне використання.	Поганий дизайн; Обмежений функціонал.
Реєстрація в черзі	Можливість зареєструватись на послугу безпосередньо з чату.	Відсутність клавіатури, що ускладнює навігацію.
Відстежити послугу	Зручна функція.	Можливість більш зручного відстеження статусу; Неповний, недосконалий інтерфейс.
Оцінити якість обслуговування	Зручний спосіб подачі зворотного зв'язку з адміністрацією.	Можливість покращення інтерфейсу для більшої зручності;

## Лістинг коду

**«Модель комп'ютерної системи надання адміністративних послуг із використанням чат-боту»**

**Main\_menu.py**

```

import telebot
import Contact_info
import Order
import Petition
import Questions
import Queue
import Statistics
import Structure
import Text
import Main_info
import Docs_info
from telebot import types
import Vacations

# Тут має бути ключ телеграм боту

# Змінні для головного меню
schedule_btn = types.KeyboardButton('📅 Загальна інформація')
docs_btn = types.KeyboardButton('📁 Інформація по послугам')
queue_btn = types.KeyboardButton('📧 Електронна черга')
order_docs_btn = types.KeyboardButton('📁 Замовити документ')
status_btn = types.KeyboardButton('🔍 Статус заявки')
get_contacts_btn = types.KeyboardButton('📞 Контактна
інформація')
structure_btn = types.KeyboardButton('👤📁👤 Структура ЦНАП')
answers_btn = types.KeyboardButton('❓ Ф.А.О.')
statistics_btn = types.KeyboardButton('📊 Статистика ЦНАП')
vacations_btn = types.KeyboardButton('📅 Вільні вакансії')
petition_btn = types.KeyboardButton('📄 Переглянути петиції')
back_btn = types.KeyboardButton('⬅️ Повернутися до меню')

@bot.message_handler(commands=['start'])
def start(message):
    main_menu(message, Text.welcome)

@bot.message_handler(commands=['main'])
def main(message):
    main_menu(message, Text.helper)

```

## Продовження лістингу

```

# Функція повернення на головне меню
def buttons(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(back_btn)
    bot.send_message(message.chat.id, text, parse_mode='html',
reply_markup=markup)
    bot.register_next_step_handler(message, main)

# КНОПКИ ГОЛОВНОГО МЕНЮ
def main_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(schedule_btn)
    markup.add(docs_btn)
    markup.add(queue_btn)
    markup.add(order_docs_btn)
    markup.add(status_btn)
    markup.add(get_contacts_btn)
    markup.add(structure_btn)
    markup.add(answers_btn)
    markup.add(statistics_btn)
    markup.add(vacations_btn)
    markup.add(petition_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message, on_click_main)

# Обробка натискання на кнопки головного меню
def on_click_main(message):
    if message.text == '📄 Загальна інформація':
        bot.send_message(message.chat.id, Main_info.heading,
parse_mode='html')
        buttons(message, Main_info.schedule)
        Main_info.location_func(message)
    if message.text == '📄 Інформація по послугам':
        docs_info_menu(message, Docs_info.docs)
    if message.text == '📄 Електронна черга':
        queue_menu(message, Queue.queue_info)
    if message.text == '📄 ЗАМОВИТИ ДОКУМЕНТ':
        order_menu(message, Order.order_info)
    if message.text == '🔍 Статус заявки':
        status_order_menu(message, Order.order_status_info)
    if message.text == '📞 Контактна інформація':
        buttons(message, Contact_info.contact_info)
    if message.text == '👤👤👤 Структура ЦНАП':
        buttons(message, Structure.structure)

```

## Продовження лістингу

```

if message.text == '❓ F.A.Q.':
    questions_menu(message, Questions.questions_info)
if message.text == '📊 Статистика ЦНАП':
    year_statistics_menu(message,
Statistics.statistics_info)
if message.text == '📅 Вільні вакансії':
    vacations_menu(message, Vacations.vacations_info)
if message.text == '📄 Переглянути петиції':
    petitions_menu(message, Petition.petitions_info)

# Кнопки меню "Інформація по документам"
def docs_info_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Docs_info.doc1_btn)
    markup.add(Docs_info.doc2_btn)
    markup.add(Docs_info.doc3_btn)
    markup.add(Docs_info.doc4_btn)
    markup.add(Docs_info.doc5_btn)
    markup.add(Docs_info.doc6_btn)
    markup.add(Docs_info.doc7_btn)
    markup.add(Docs_info.doc8_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_docs_info)

# Обробка натискання на кнопки меню "Інформація по документам"
def on_click_docs_info(message):
    if message.text == '👤 Громадянство, міграція, переїзд':
        docs_theme1_info_menu(message,
Docs_info.docs_theme1_info)
    if message.text == '👨👩👧👦 Сім'я':
        docs_theme2_info_menu(message,
Docs_info.docs_theme2_info)
    if message.text == '🏠 Нерухоме майно та будівництво':
        docs_theme3_info_menu(message,
Docs_info.docs_theme3_info)
    if message.text == '🚗 Транспорт':
        docs_theme4_info_menu(message,
Docs_info.docs_theme4_info)
    if message.text == '🏡 Сільське господарство':
        docs_theme5_info_menu(message,
Docs_info.docs_theme5_info)
    if message.text == '☁️ Природні ресурси та екологія':
        docs_theme6_info_menu(message,
Docs_info.docs_theme6_info)

```

## Продовження лістингу

```

    if message.text == '🇸 Реєстрація бізнесу та громадських
формувань':
        docs_theme7_info_menu(message,
Docs_info.docs_theme7_info)
    if message.text == '⚽ Спорт':
        docs_theme8_info_menu(message,
Docs_info.docs_theme8_info)

# Кнопки меню "Інформація по документам" на тему
"Громадянство, міграція, переїзд"
def docs_theme1_info_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Docs_info.theme1_doc1_btn)
    markup.add(Docs_info.theme1_doc2_btn)
    markup.add(Docs_info.theme1_doc3_btn)
    markup.add(Docs_info.theme1_doc4_btn)
    markup.add(Docs_info.theme1_doc5_btn)
    markup.add(Docs_info.theme1_doc6_btn)
    markup.add(Docs_info.theme1_doc7_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_theme1_docs_info)

# Обробка натискання на кнопки меню "Інформація по документам"
теми "Громадянство, міграція, переїзд"
def on_click_theme1_docs_info(message):
    if message.text == 'Реєстрація місця
проживання/перебування':
        buttons(message, Docs_info.theme1_doc1_info)
    if message.text == 'Зняття із
задекларованого/зареєстрованого місця проживання':
        buttons(message, Docs_info.theme1_doc2_info)
    if message.text == 'Вклеювання до паспорта громадянина
України фотокартки при досягненні 25 і 45-річного віку':
        buttons(message, Docs_info.theme1_doc3_info)
    if message.text == 'Оформлення і видача закордонного
паспорта громадянина України':
        buttons(message, Docs_info.theme1_doc4_info)
    if message.text == 'Оформлення і видача закордонного
паспорта громадянина України у разі втрати або викрадення':
        buttons(message, Docs_info.theme1_doc5_info)
    if message.text == 'Оформлення і видача паспорта
громадянина України вперше після досягнення 14-річного віку':
        buttons(message, Docs_info.theme1_doc6_info)
    if message.text == 'Оформлення і видача паспорта
громадянина України у зв'язку з втратою/викраденням':
        buttons(message, Docs_info.theme1_doc7_info)

```

## Продовження лістингу

```

# Кнопки меню "Інформація по документам" на тему "Сім'я"
def docs_theme2_info_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Docs_info.theme2_doc1_btn)
    markup.add(Docs_info.theme2_doc2_btn)
    markup.add(Docs_info.theme2_doc3_btn)
    markup.add(Docs_info.theme2_doc4_btn)
    markup.add(Docs_info.theme2_doc5_btn)
    markup.add(Docs_info.theme2_doc6_btn)
    markup.add(Docs_info.theme2_doc7_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_theme2_docs_info)

# Обробка натискання на кнопки меню "Інформація по документам"
теми "Сім'я"
def on_click_theme2_docs_info(message):
    if message.text == 'Державна реєстрація шлюбу':
        buttons(message, Docs_info.theme2_doc1_info)
    if message.text == 'Державна реєстрація розірвання шлюбу':
        buttons(message, Docs_info.theme2_doc2_info)
    if message.text == 'Державна реєстрація зміни імені':
        buttons(message, Docs_info.theme2_doc3_info)
    if message.text == 'Державна реєстрація смерті':
        buttons(message, Docs_info.theme2_doc4_info)
    if message.text == 'Державна реєстрація народження':
        buttons(message, Docs_info.theme2_doc5_info)
    if message.text == 'Видача посвідчення батьків та дитини з
багатодітної сім'ї':
        buttons(message, Docs_info.theme2_doc6_info)
    if message.text == 'Продовження строку дії посвідчень
батьків та дитини з багатодітної сім'ї':
        buttons(message, Docs_info.theme2_doc7_info)

# Кнопки меню "Інформація по документам" на тему "Нерухоме
майно та будівництво"
def docs_theme3_info_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Docs_info.theme3_doc1_btn)
    markup.add(Docs_info.theme3_doc2_btn)
    markup.add(Docs_info.theme3_doc3_btn)
    markup.add(Docs_info.theme3_doc4_btn)
    markup.add(Docs_info.theme3_doc5_btn)
    markup.add(Docs_info.theme3_doc6_btn)

```

## Продовження лістингу

```

        bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
        bot.register_next_step_handler(message,
on_click_theme3_docs_info)

    # Обробка натискання на кнопки меню "Інформація по документам"
теми "Нерухоме майно та будівництво"
    def on_click_theme3_docs_info(message):
        if message.text == 'Надання інформації з Державного
реєстру речових прав на нерухоме майно':
            buttons(message, Docs_info.theme3_doc1_info)
        if message.text == 'Надання містобудівних умов та обмежень
забудови земельної ділянки':
            buttons(message, Docs_info.theme3_doc2_info)
        if message.text == 'Видача будівельного паспорта забудови
земельної ділянки':
            buttons(message, Docs_info.theme3_doc3_info)
        if message.text == 'Видача свідоцтва про право власності
на житло':
            buttons(message, Docs_info.theme3_doc4_info)
        if message.text == 'Видача дубліката свідоцтва про право
власності':
            buttons(message, Docs_info.theme3_doc5_info)
        if message.text == 'Взяття на облік громадян, які
потребують поліпшення житлових умов':
            buttons(message, Docs_info.theme3_doc6_info)

    # Кнопки меню "Інформація по документам" на тему "Транспорт"
    def docs_theme4_info_menu(message, text):
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
        markup.add(Docs_info.theme4_doc1_btn)
        markup.add(Docs_info.theme4_doc2_btn)
        markup.add(Docs_info.theme4_doc3_btn)
        markup.add(Docs_info.theme4_doc4_btn)
        bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
        bot.register_next_step_handler(message,
on_click_theme4_docs_info)

    # Обробка натискання на кнопки меню "Інформація по документам"
теми "Транспорт"
    def on_click_theme4_docs_info(message):
        if message.text == 'Реєстрація великотоннажних та інших
технологічних транспортних засобів':
            buttons(message, Docs_info.theme4_doc1_info)
        if message.text == 'Зняття з обліку великотоннажних та
інших технологічних транспортних засобів':

```

## Продовження лістингу

```

        buttons(message, Docs_info.theme4_doc2_info)
        if message.text == 'Перереєстрація великотоннажних та
інших технологічних транспортних засобів':
            buttons(message, Docs_info.theme4_doc3_info)
            if message.text == ('Обмін або видача в зв'язку з втратою
(викрадення) посвідчення водія на право керування '
                'транспортними засобами'):
                    buttons(message, Docs_info.theme4_doc4_info)

# Кнопки меню "Інформація по документам" на тему "Сільське
господарство"
def docs_theme5_info_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Docs_info.theme5_doc1_btn)
    markup.add(Docs_info.theme5_doc2_btn)
    markup.add(Docs_info.theme5_doc3_btn)
    markup.add(Docs_info.theme5_doc4_btn)
    markup.add(Docs_info.theme5_doc5_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_theme5_docs_info)

# Обробка натискання на кнопки меню "Інформація по документам"
теми "Сільське господарство"
def on_click_theme5_docs_info(message):
    if message.text == 'Видача посвідчення на право роботи з
пестицидами':
        buttons(message, Docs_info.theme5_doc1_info)
        if message.text == ('Видача дозволу для операторів ринку,
що виробляють та/або зберігають харчові продукти '
            'тваринного походження'):
                buttons(message, Docs_info.theme5_doc2_info)
                if message.text == ('Анулювання дозволу операторам ринку,
що виробляють та/або зберігають харчові продукти '
                    'тваринного походження'):
                            buttons(message, Docs_info.theme5_doc3_info)
                            if message.text == 'Видача дозволу з переробки неїстівних
продуктів тваринного походження':
                                    buttons(message, Docs_info.theme5_doc4_info)
                                    if message.text == 'Реєстрація пасіки':
                                            buttons(message, Docs_info.theme5_doc5_info)

# Кнопки меню "Інформація по документам" на тему "Сільське
господарство"
def docs_theme6_info_menu(message, text):

```

## Продовження лістингу

```

markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
markup.add(Docs_info.theme6_doc1_btn)
markup.add(Docs_info.theme6_doc2_btn)
markup.add(Docs_info.theme6_doc3_btn)
markup.add(Docs_info.theme6_doc4_btn)
markup.add(Docs_info.theme6_doc5_btn)
bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
bot.register_next_step_handler(message,
on_click_theme6_docs_info)

# Обробка натискання на кнопки меню "Інформація по документам"
теми "Сільське господарство"
def on_click_theme6_docs_info(message):
    if message.text == 'Рішення щодо продажу земельних ділянок
юридичним та фізичним особам':
        buttons(message, Docs_info.theme6_doc1_info)
    if message.text == 'Видача довідки про наявність та розмір
земельної частки (паю)':
        buttons(message, Docs_info.theme6_doc2_info)
    if message.text == 'Рішення щодо передачі у власність та
надання в оренду громадянам земельних ділянок':
        buttons(message, Docs_info.theme6_doc3_info)
    if message.text == 'Видача витягу із технічної
документації з нормативної грошової оцінки земельних ділянок':
        buttons(message, Docs_info.theme6_doc4_info)
    if message.text == 'Рішення щодо надання земельної ділянки
у користування або про передачу у власність':
        buttons(message, Docs_info.theme6_doc5_info)

# Кнопки меню "Інформація по документам" на тему "Сільське
господарство"
def docs_theme7_info_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Docs_info.theme7_doc1_btn)
    markup.add(Docs_info.theme7_doc2_btn)
    markup.add(Docs_info.theme7_doc3_btn)
    markup.add(Docs_info.theme7_doc4_btn)
    markup.add(Docs_info.theme7_doc5_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_theme7_docs_info)

# Обробка натискання на кнопки меню "Інформація по документам"
теми "Сільське господарство"

```

## Продовження лістингу

```

def on_click_theme7_docs_info(message):
    if message.text == 'Державна реєстрація фізичної особи
    підприємцем':
        buttons(message, Docs_info.theme7_doc1_info)
    if message.text == 'Державна реєстрація припинення
    підприємницької діяльності фізичної особи':
        buttons(message, Docs_info.theme7_doc2_info)
    if message.text == 'Державна реєстрація змін до відомостей
    про фізичну особу - підприємця':
        buttons(message, Docs_info.theme7_doc3_info)
    if message.text == ('Видача витягу з Єдиного державного
    реєстру юридичних осіб, фізичних осіб - підприємців та '
        'громадських формувань'):
        buttons(message, Docs_info.theme7_doc4_info)
    if message.text == ('Виправлення помилки, допущеної у
    відомостях Єдиного державного реєстру юридичних осіб, '
        'фізичних осіб'):
        buttons(message, Docs_info.theme7_doc5_info)

    # Кнопки меню "Інформація по документам" на тему "Сільське
    господарство"
    def docs_theme8_info_menu(message, text):
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
        one_time_keyboard=True)
        markup.add(Docs_info.theme8_doc1_btn)
        markup.add(Docs_info.theme8_doc2_btn)
        bot.send_message(message.chat.id, text,
        reply_markup=markup, parse_mode='html')
        bot.register_next_step_handler(message,
        on_click_theme8_docs_info)

    # Обробка натискання на кнопки меню "Інформація по документам"
    теми "Сільське господарство"
    def on_click_theme8_docs_info(message):
        if message.text == 'Присвоєння спортсменам II та III
        спортивних розрядів':
            buttons(message, Docs_info.theme8_doc1_info)
            if message.text == ('Присвоєння спортивних розрядів
            спортсменам: "Кандидат у майстри спорту України" та I '
                'спортивний розряд'):
                buttons(message, Docs_info.theme8_doc2_info)

    # Кнопки меню послуг "Електронна черга"
    def queue_menu(message, text):
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
        one_time_keyboard=True)
        markup.add(Queue.doc1_btn)
        markup.add(Queue.doc2_btn)

```

## Продовження лістингу

```

markup.add(Queue.doc3_btn)
markup.add(Queue.doc4_btn)
markup.add(Queue.doc5_btn)
markup.add(Queue.doc6_btn)
markup.add(Queue.doc7_btn)
markup.add(Queue.doc8_btn)
bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
bot.register_next_step_handler(message,
on_click_queue_menu)

# Обробка натискання на кнопки меню послуги "Електронна черга"
def on_click_queue_menu(message):
    if message.text == '👤 Громадянство, міграція, переїзд':
        Queue.service = "Громадянство, міграція, переїзд"
        queue_day_btn(message, Queue.queue_text_day)
    if message.text == '👨👩👦 Сім'я':
        Queue.service = "Сім'я"
        queue_day_btn(message, Queue.queue_text_day)
    if message.text == '🏠 Нерухоме майно та будівництво':
        Queue.service = "Нерухоме майно та будівництво"
        queue_day_btn(message, Queue.queue_text_day)
    if message.text == '🚗 Транспорт':
        Queue.service = "Транспорт"
        queue_day_btn(message, Queue.queue_text_day)
    if message.text == '🏡 Сільське господарство':
        Queue.service = "Сільське господарство"
        queue_day_btn(message, Queue.queue_text_day)
    if message.text == '☁️ Природні ресурси та екологія':
        Queue.service = "Природні ресурси та екологія"
        queue_day_btn(message, Queue.queue_text_day)
    if message.text == '💰 Реєстрація бізнесу та громадських
формувань':
        Queue.service = "Реєстрація бізнесу та громадських
формувань"
        queue_day_btn(message, Queue.queue_text_day)
    if message.text == '⚽ Спорт':
        Queue.service = "Спорт"
        queue_day_btn(message, Queue.queue_text_day)

# Кнопки меню обрання дати послуги "Електронна черга"
def queue_day_btn(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Queue.queue1_day_btn)
    markup.add(Queue.queue2_day_btn)
    markup.add(Queue.queue3_day_btn)

```

## Продовження лістингу

```

markup.add(Queue.queue4_day_btn)
markup.add(Queue.queue5_day_btn)
bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
bot.register_next_step_handler(message,
on_click_queue_day_btn)

# Обробка натискання на кнопки меню обрання дати послуги
"Електронна черга"
def on_click_queue_day_btn(message):
    if message.text == 'Понеділок - 22.01.2024':
        Queue.day = 'Понеділок - 22.01.2024'
        queue_time_btn(message, Queue.queue_text_time)
    if message.text == 'Вівторок - 23.01.2024':
        Queue.day = 'Вівторок - 23.01.2024'
        queue_time_btn(message, Queue.queue_text_time)
    if message.text == 'Середа - 24.01.2024':
        Queue.day = 'Середа - 24.01.2024'
        queue_time_btn(message, Queue.queue_text_time)
    if message.text == 'Четвер - 25.01.2024':
        Queue.day = 'Четвер - 25.01.2024'
        queue_time_btn(message, Queue.queue_text_time)
    if message.text == 'П\'ятниця - 26.01.2024':
        Queue.day = 'П\'ятниця - 26.01.2024'
        queue_time_btn(message, Queue.queue_text_time)

# Кнопки меню обрання часу послуги "Електронна черга"
def queue_time_btn(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Queue.queue1_time_btn)
    markup.add(Queue.queue4_time_btn)
    markup.add(Queue.queue5_time_btn)
    markup.add(Queue.queue7_time_btn)
    markup.add(Queue.queue8_time_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_queue_time_btn)

# Обробка натискання на кнопки меню обрання часу послуги
"Електронна черга"
def on_click_queue_time_btn(message):
    if message.text == '08:00':
        Queue.time = '08:00'
        buttons(message, Queue.answer(Queue.service,
Queue.day, Queue.time))
    if message.text == '11:00':

```

## Продовження лістингу

```

        Queue.time = '11:00'
        buttons(message, Queue.answer(Queue.service,
Queue.day, Queue.time))
        if message.text == '12:00':
            Queue.time = '12:00'
            buttons(message, Queue.answer(Queue.service,
Queue.day, Queue.time))
        if message.text == '14:00':
            Queue.time = '14:00'
            buttons(message, Queue.answer(Queue.service,
Queue.day, Queue.time))
        if message.text == '15:00':
            Queue.time = '15:00'
            buttons(message, Queue.answer(Queue.service,
Queue.day, Queue.time))

# Кнопки меню послуг "Замовити документ"
def order_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Order.order1_btn)
    markup.add(Order.order2_btn)
    markup.add(Order.order3_btn)
    markup.add(Order.order4_btn)
    markup.add(Order.order5_btn)
    markup.add(Order.order_back_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_order_menu)

# Обробка кнопок меню послуг "Замовити документ"
@bot.message_handler(content_types=['photo', 'file'])
def on_click_order_menu(message):
    if message.text == 'Довідка про склад сім'ї, або
zareєстрованих у житловому приміщенні/будинку осіб':
        bot.send_message(message.chat.id, Order.order1_info,
parse_mode='html')
        bot.register_next_step_handler(message,
on_click_order_menu_end)
    if message.text == 'Довідка про реєстрацію місця
проживання (Додаток 13)':
        bot.send_message(message.chat.id, Order.order2_info,
parse_mode='html')
        bot.register_next_step_handler(message,
on_click_order_menu_end)
    if message.text == 'Видача довідки про наявність та розмір
земельної частки (паю)':

```

## Продовження лістингу

```

        bot.send_message(message.chat.id, Order.order3_info,
        parse_mode='html')
        bot.register_next_step_handler(message,
        on_click_order_menu_end)
        if message.text == 'Присвоєння спортсменам II та III
        спортивних розрядів':
            bot.send_message(message.chat.id, Order.order4_info,
            parse_mode='html')
            bot.register_next_step_handler(message,
            on_click_order_menu_end)
            if message.text == 'Видача посвідчення на право роботи з
            пестицидами':
                bot.send_message(message.chat.id, Order.order5_info,
                parse_mode='html')
                bot.register_next_step_handler(message,
                on_click_order_menu_end)
                if message.text == '✘ Скасувати дію':
                    buttons(message, 'Дія скасована.')

    # Вивід повідомлення про успішне замовлення
    def on_click_order_menu_end(message):
        buttons(message, Order.thanks)

    # Кнопки меню послуг "Статус заявки"
    def status_order_menu(message, text):
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
        one_time_keyboard=True)
        markup.add(Order.order1_btn)
        markup.add(Order.order2_btn)
        markup.add(Order.order3_btn)
        markup.add(Order.order4_btn)
        markup.add(Order.order5_btn)
        bot.send_message(message.chat.id, text,
        reply_markup=markup, parse_mode='html')
        bot.register_next_step_handler(message,
        on_click_status_order_menu)

    # Обробка кнопок меню послуг "Статус заявки"
    def on_click_status_order_menu(message):
        if message.text == 'Довідка про склад сім'ї, або
        зареєстрованих у житловому приміщенні/будинку осіб':
            buttons(message, Order.order_answer_yes)
        if message.text == 'Довідка про реєстрацію місця
        проживання (Додаток 13)':
            buttons(message, Order.order_answer_no)
        if message.text == 'Видача довідки про наявність та розмір
        земельної частки (паю)':
            buttons(message, Order.order_answer_yes)

```

## Продовження лістингу

```

    if message.text == 'Присвоєння спортсменам II та III
спортивних розрядів':
        buttons(message, Order.order_answer_no)
    if message.text == 'Видача посвідчення на право роботи з
пестицидами':
        buttons(message, Order.order_answer_yes)

# Кнопки меню послуг "F.A.Q."
def questions_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Questions.questions1_btn)
    markup.add(Questions.questions2_btn)
    markup.add(Questions.questions3_btn)
    markup.add(Questions.questions4_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_questions_menu)

# Обробка кнопок меню послуг "F.A.Q."
def on_click_questions_menu(message):
    if message.text == 'Електронна черга':
        questions_queue_menu(message,
Questions.questions1_info)
    if message.text == 'Стан замовлення документу':
        questions_status_menu(message,
Questions.questions2_info)
    if message.text == 'Шеринг документів':
        questions_sharing_menu(message,
Questions.questions3_info)
    if message.text == '«Малятко»':
        questions_baby_menu(message,
Questions.questions4_info)

# Кнопки меню відповідей на тему "Електронна черга"
def questions_queue_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Questions.questions1_btn1)
    markup.add(Questions.questions1_btn2)
    markup.add(Questions.questions1_btn3)
    markup.add(Questions.questions1_btn4)
    markup.add(Questions.questions1_btn5)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_questions_queue_menu)

```

## Продовження лістингу

```

# Обробка натискання на кнопки меню відповідей на тему
"Електронна черга"
def on_click_questions_queue_menu(message):
    if message.text == 'Неможливо записатися в онлайн-черзі':
        buttons(message, Questions.questions1_btn1_info)
    if message.text == 'Як здійснити запис до електронної
черги?':
        buttons(message, Questions.questions1_btn2_info)
    if message.text == 'Які категорії громадян мають право на
позачергове обслуговування в ЦНАП м. Тернівка?':
        buttons(message, Questions.questions1_btn3_info)
    if message.text == 'Я записався на 5-10 хвилин - чому
мені відмовляють?':
        buttons(message, Questions.questions1_btn4_info)
    if message.text == 'Я не зберіг(загубив) чек електронної
черги':
        buttons(message, Questions.questions1_btn5_info)

# Кнопки меню відповідей на тему "Стан замовлення документу"
def questions_status_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Questions.questions2_btn1)
    markup.add(Questions.questions2_btn2)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_questions_status_menu)

# Обробка натискання на кнопки меню відповідей на тему "Стан
замовлення документу"
def on_click_questions_status_menu(message):
    if message.text == 'Де подивитись інформацію про стан
замовлення документу?':
        buttons(message, Questions.questions2_btn1_info)
    if message.text == 'Який час очікування на замовлення
документу?':
        buttons(message, Questions.questions2_btn2_info)

# Кнопки меню відповідей на тему "Шеринг документів"
def questions_sharing_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Questions.questions3_btn1)
    markup.add(Questions.questions3_btn2)
    markup.add(Questions.questions3_btn3)

```

## Продовження лістингу

```

        markup.add(Questions.questions3_btn4)
        bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
        bot.register_next_step_handler(message,
on_click_questions_sharing_menu)

# Обробка натискання на кнопки меню відповідей на тему "Шеринг
документів"
def on_click_questions_sharing_menu(message):
    if message.text == 'Що таке «Шеринг документів»?':
        buttons(message, Questions.questions3_btn1_info)
    if message.text == 'Як використовувати послугу?':
        buttons(message, Questions.questions3_btn2_info)
    if message.text == 'Які документи можна шерити?':
        buttons(message, Questions.questions3_btn3_info)
    if message.text == 'Як працює шеринг документів?':
        buttons(message, Questions.questions3_btn4_info)

# Кнопки меню відповідей на тему "єМалятко"
def questions_baby_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Questions.questions4_btn1)
    markup.add(Questions.questions4_btn2)
    markup.add(Questions.questions4_btn3)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_questions_baby_menu)

# Обробка натискання на кнопки меню відповідей на тему
"єМалятко"
def on_click_questions_baby_menu(message):
    if message.text == 'Що таке «єМалятко»?':
        buttons(message, Questions.questions4_btn1_info)
    if message.text == 'Де можна скористатися послугою
«єМалятко» в м. Тернівка?':
        buttons(message, Questions.questions4_btn2_info)
    if message.text == 'Де можна дізнатися докладну інформацію
про послугу «єМалятко»?':
        buttons(message, Questions.questions4_btn3_info)

# Кнопки меню відповідей на тему "єМалятко"
def year_statistics_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Statistics.year2023_btn)

```

## Продовження лістингу

```

        markup.add(Statistics.year2024_btn)
        bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
        bot.register_next_step_handler(message,
on_click_year_statistics_menu)

# Обробка натискання на кнопки меню відповідей на тему
"єМалятко"
def on_click_year_statistics_menu(message):
    if message.text == '2023 рік':
        buttons(message, Statistics.year2023_info)
    if message.text == '2024 рік':
        buttons(message, Statistics.year2024_info)

# Кнопки меню послуг "Вільні вакансії"
def vacations_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Vacations.vacation1_btn)
    markup.add(Vacations.vacation2_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_vacations_menu)

# Обробка натискання на кнопки меню "Вільні вакансії"
def on_click_vacations_menu(message):
    if message.text == 'Начальник відділу інтерактивних,
інформаційних, адміністративних послуг':
        buttons(message, Vacations.vacation1_info)
    if message.text == 'Адміністратор ЦНАП':
        buttons(message, Vacations.vacation2_info)

# Кнопки меню послуг "Переглянути петиції"
def petitions_menu(message, text):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True)
    markup.add(Petition.petition1_btn)
    markup.add(Petition.petition2_btn)
    markup.add(Petition.petition3_btn)
    markup.add(Petition.petition4_btn)
    markup.add(Petition.petition5_btn)
    markup.add(Petition.petition6_btn)
    bot.send_message(message.chat.id, text,
reply_markup=markup, parse_mode='html')
    bot.register_next_step_handler(message,
on_click_petitions_menu)

```

## Продовження лістингу

```
# Обробка натискання на кнопки меню "Переглянути петиції"
def on_click_petitions_menu(message):
    if message.text == 'Транспортний рух міста':
        buttons(message, Petition.petition1_info)
        Petition.petition_func(message,
Petition.petition_accept)
    if message.text == 'Ремонт дороги за адресою - вул. Миру
15':
        buttons(message, Petition.petition2_info)
        Petition.petition_func(message,
Petition.petition_accept)
    if message.text == 'Створення дитячого майданчика':
        buttons(message, Petition.petition3_info)
        Petition.petition_func(message,
Petition.petition_accept)
    if message.text == 'Обладнати пішохідний перехід':
        buttons(message, Petition.petition4_info)
        Petition.petition_func(message,
Petition.petition_accept)
    if message.text == 'Вуличне освітлення':
        buttons(message, Petition.petition5_info)
        bot.send_message(message.chat.id,
Petition.petition_over)
    if message.text == 'Бювети питної води для населення':
        buttons(message, Petition.petition6_info)
        bot.send_message(message.chat.id,
Petition.petition_over)

bot.polling(none_stop=True)
```