МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА
на тему:
Розробка чат-боту для передачі показників лічильника за світло

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи КБ-21-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»

_____ Дубини Максима Максимовича
(підпис) (прізвище, ім'я, по батькові)

Науковий керівник _____ Філоненко Михайло Миколайович, к.ф.-м.н., доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної програми _____ Омельчук Антон Анатолійович, к.т.н.
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних технологій і систем протокол від «13» травня 2025 № 18

Завідувач кафедри _____ Філоненко Михайло Миколайович, к.ф.-м.н., доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2025

Реєстрація № 59 — 26.05.2025 (дата) — (підпис відповідальної особи)

Результати перевірки — до захисту (до захисту, на доопрацювання) — 26.05.2025 (дата) — (підпис)

Результат захисту — 4 (добре) (оцінка) — 05.06.2025 (дата) — (підпис)

Голова комісії: (підпис) В.М. Франчук

Члени комісії:

(підпис) М.М. Філоненко

(підпис) С.М. Грищенко

(підпис) А.А. Омельчук

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	**від 0 до 30**	25		Філоненко М.М. Франчук В.М.
2. Дотримання вимог щодо оформлення	**від 0 до 10**	7		Філоненко М.М. Грищенко С.М.
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	**від 0 до 5**	3		Омельчук А.А.
4. Рецензент	**від 0 до 5**	4		Суховецька Л.П.
РАЗОМ БАЛІВ	**від 0 до 50**	39		

Оцінювання кваліфікаційної роботи науковим керівником – 36 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалоюЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
75	C	4 (добре)	05.06.2025	Франчук В.М. Філоненко М.М. Грищенко С.М. Омельчук А.А. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань 12 «Інформаційні технології»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Інформаційні управляючі системи і технології (за галузями)»

Затверджую
Завідувач кафедри_____ Філоненко М.М.
«_14_»_листопада_2024

Завдання на кваліфікаційну роботу

здобувачу вищої освіти

Максиму Максимовичу Дубині
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Розробка чат-боту для передачі показників лічильника за світло
Затверджена наказом ректора від «13» листопада 2024 № 1717
2. Строк здачі закінченої роботи «08» травня 2025
3. Мета роботи Розробка чат-боту для автоматизації передачі показників лічильника за електроенергію, що забезпечить зручність для користувачів та підвищить ефективність обробки даних для постачальників енергії.
Об'єкт дослідження. Чат-боти для автоматизації процесів збору даних.
Предмет дослідження. Процес створення чат-боту для передачі показників лічильника за електроенергію.
База дослідження Чат-бот для передачі показників лічильника, інтегрований із системами обліку електроенергії.
4. Вихідні дані до роботи Інтерфейс чат-боту для введення показників лічильників. Схема роботи чат-боту для передачі даних. Архітектура системи чат-боту. Схема інтеграції чат-боту з базою даних енергетичної компанії.
5. Перелік графічного матеріалу головна сторінка веб-сайту готелю, сторінка опису номера, сторінка підтвердження замовлення, базова архітектура ве-сайту, архітектура веб-сайту готеля, схема бази даних веб-сайту готеля
6. Зміст кваліфікаційної роботи
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ
1.1 Сучасний стан систем передачі показників лічильників за електроенергію в Україні
1.2. Аналіз інструментів і технологій для розробки чат-ботів
РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ВЕБ-САЙТУ
2.1. Принципи дизайну та етапи створення чат-боту
2.2. Архітектура чат-боту та інтеграція з системами обліку електроенергії
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТУ

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	За фактом
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 30.10.2024	До 30.10.2024
2	І розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	27.12.2024	27.12.2024
3	ІІ розділ (аналіз експериментальної частини про методики досліджень)	17.02.2025	17.02.2025
4	ІІІ розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	16.04.2025	16.04.2025
5	Висновки	01.05.2025	01.05.2025
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	09.05.2025	09.05.2025
7	Отримання відгуку керівника	27.05.2025	27.05.2025
8	Отримання зовнішньої рецензії	29.05.2025	29.05.2025

8. Дата видачі завдання «14» листопада 2024

9. Керівник кваліфікаційної роботи _____ Філоненко М.М.
(підпис) (прізвище, ініціали)

10. Керівник освітньо-професійної програми _____ Омельчук А.А.
(підпис) (прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти _____ Дубина М.М.
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Розробка чат-боту для передачі показників лічильника за світло

Мета роботи полягає розробка чат-боту для автоматизації передачі показників лічильника за електроенергію, що забезпечить зручність для користувачів та підвищить ефективність обробки даних для постачальників енергії. Зі сторони користувачів це досягається шляхом створення інтуїтивно зрозумілого інтерфейсу для швидкої передачі даних, а зі сторони постачальників – завдяки автоматизації збору та обробки інформації.

Об'єкт кваліфікаційної роботи – розробка чат-ботів для автоматизації процесів.

Предмет дослідження – створення чат-боту для передачі показників лічильника за електроенергію.

Одержані результати можуть бути використані при оптимізації бізнес процесів готелю та залученню нових клієнтів завдяки використанню веб-сайту

Кваліфікаційна робота містить 50 сторінок, 6 рисунків, список використаних джерел 41 найменувань.

Ключові слова: чат-бот, лічильник, зручність, інформація.

Апробація результатів роботи. Основні результати дослідження представлено на конференції «Інноваційні технології в енергетиці та автоматизації» (м. Київ, 15 березня 2025 р.).

Рік виконання кваліфікаційної роботи 2025

Рік захисту роботи 2025

ANNOTATION

Development of a Chatbot for Submitting Electricity Meter Readings

The aim of this work is to develop a chatbot for automating the submission of electricity meter readings, which will provide convenience for users and improve data processing efficiency for energy suppliers. For users, this is achieved through the creation of an intuitive interface for quick data submission, and for suppliers – through the automation of information collection and processing.

Object of the qualification work – development of chatbots for process automation.

Subject of the research – creation of a chatbot for submitting electricity meter readings.

The obtained results can be used to optimize hotel business processes and attract new clients through the use of a website.

The qualification work contains 50 pages, 6 figures, and a list of 41 references.

Keywords: chatbot, meter, convenience, information.

Approval of research results. The main results of the study were presented at the conference “Innovative Technologies in Energy and Automation” (Kyiv, March 15, 2025).

Year of the qualification work completion: 2025

Year of the work defense: 2025

Зміст

ВСТУП 3
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 6
1.1 Сучасний стан систем передачі показників лічильників за електроенергію в Україні 6
1.2. Аналіз інструментів і технологій для розробки чат-ботів 8
РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ВЕБ-САЙТУ 18
2.1. Принципи дизайну та етапи створення чат-боту 18
2.2. Архітектура чат-боту та інтеграція з системами обліку електроенергії22
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТУ 38
ВИСНОВКИ 43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 45

ВСТУП

Сучасний світ важко уявити без використання інформаційних технологій, які значно спрощують взаємодію між підприємствами та клієнтами. Зокрема, автоматизація процесів збору даних, таких як передача показників лічильників за електроенергію, набуває все більшої популярності. Інтернет-технології дозволяють користувачам швидко та зручно виконувати повсякденні завдання, використовуючи різноманітні гаджети та програми.

Чат-боти стали важливим інструментом у сфері автоматизації комунікацій, забезпечуючи оперативне вирішення запитів користувачів без необхідності залучення людських ресурсів. Вони дозволяють оптимізувати бізнес-процеси, зменшуючи витрати часу та підвищуючи якість обслуговування. У сфері енергопостачання чат-боти можуть значно спростити процес передачі показників лічильників, мінімізуючи помилки та забезпечуючи зручність для клієнтів.

Актуальність теми зумовлена необхідністю вдосконалення взаємодії між споживачами електроенергії та постачальниками. Зростання попиту на автоматизовані рішення для передачі показників лічильників вказує на потребу в розробці зручних та ефективних інструментів. Чат-боти дозволяють не лише полегшити цей процес для користувачів, але й оптимізувати роботу енергетичних компаній, зменшуючи навантаження на кол-центри та підвищуючи точність даних.

Проблему автоматизації процесів у сфері енергопостачання розглядали в своїх працях учені .

Мета кваліфікаційної роботи – розробка чат-боту для автоматизації передачі показників лічильника за електроенергію, що забезпечить зручність для користувачів та підвищить ефективність обробки даних для постачальників енергії. Зі сторони користувачів це досягається шляхом створення інтуїтивно зрозумілого інтерфейсу для швидкої передачі даних, а зі сторони постачальників – завдяки автоматизації збору та обробки інформації.

Для досягнення мети необхідно вирішити такі *завдання*:

Провести аналіз предметної області: дослідити сучасні системи передачі показників лічильників та визначити актуальність використання чат-ботів у цій сфері.

Розробити дизайн та логіку чат-боту: вивчити сучасні підходи до створення чат-ботів та розробити зручний і функціональний інтерфейс, що відповідає потребам користувачів.

Реалізувати програмну частину чат-боту: створити чат-бот, який включає функціонал для введення показників лічильників, перевірки даних та їх передачі до системи постачальника.

Забезпечити інтеграцію з системами обліку: реалізувати можливість взаємодії чат-боту з базами даних енергетичних компаній для автоматичного збереження та обробки отриманих даних.

Оптимізувати продуктивність: забезпечити швидку роботу чат-боту та мінімізувати час обробки запитів користувачів.

Об'єкт кваліфікаційної роботи – розробка чат-ботів для автоматизації процесів.

Предмет дослідження – створення чат-боту для передачі показників лічильника за електроенергію.

Методи дослідження:

Теоретичні: аналіз літератури, узагальнення та систематизація даних з нормативних документів, наукових статей та Інтернет-ресурсів для визначення теоретичних основ розробки чат-ботів.

Емпіричні: аналіз досвіду впровадження чат-ботів у сфері енергопостачання для оцінки стану вирішення проблеми.

База дослідження – чат-бот для передачі показників лічильника, інтегрований із системами обліку електроенергії.

Науково-практична значимість одержаних результатів полягає у створенні інструменту, який сприяє спрощенню передачі показників лічильників, підвищенню точності даних та зменшенню навантаження на операторів енергетичних компаній. Розроблений чат-бот дозволить

користувачам швидко та зручно передавати показники, а постачальникам – оптимізувати процеси збору та обробки інформації, що підвищить ефективність їхньої роботи.

Практичне значення чат-боту полягає в його здатності:

По-перше, полегшити взаємодію між споживачами та енергетичними компаніями, зменшивши кількість помилок при передачі даних.

По-друге, підвищити рівень автоматизації в енергетичному секторі, що сприятиме економії ресурсів та покращенню якості обслуговування.

Апробація результатів роботи. Основні результати дослідження представлено на конференції «Інноваційні технології в енергетиці та автоматизації» (м. Київ, 15 березня 2025 р.).

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Сучасний стан систем передачі показників лічильників за електроенергію в Україні

Станом на 2025 рік в Україні спостерігається поступове впровадження сучасних систем обліку та передачі показників лічильників електроенергії, проте рівень автоматизації та поширення смарт-технологій залишається обмеженим.

Близько 83% лічильників в Україні є електронними, що є необхідною умовою для інтеграції в автоматизовані системи обліку, однак лише 8% з них обладнані функцією дистанційного збору даних та можливістю обліку за багатозонними тарифами. Менше 5% лічильників інтегровані в автоматизовані системи комерційного обліку, які дозволяють дистанційно знімати показання та виявляти втрати електроенергії.

У деяких регіонах впроваджуються пілотні проєкти, зокрема у Волинській області розгортається система смарт-лічильників у Луцьку та навколишніх селах з метою зменшення втрат електроенергії та можливістю застосування різних тарифів, а в Миргороді в навчальних закладах запроваджено автоматичний онлайн моніторинг енергоспоживання для оперативного реагування на відхилення та оптимізації витрат. Серед технологій передачі даних поширення набуває LoRaWAN, яка є ефективною у сільській місцевості та невеликих містах завдяки низькому енергоспоживанню та широкому радіусу дії. На сьогодні споживачі можуть передавати показники лічильників через офіційні сайти постачальників електроенергії, мобільні додатки, боти в месенджерах Viber і Telegram, а також за допомогою SMS на короткі номери. Хоча в Україні відбуваються позитивні зміни в сфері обліку електроенергії, повноцінне впровадження сучасних технологій ще потребує значних інвестицій, часу та підтримки на державному рівні, однак локальні ініціативи та поступове використання інноваційних рішень створюють передумови для покращення ситуації в найближчому майбутньому.

У контексті впровадження сучасних систем обліку електроенергії в Україні важливо також враховувати роль законодавчої бази та стратегічних ініціатив держави, спрямованих на модернізацію енергетичного сектору. Зокрема, Національна комісія, що здійснює державне регулювання у сферах енергетики та комунальних послуг, визначила розвиток інтелектуальних систем обліку як пріоритетний напрямок. У рамках відповідних стратегічних документів, включаючи Енергетичну стратегію України до 2035 року, передбачено масштабне впровадження смарт-лічильників, що дозволяють здійснювати двосторонній обмін інформацією між споживачем і постачальником.

Крім того, впровадження смарт-лічильників пов’язане з розвитком концепції «розумного будинку» та «розумного міста», де автоматизовані системи обліку є одним із ключових елементів енергоефективності. Завдяки поєднанню сучасних лічильників з системами енергоменеджменту, споживачі можуть отримувати аналітику споживання в реальному часі, що сприяє економії ресурсів та зменшенню навантаження на мережу в пікові години.

На рівні громад важливим фактором успішного впровадження новітніх технологій є підвищення обізнаності населення щодо переваг смарт-обліку, зокрема прозорості нарахування, можливості використання багатозонних тарифів, уникнення помилок під час зняття показників та зменшення кількості побутових конфліктів щодо обсягів споживання. Також зростає попит на цифрові сервіси, які дозволяють не тільки передавати показники, але й керувати споживанням через мобільні застосунки та інтернет-платформи.

Паралельно з цим, одним із викликів залишається технічна готовність мереж та житлового фонду до впровадження нових систем. Багато старих будівель не мають необхідної інфраструктури або потребують модернізації щитових і проводки. Крім того, існує проблема кібербезпеки — з огляду на цифрову природу смарт-лічильників, необхідно впроваджувати надійні протоколи захисту даних, аби уникнути втручання в систему або викрадення персональної інформації споживачів.

Загалом, перспективи розвитку систем автоматизованого обліку електроенергії в Україні виглядають позитивно, хоча темпи впровадження залишаються нерівномірними. Успішна реалізація таких проєктів потребує скоординованої роботи держави, бізнесу та громадянського суспільства, а також залучення міжнародної технічної допомоги, зокрема у формі грантів, кредитів та технологічного ноу-хау.

1.2. Аналіз інструментів і технологій для розробки чат-ботів

Розробка чат-ботів є складним процесом, який включає вибір відповідних технологій і інструментів для забезпечення ефективної взаємодії між користувачем і системою. Вибір технологій залежить від кількох факторів, таких як складність завдань, тип платформи, вимоги до інтеграції та безпеки. У цьому підрозділі буде розглянуто основні інструменти і технології, що використовуються для створення чат-ботів, зокрема для автоматизації передачі показників лічильників за електроенергію.

Платформи для створення чат-ботів є важливим елементом у процесі розробки. Однією з найбільш популярних є Dialogflow від Google, яка підтримує інтеграцію з різними каналами комунікації, включаючи месенджери, веб-сайти та голосових помічників. Ця платформа використовує штучний інтелект для розпізнавання намірів користувачів і обробки складних запитів. Для задач автоматизації збору показників лічильників цей інструмент є ефективним завдяки можливості обробляти різноманітні типи даних і інтегруватися з іншими сервісами.

Іншим популярним інструментом є Microsoft Bot Framework, який надає набір інструментів для розробки, тестування та деплою чат-ботів. Ця платформа дозволяє створювати боти для різних каналів, таких як Skype, Facebook Messenger, а також для голосових помічників. Вона включає інструменти для обробки природної мови (NLP) і може бути корисною для інтеграції з системами обліку електроенергії.

Rasa, платформа з відкритим кодом, є ще одним потужним інструментом для створення чат-ботів, орієнтованих на глибоке навчання. Вона дозволяє

розробляти боти, які можуть ефективно розпізнавати контекст і підтримувати складні діалоги. Rasa може бути особливо корисною для побудови чат-ботів для передачі показників лічильників, оскільки вона дозволяє інтегрувати зовнішні системи та бази даних, а також налаштовувати персоналізовані взаємодії.

Для створення чат-ботів на основі месенджера Facebook можна використовувати платформу ManyChat, яка має зручний інтерфейс drag-and-drop, що дозволяє створювати бота без глибоких знань програмування. Вона також підтримує інтеграцію з SMS та веб-сайтами. Ця платформа може бути корисною для реалізації простих сценаріїв передачі даних від користувачів.

Обробка природної мови (NLP) є важливою частиною чат-ботів, оскільки дозволяє ботам розуміти запити користувачів і правильно на них реагувати. Технології NLP, такі як spaCy, дають можливість розпізнавати сутності в тексті, аналізувати граматику та структуру. Це особливо корисно для чат-ботів, які повинні обробляти текстові запити користувачів, наприклад, для виявлення дат чи цифр, коли клієнти передають показники лічильників. Бібліотека NLTK також забезпечує аналіз тексту, включаючи токенізацію, стемінг та лематизацію, що є корисним для обробки вхідних запитів.

Для глибшого розуміння контексту тексту можна використовувати технології, як BERT, що використовують трансформери для покращення якості розпізнавання намірів і сутностей. Це особливо важливо для чат-ботів, які працюють з різноманітними запитами і повинні ефективно розпізнавати контекст кожного.

Інтеграція з базами даних є важливим аспектом для чат-ботів, які повинні працювати з великими об'ємами даних, наприклад, для збереження показників лічильників. Для цього використовуються технології, такі як REST API, що дозволяють чат-ботам отримувати та передавати дані між собою і системами обліку. Вебхуки також дозволяють відправляти дані у реальному часі, що є корисним для автоматичної передачі показників.

SQL та NoSQL бази даних використовуються для зберігання інформації. Для структурованих даних, таких як показники лічильників, часто використовуються SQL бази даних, наприклад, MySQL або PostgreSQL, тоді як NoSQL бази даних, такі як MongoDB, використовуються для зберігання неструктурованих даних.

Захист даних є важливим аспектом при розробці чат-ботів, особливо коли йдеться про передачу персональної інформації. Для цього використовуються протоколи шифрування, такі як TLS/SSL, що забезпечують конфіденційність даних, а також технології аутентифікації та авторизації, такі як OAuth2, JWT та двофакторна аутентифікація.

Вибір технологій для розробки чат-бота залежить від специфіки задачі. Для автоматизації передачі показників лічильників важливо вибрати інструменти, які забезпечують надійну обробку даних, інтеграцію з існуючими системами і високий рівень безпеки для захисту персональних даних користувачів.

У процесі розробки чат-ботів важливо також враховувати інтерфейси користувача та їх зручність. Інтерфейс чат-бота має бути інтуїтивно зрозумілим та простим для взаємодії, що дозволить користувачам без труднощів передавати показники лічильників та отримувати відповідні підтвердження або зворотний зв'язок. Для цього можуть бути використані такі інструменти, як BotUI або Botpress, які пропонують готові компоненти для створення зручних інтерфейсів без необхідності в складному програмуванні. Ці інструменти дозволяють інтегрувати візуальні елементи, такі як кнопки та меню, що покращують взаємодію з користувачем.

Однією з важливих складових є також тестування чат-ботів, яке дозволяє виявити недоліки в роботі системи та покращити її функціональність. Для тестування бота можна використовувати спеціалізовані інструменти, такі як Botium, що дозволяють автоматизувати перевірку чат-ботів на відповідність очікуваним результатам та коректність обробки запитів. Це дозволяє уникнути помилок при інтеграції та на етапі експлуатації.

У контексті автоматизації передачі показників лічильників важливим є також вибір платформи для інтеграції з існуючими системами обліку. Чат-боти, які взаємодіють із базами даних постачальників енергетичних послуг, повинні забезпечувати безперебійну та безпечну передачу даних. Це можливе через використання REST API для зв'язку з різними сервісами, що дозволяє швидко отримувати дані від користувачів і передавати їх в базу даних. Для більш ефективної роботи можна використовувати технології, які підтримують синхронізацію даних у реальному часі, що важливо для актуальності показників лічильників.

Особливою увагою повинна користуватися безпека даних, оскільки чат-боти працюють із персональними та чутливими даними. Для забезпечення захисту інформації можна використовувати технології шифрування, а також реалізувати процедури автентифікації, щоб уникнути несанкціонованого доступу до даних. Важливо також забезпечити відповідність стандартам і регламентам захисту даних, таким як GDPR, для забезпечення прав користувачів на конфіденційність.

Таким чином, вибір технологій для розробки чат-бота є ключовим аспектом для забезпечення ефективної роботи системи автоматизації передачі показників лічильників. Успішна реалізація таких проектів вимагає комплексного підходу до вибору інструментів, інтеграції з існуючими системами, забезпечення зручності користування і високого рівня безпеки.

У розробці чат-ботів для автоматизації передачі показників лічильників важливо також враховувати масштабованість і адаптивність системи. Оскільки обсяг користувачів може значно змінюватися, важливо, щоб чат-бот був здатний обробляти великий потік запитів одночасно без втрати продуктивності. Для цього можна використовувати хмарні платформи, як AWS або Google Cloud, які забезпечують високу доступність та масштабованість, дозволяючи підвищити ефективність обробки запитів у пикові моменти навантаження.

Інтеграція чат-ботів з аналітичними інструментами також є важливою для оптимізації роботи системи. Вона дозволяє збирати статистичні дані про використання бота, з'ясувати, які функції є найбільш популярними, виявити можливі проблеми на етапах взаємодії з користувачем. Наприклад, за допомогою аналітики можна виявити найбільш поширені помилки при введенні показників або зрозуміти, де користувачі стикаються з труднощами. Такі інсайти дозволяють удосконалювати чат-бота, оптимізуючи його взаємодію з користувачами та підвищуючи точність збору даних.

Ще одним важливим аспектом є забезпечення багатомовності чат-бота. Оскільки в Україні існує кілька офіційних мов (українська та англійська), а також великі етнічні та мовні групи, підтримка кількох мов може значно покращити взаємодію з користувачами та зробити процес передачі показників доступним для ширшого кола споживачів. Це також дозволяє забезпечити більш високий рівень інклюзивності для користувачів, що є важливим аспектом соціальної відповідальності бізнесу.

Розробка чат-бота також передбачає постійне вдосконалення та оновлення його функцій. З часом можуть виникати нові технології та методи обробки даних, які дозволяють значно покращити продуктивність бота або додати нові корисні функції. Важливою частиною цього процесу є спрощення взаємодії з користувачем, а також підвищення безпеки та ефективності процесів обробки даних.

Крім того, важливим є тестування та зворотний зв'язок від користувачів. Розробка чат-бота не завершується після запуску: необхідно постійно оцінювати ефективність його роботи через зворотний зв'язок від кінцевих користувачів і проводити тестування. Це дозволяє виявляти потенційні проблеми, такі як неочікувані помилки у процесі введення даних, та здійснювати коригування у системі.

Таким чином, процес розробки чат-бота для автоматизації передачі показників лічильників є комплексним і багатоетапним. Від правильного вибору технологій та інструментів залежить не тільки функціональність і

зручність використання бота, а й ефективність збору даних та інтеграція з іншими системами. Успішна реалізація проекту дозволить значно полегшити взаємодію між користувачами та енергетичними компаніями, покращити точність даних та зменшити навантаження на обслуговуючий персонал, що в кінцевому підсумку сприятиме підвищенню ефективності та якості обслуговування.

Таким чином процес розробки чат-бота для автоматизації передачі показників лічильників є комплексним і багатоетапним. Від правильного вибору технологій та інструментів залежить не тільки функціональність і зручність використання бота, а й ефективність збору даних та інтеграція з іншими системами. Успішна реалізація проекту дозволить значно полегшити взаємодію між користувачами та енергетичними компаніями, покращити точність даних та зменшити навантаження на обслуговуючий персонал що в кінцевому підсумку сприятиме підвищенню ефективності та якості обслуговування. Одним з інструментів який активно використовується для розробки таких систем є Visual Studio Code. Це потужний редактор коду який надає зручне середовище для програмування та тестування чат-ботів завдяки підтримці різних мов програмування численним плагінам і можливості інтеграції з багатьма іншими інструментами що значно спрощує процес розробки

Visual Studio Code є безкоштовним кросплатформеним редактором який підтримує численні мови програмування серед яких JavaScript Python Java C++ PHP та багато інших. Завдяки великій кількості доступних плагінів VS Code стає одним із найбільш популярних інструментів для розробки чат-ботів зокрема тих що автоматизують передачу показників лічильників електроенергії. Він дозволяє зручно працювати з різними фреймворками для створення чат-ботів такими як Microsoft Bot Framework Rasa чи Dialogflow і підтримує інтеграцію з зовнішніми системами що робить його ідеальним вибором для розробників

Botfather — це офіційний бот Telegram, який використовується для створення та налаштування нових чат-ботів на платформі Telegram. Це зручний інструмент для розробників, адже через нього можна легко створити нового бота, отримати його токен для доступу до Telegram API та налаштувати основні параметри бота, не маючи потреби в програмуванні чи роботі безпосередньо з API. Для початку необхідно знайти Botfather в Telegram. Це можна зробити, шукаючи його за допомогою пошукового рядка (BotFather). Після того, як почнеться чат з Botfather, потрібно надіслати команду /newbot для створення нового бота. Botfather попросить вас вказати ім’я для бота та його унікальний username (який має закінчуватися на "bot", наприклад, mynewbot). Після успішного створення бота, Botfather надасть токен — спеціальний ключ, який дозволяє взаємодіяти з API Telegram. Цей токен необхідний для програмного доступу до чат-бота. Також можна налаштувати різні параметри бота через Botfather, такі як опис, зображення для аватара, команди, що будуть підтримуватись ботом, тощо. За допомогою команд, таких як /setdescription, /setabouttext, /setuserpic тощо, можна змінювати інформацію про бота. Через Botfather також можна додавати, видаляти або змінювати команди бота (наприклад, /start, /help), які користувачі будуть використовувати для взаємодії з ним. Основною задачею Botfather є спрощення процесу створення чат-ботів у Telegram. Він автоматизує створення бота, надаючи токен та допомагаючи в налаштуванні базових параметрів. З допомогою токену, наданого Botfather, розробник може підключити свого бота до Telegram API та почати працювати з ним через програмний код. Botfather забезпечує інтерфейс для подальшого управління ботом, зокрема для налаштування його параметрів і команд, які будуть доступні користувачам. Якщо є кілька ботів, Botfather дозволяє легко управляти ними за допомогою одного чату, допомагаючи розробнику зручніше налаштовувати різні боти без необхідності створювати нові акаунти чи з’єднувати їх із додатковими інтерфейсами. Таким чином, Botfather є основним інструментом для розробників, який дозволяє швидко створити

бота, налаштувати його, а також отримати доступ до Telegram API для подальшої розробки та налаштування чат-бота відповідно до потреб бізнесу чи проекту.

Бази даних є невід'ємною частиною будь-якої інформаційної системи, включаючи чат-ботів, оскільки вони забезпечують зберігання, обробку та доступ до великих обсягів даних. Для чат-ботів, які обробляють і зберігають інформацію про користувачів, їх запити та інші пов'язані дані, бази даних є критично важливими для ефективної роботи системи. Система управління базами даних (СКБД) є програмним забезпеченням, що дозволяє створювати, читати, оновлювати та видаляти дані з бази даних. Бази даних можуть бути різними в залежності від їх структури, типу та способу організації даних. Найбільш популярними є реляційні (SQL) та нереляційні (NoSQL) бази даних. SQL (Structured Query Language) — це мова запитів, яка використовується для роботи з реляційними базами даних. Реляційна база даних — це тип бази даних, в якій дані організовані у вигляді таблиць, що пов'язані між собою через ключі. Основною особливістю таких баз є структурованість даних, що дозволяє зберігати інформацію в таблицях і взаємодіяти з нею за допомогою SQL-запитів. Реляційні бази даних мають такі основні характеристики: таблиці, в яких зберігаються дані, кожен рядок таблиці відповідає окремому запису, а кожна колонка — певному атрибуту даних, зв'язки між таблицями, що дозволяють зберігати відносини між записами, нормалізацію даних, що запобігає надмірності та забезпечує ефективне оновлення даних, а також ACID-принципи (атомарність, узгодженість, ізоляція та довговічність), які забезпечують надійність транзакцій. Популярні системи реляційних баз даних включають MySQL, PostgreSQL і Microsoft SQL Server, кожна з яких має свої особливості. MySQL є однією з найбільш поширених СКБД з відкритим кодом, що підтримує SQL. MySQL часто використовується для веб-додатків і є чудовим вибором для чат-ботів, де потрібна висока швидкість обробки запитів і надійність. PostgreSQL — потужна реляційна база даних, яка підтримує розширені можливості для роботи з даними, такі як збереження

географічних даних, що може бути корисно для складніших чат-ботів. Microsoft SQL Server є популярною в корпоративному середовищі, де важлива масштабованість та безпека. SQL (Structured Query Language) є мовою, яка дозволяє здійснювати запити до реляційних баз даних. Основні операції в SQL включають SELECT, який використовується для отримання даних з таблиці, INSERT, що додає нові дані в таблицю, UPDATE, що оновлює наявні дані, і DELETE, що видаляє дані з таблиці. Запити можуть бути складними, включаючи фільтрацію, сортування, об'єднання таблиць (JOIN), підзапити та агрегування даних, що дозволяє ефективно працювати з великими наборами структурованих даних. Для чат-ботів використання SQL дозволяє зберігати дані про користувачів, їх запити та інші дані, які можуть бути необхідні для аналізу чи персоналізації відповіді. Наприклад, зберігання даних про показники лічильників, запити користувачів на передачу показників, час останньої взаємодії та інші параметри може бути здійснене через реляційну базу даних. Таким чином, реляційні бази даних і SQL є важливими інструментами для зберігання та обробки даних у чат-ботах, зокрема для тих, які займаються автоматизацією процесів, таких як передача показників лічильників. Вони дозволяють не тільки зберігати інформацію, але й ефективно здійснювати її обробку, запити та аналіз, що є критично важливим для надання точних і швидких відповідей користувачам.

Загальний висновок щодо використання BotFather, SQL, Visual Studio Code та розробки телеграм-бота для передачі показів лічильника можна сформулювати так: створення ефективного телеграм-бота для автоматизації процесу передачі показників лічильників електроенергії вимагає поєднання потужних інструментів і технологій, що дозволяють забезпечити надійність, зручність та безпеку.

BotFather є важливим інструментом для створення та налаштування самого бота в Telegram, даючи змогу легко створити бота, налаштувати його параметри та отримати токен для підключення до API Telegram. Відповідно,

це значно спрощує процес інтеграції та стартової настройки бота, зокрема для передачі показників лічильників.

SQL-бази даних є важливою частиною для ефективного зберігання та обробки даних, що передаються користувачами через бот. Завдяки реляційним базам даних можна організувати правильне зберігання показників лічильників, а також здійснювати складні запити для виведення, аналізу та оновлення цих даних. Використання SQL дозволяє ефективно організувати структуру даних, забезпечити безпеку і доступність даних, а також надавати точні й швидкі відповіді на запити користувачів.

Visual Studio Code, як потужний і універсальний редактор, надає всі необхідні інструменти для розробки, тестування та дебагінгу коду бота. Він підтримує різні мови програмування, даючи змогу розробникам працювати в зручному середовищі для створення кросплатформених рішень, а також має потужну підтримку для роботи з Git, що дозволяє ефективно керувати версіями проекту.

Загалом, реалізація телеграм-бота для передачі показників лічильників електроенергії є складним, але високоефективним процесом, який вимагає інтеграції різних технологій та інструментів. Використання BotFather для налаштування бота, SQL для зберігання та обробки даних, а також Visual Studio Code для розробки і тестування коду дозволяє створити зручну та надійну систему автоматизованого збору показників. Це забезпечує зручність для користувачів, дозволяє зменшити навантаження на персонал і підвищує точність даних, що в результаті сприяє покращенню обслуговування та ефективності роботи енергетичних компаній.

РОЗДІЛ 2. ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ВЕБ-САЙТУ

2.1. Принципи дизайну та етапи створення чат-боту

Створення чат-боту для подачі показників світла передбачає ретельне планування як з точки зору дизайну, так і технічної реалізації. Принципи дизайну чат-боту повинні включати зрозумілість і простоту, адже інтерфейс повинен бути інтуїтивно зрозумілим для користувача, з використанням простих запитів та відповідей. Важливо, щоб бот мав чітку мету і функціональність, щоб користувачі могли одразу зрозуміти, як саме він працює, наприклад, для подачі показників світла[39].

Додатково, важливо забезпечити персоналізацію, щоб бот міг надавати індивідуальні рекомендації або дані відповідно до попередніх запитів чи профілю користувача.

Технічна коректність також є важливою, адже бот має точно обробляти дані і надавати правильну інформацію, а також здійснювати надійний зв’язок із системою або базою даних для отримання актуальних даних.

Взаємодія з користувачем також має бути ключовим елементом, оскільки важливо додати можливість уточнювати або змінювати запити. Бот повинен підтримувати різні типи запитів, такі як введення тексту, вибір з меню або голосові команди.

Доступність і зручність також є важливими принципами дизайну, оскільки бот має бути зручним у використанні на різних пристроях та інтегрованим з різними платформами, такими як месенджери.

Створення чат-боту складається з кількох етапів. Спочатку потрібно провести аналіз вимог, з’ясувавши потреби користувачів та цілі чат-боту, визначити потенційні сценарії використання, наприклад, коли і як часто користувачі подаватимуть показники світла. На наступному етапі проектується архітектура бота: розробляється схема взаємодії користувача з ботом, структура діалогів, а також визначається, з якими системами та базами даних бот має взаємодіяти для отримання та подачі показників світла. Потім створюється прототип бота, де розробляється інтерфейс користувача, а також

визначаються основні сценарії та діалоги. Після цього розробляється функціонал бота: програмується взаємодія бота з користувачем, обробка даних та валідація введених показників світла, а також інтеграція з базою даних або API для обробки інформації. Після цього проводиться тестування: тестуються різні сценарії використання, виявляються потенційні помилки, а також перевіряється, як бот реагує на неправильні введення. Після успішного тестування бот запускається в продуктивне середовище, і починається моніторинг його роботи, збирання зворотного зв'язку від користувачів і аналіз статистики використання(див. рис. 2.1).

Оновлення та оптимізація бота також є важливими етапами після запуску, оскільки на основі отриманих даних можна вдосконалювати функціональність бота та покращувати користувацький досвід.

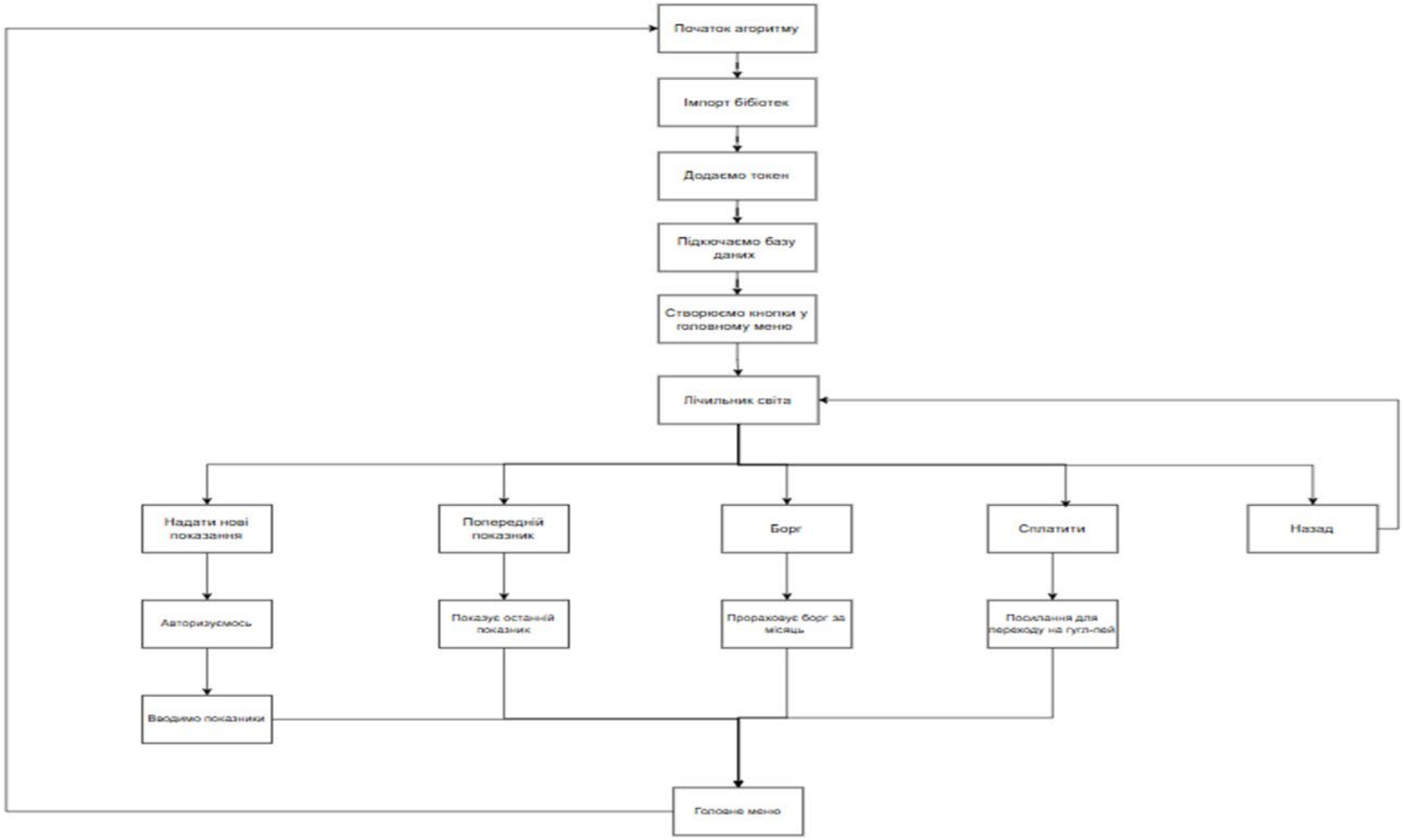


Рис.2.1 Блок-схема алгоритму роботи чат-бот

Алгоритм починається з імпорту необхідних бібліотек. Після цього додається токен доступу та підключається база даних. Далі створюються кнопки для головного меню, після чого користувач переходить до головного розділу — "Лічильник сайта". У цьому меню користувач має кілька варіантів дій: надати нові показання, переглянути попередній показник, перевірити борг, здійснити оплату або повернутися назад.

Якщо користувач обирає "Надати нові показання", система спочатку проводить авторизацію, після чого дає можливість ввести нові показання. Якщо вибрано "Попередній показник", користувачу показується останнє збережене значення. При виборі "Борг" програма показує суму боргу за поточний місяць. Якщо обрано "Сплатити", користувач отримує посилання для переходу на платіжну платформу (наприклад, Google Pay). Кнопка "Назад" дозволяє повернутися до головного меню. Після виконання будь-якої з цих дій користувач автоматично перенаправляється назад у головне меню для подальшої роботи.

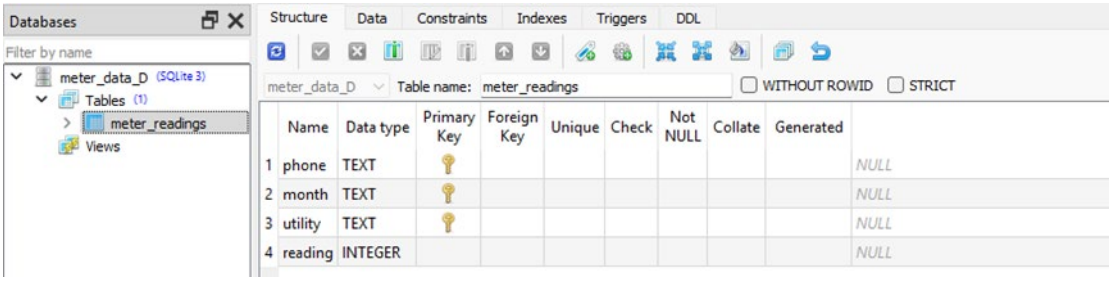


Рис.2.2 База даних

База даних meter_data_D(див. рис. 2.2), яка використовується в чат-боті для збору та обробки показників лічильників, містить одну основну таблицю - meter_readings. Ця таблиця призначена для збереження інформації про щомісячні показання, які користувачі передають через бот. Вона складається з чотирьох основних полів:

phone (TEXT) - зберігає номер телефону користувача. Це ключовий ідентифікатор, який дозволяє зв’язати кожен запис з конкретним користувачем. Це поле використовується як частина складеного первинного ключа.

month (TEXT) - позначає місяць, за який подано показання. Значення зберігається у форматі, наприклад, 2025-04, що дозволяє точно вказувати період. Це поле також входить до складеного первинного ключа.

utility (TEXT) - тип комунальної послуги, для якої подано показання. Наприклад: electricity (світло), water (вода), gas (газ), heating (опалення). Це дозволяє використовувати одну таблицю для збереження показників з різних лічильників. Це поле теж входить до складеного первинного ключа.

reading (INTEGER) - фактичне значення показника лічильника, яке користувач подав через бот.

Три ключі — phone, month та utility — використовуються в складеному первинному ключі для забезпечення унікальності кожного запису. Це дозволяє уникнути дублювання даних, оскільки один користувач може подати показники для різних місяців та типів послуг. Така структура також забезпечує легку обробку та порівняння даних за періоди, полегшує аналітику та формування звітів, а також дозволяє зберігати історію показників для кожного користувача.

Ця структура дозволяє чат-боту:

- Зберігати історію показників для кожного користувача.
- Обчислювати споживання за періоди, порівнюючи показники різних місяців.
- Формувати звіти та розраховувати умовну заборгованість.
- Надавати нагадування, якщо показання за поточний місяць ще не передано.

У перспективі до цієї бази можна додати інші таблиці, наприклад users (для зберігання імені, адреси тощо) або rates (для тарифів), які можна зв’язати по phone чи utility.

2.2. Архітектура чат-боту та інтеграція з системами обліку електроенергії

Телеграм-боти часто пишуть на Python з кількох причин, які роблять цю мову програмування ідеальним вибором для створення таких ботів.

- має безліч бібліотек і фреймворків, що значно спрощує процес створення бота. Найпопулярніша з них - python-telegram-bot, яка забезпечує зручний інтерфейс для роботи з Telegram Bot API. Це дозволяє програмістам швидко створювати боти без необхідності занурюватися в складні деталі API.
- відомий своєю простотою і читабельністю. Завдяки чистому і інтуїтивно зрозумілому синтаксису, Python дозволяє швидко розробляти код навіть тим, хто тільки починає програмувати.
- має величезну спільноту розробників, що завжди готові допомогти. Це означає, що можна легко знайти документацію, приклади коду, а також поради та рішення на форумах чи в соцмережах. Також доступні готові рішення для різних задач, таких як інтеграція з базами даних, розпізнавання зображень чи машинне навчання.
- надає зручні інструменти для створення асинхронних програм (за допомогою бібліотеки asyncio), що дозволяє боту працювати з великою кількістю одночасних запитів. Це особливо важливо для чат-ботів, які повинні обробляти велику кількість повідомлень за короткий час.
- дозволяє легко інтегруватися з різними зовнішніми сервісами та базами даних. Наприклад, можна використовувати SQLAlchemy для роботи з базами даних, а також бібліотеки для роботи з API сторонніх сервісів (як-от платіжні системи, сервіси для обробки платежів чи штучного інтелекту).
- кросплатформенний, що означає, що створений бот буде працювати на різних операційних системах, таких як Windows, Linux, MacOS, без необхідності додаткових налаштувань. Це дуже зручно при розгортанні бота на різних серверах.
- лідер у цих галузях. Бібліотеки як-от TensorFlow, PyTorch, Keras і багато інших дозволяють легко інтегрувати в бота функції, що використовують AI або ML.

- Python і бібліотеки для роботи з Telegram боти мають детальну документацію, що дозволяє швидко освоїтися навіть початківцям.
- Python-програми легко розгортаються на різних хостингах, таких як Heroku, AWS, DigitalOcean, що робить деплой бота швидким та простим. Крім того, Python підтримує різні види віртуалізацій (наприклад, через Docker), що дозволяє створювати та запускати бота в ізольованому середовищі.
- дозволяє будувати масштабовані рішення для ботів. За допомогою таких технологій, як Celery для обробки фонових задач або використання мікросервісів, можна створити ботів, які обробляють великий обсяг запитів.

Телеграм-боти пишуть на Python, бо ця мова програмування надає потужні інструменти для швидкої розробки, підтримує асинхронні операції, має великі можливості для інтеграції з іншими системами та має величезну спільноту розробників. Крім того, є чудовим вибором для створення ботів, які використовують штучний інтелект або машинне навчання, завдяки широкому набору бібліотек у цих галузях.

Токен Telegram-бота — це необхідний елемент для будь-якого бота, який працює з Telegram API. Він забезпечує ідентифікацію бота, дозволяє йому отримувати доступ до функцій Telegram і гарантує безпеку та правильну взаємодію з сервісом. Токен є основним інструментом, який дозволяє боту "розмовляти" з Telegram і обробляти запити від користувачів.

Наступні лістинги реалізують чат-бота для Telegram, який дозволяє користувачам вводити показники лічильника світла, перевіряти попередні показники, розраховувати борг на основі тарифу, а також здійснювати сплату (віртуально через Google Pay). При запуску бота користувач отримує головне меню з опцією для введення показників лічильника світла. Якщо користувач вибирає "Надати нові показання", бот запитує номер телефону (з валідацією формату), після чого користувач вводить показник лічильника. Після введення показника, бот зберігає його в базу даних SQLite. Якщо користувач хоче перевірити попередні показники, бот здійснює запит до бази даних і повертає

відповідне значення за попередній місяць. Крім того, бот дозволяє користувачеві розрахувати борг, порівнюючи показники поточного і попереднього місяця, враховуючи тариф на світло (4.32 грн за одиницю). Розрахунок боргу здійснюється за формулою: різниця між показниками * тариф. Бот також пропонує "псевдо" оплату через Google Pay, що є симуляцією. Всі введені показники зберігаються в базі даних за допомогою SQL запитів, а контекст користувача зберігається в пам'яті під час сеансу. Якщо користувач хоче почати спілкування заново, він може натискати "Почати спочатку", що очищає збережений контекст. Бот працює в режимі постійного опитування (polling), що дозволяє обробляти вхідні повідомлення без зупинок[25][36].

Лістинг1 - Імпорт бібліотек

import telebot

Імпортує бібліотеку Telebot (офіційна назва — pyTelegramBotAPI), яка використовується для створення Telegram-ботів. Вона дозволяє легко обробляти повідомлення, команди тощо.

import sqlite3

Імпортує модуль sqlite3, який дозволяє працювати з локальними базами даних SQLite. Ви можете створювати таблиці, зберігати дані, виконувати запити без окремого серверу бази даних.

import re

Імпортує модуль re, який забезпечує роботу з регулярними виразами у Python. Регулярні вирази потрібні для пошуку, валідації чи заміни частин тексту за певним шаблоном (наприклад, перевірка email чи витягування дати з тексту).

from datetime import datetime

Імпортує клас datetime з модуля datetime. Він використовується для роботи з датами і часом: можна отримати поточну дату, різницю між датами, форматувати дату тощо.

from dateutil.relativedelta import relativedelta

Імпортує relativedelta з пакета dateutil. Це розширений інструмент для обчислення різниці між датами. Наприклад, він дозволяє додати/відняти рівно 1 місяць або 1 рік (на відміну від звичайної різниці у днях).

Лістинг2 - Налаштування

TOKEN = '7750374616:AAHV436kO0ZBT45zT30cQI-2hJmRTm1qMFA'

Це токен Telegram-бота. Він унікальний для кожного бота і видається BotFather. Токен не має бути публічним (його можуть використати інші для управління ботом).

bot = telebot.TeleBot(TOKEN)

Створюється об'єкт бота через бібліотеку telebot, прив'язаний до вказаного токену. Тепер через цей об'єкт можна обробляти повідомлення, надсилати відповіді тощо.

db_path = 'meter_data_D.db'

Вказується шлях до файлу бази даних SQLite. У цьому файлі зберігатимуться дані (наприклад, показники лічильників або інформація про користувачів).

TARIFFS = {

"Світло": 4.32, # Тариф для світла

}

Це словник тарифів: наприклад, "Світло" коштує 4.32 гривні за одиницю споживання. У майбутньому можна легко додати сюди інші ресурси (наприклад, воду, газ).

user_context = {}

Це порожній словник для зберігання контексту користувача під час взаємодії з ботом. Наприклад, коли користувач вводить показники лічильника у кілька етапів, цей словник допомагає зберігати проміжні дані.

Лістинг3 - База даних

def connect_db():

return sqlite3.connect(db_path)

Функція для підключення до бази даних. Вона відкриває файл бази даних meter_data_D.db і повертає об'єкт підключення Connection. Такий підхід дозволяє не повторювати однаковий код підключення в кожній функції.

```
def create_table():
conn = connect_db()
cursor = conn.cursor()
cursor.execute('''
CREATE TABLE IF NOT EXISTS meter_readings (
phone TEXT,
month TEXT,
utility TEXT,
reading INTEGER,
PRIMARY KEY (phone, month, utility)
)
''')
conn.commit()
conn.close()
```

Функція створення таблиці meter_readings, якщо вона ще не існує:

- phone — номер телефону користувача;
- month — місяць у форматі YYYY-MM;
- utility — назва комунальної послуги (наприклад, "Світло");
- reading — показник лічильника.

PRIMARY KEY поєднує phone + month + utility, що гарантує, що на кожен місяць для одного користувача і певної послуги буде лише один запис. Це захищає від дублювання даних.

Лістинг4 - Дата та телефон

```
def get_month(delta=0):
now = datetime.now() + relativedelta(months=-delta)
return now.strftime("%Y-%m")
```

Функція отримання місяця у форматі YYYY-MM.

- Якщо delta=0 — повертає поточний місяць;

- Якщо delta=1 — повертає попередній місяць, і так далі.

Приклад:

- get_month(0) → '2025-04'

- get_month(1) → '2025-03'

def is_valid_phone(phone):

return re.fullmatch(r'0\d{9}', phone)

Функція перевірки валідності номера телефону.

Використовується регулярний вираз:

- Номер має починатися з 0;

- Далі мають бути рівно 9 цифр.

Приклад валідного номера: 0671234567

Якщо номер відповідає шаблону — поверне об'єкт співпадіння (True у логічних перевірках), інакше — None.

Лістинг5 - Початок роботи

@bot.message_handler(commands=['start'])

def start_handler(message):

Це обробник команди /start. Коли користувач пише /start, бот: створює клавіатуру з кнопкою "Лічильник світла"; надсилає повідомлення "Привіт! Натисніть на кнопку для роботи з лічильником світла."

@bot.message_handler(func=lambda msg: msg.text == "Лічильник світла")

def select_utility(message):

Це обробник натискання на кнопку "Лічильник світла": зберігає у user_context, що користувач працює з послугою "Світло". Показує нову клавіатуру з 4 діями: Надати нові показання, Попередній показник, Борг, Сплатити, Назад. Кнопки організовано у два стовпці для зручності (row_width=2).

@bot.message_handler(func=lambda msg: msg.text == "Надати нові показання")

def ask_phone(message):

Якщо користувач натискає "Надати нові показання": якщо у контексті (user_context) вже є номер телефону користувача, одразу запрошує нове показання. Якщо немає номера телефону — просить ввести телефон у форматі 0991234567.

def process_phone(message):

Функція, яка обробляє введений номер телефону- очищає текст (strip), перевіряє правильність через is_valid_phone(), якщо номер невірний — повідомляє і просить ввести ще раз, якщо номер правильний — зберігає телефон у user_context[chat_id]['phone']. Без збереження валідного номера не дозволяє рухатись далі.

Лістинг6 - Нуль за попередній місяць

init_month = get_month(1)

Отримує попередній місяць (наприклад, якщо зараз квітень — то 2025-03).

```
conn = connect_db()
cursor = conn.cursor()
cursor.execute("INSERT OR IGNORE INTO meter_readings (...) VALUES (?, ?, ?, ?)",
               (phone, init_month, "Світло", 0))
```

Додає нульовий запис у базу за минулий місяць (якщо його ще немає). Це важливо для обчислення різниці споживання в майбутньому. INSERT OR IGNORE означає: якщо такий запис уже є (тобто існує PRIMARY KEY), то нічого не робити.

Лістинг7 - Збереження нових показань

```
def process_reading(message):
   try:
      reading = int(message.text)
```

Отримує нове значення лічильника та перевіряє, що воно ціле число.

```
ctx = user_context[message.chat.id]
```

phone = ctx['phone']

utility = ctx['utility']

month = get_month()

Отримує дані з user_context- номер телефону, послугу ("Світло"), поточний місяць.

cursor.execute("REPLACE INTO meter_readings (...) VALUES (?, ?, ?, ?)",

(phone, month, utility, reading))

REPLACE означає- якщо запис із таким ключем вже є - він буде перезаписаний. Якщо немає - створюється новий. Це дозволяє оновлювати показання на той самий місяць.

bot.send_message(message.chat.id, "Показання збережено.")

Підтвердження користувачу, що все пройшло успішно.

except ValueError:

bot.send_message(message.chat.id, "Будь ласка, введіть ціле число.")

Якщо користувач ввів не число (наприклад, текст), бот просить його повторити.

Лістинг8 - Обробник кнопки "Попередній показник"

prev_month = get_month(1)

Отримує попередній місяць (наприклад, '2025-03').

cursor.execute("SELECT reading FROM meter_readings WHERE phone=? AND month=? AND utility=?",

(phone, prev_month, utility))

Запит до бази даних: знайти показання за попередній місяць для вказаного користувача і послуги.

Вивід результату:

Якщо дані знайдено — бот виводить: "Показання за 2025-03: 123"

Якщо ні — повідомляє: "Немає даних за попередній місяць."

Лістинг9 - Розрахунок боргу

ctx = user_context.get(message.chat.id)

Перевіряємо, чи користувач уже вводив дані.

curr_month = get_month()

prev_month = get_month(1)

Поточний і попередній місяці — для порівняння показників.

cursor.execute(...) # Поточне

cursor.execute(...) # Попереднє

З бази даних отримуються показники обох місяців для обраної послуги.

diff = current[0] - previous[0]

tariff = TARIFFS[utility]

*total = diff * tariff*

Різниця * тариф = сумарна сума до сплати.

bot.send_message(..., f"Борг за {utility}: {diff} x {tariff} = {total:.2f} грн")

Користувач отримає повідомлення на зразок:

"Борг за Світло: 45 x 4.32 = 194.40 грн"

Лістинг10 - Реалізація кнопки "Сплатити"

@bot.message_handler(func=lambda msg: msg.text == "Сплатити")

def fake_payment(message):

Реакція бота на натискання кнопки "Сплатити".

bot.send_message(message.chat.id, "Для сплати скористайтесь Google Pay (симуляція):\nhttps://pay.google.com")

Бот відправляє посилання на Google Pay, імітує реальну оплату. Це зручно на етапі тестування або якщо немає інтеграції з реальними платіжними системами.

Лістинг11 - Обробник кнопки "Почати спочатку"

@bot.message_handler(func=lambda msg: msg.text == "Почати спочатку")

def restart_bot(message):

user_context.pop(message.chat.id, None)

return start_handler(message)

Видаляє весь контекст користувача (user_context) для chat.id. Потім викликає start_handler(message) — тобто повертає користувача на початкове

меню, наче він щойно натиснув /start. Дуже зручно, коли користувач хоче змінити телефон або розпочати взаємодію з нуля.

Лістинг12: Запуск бота

```
if __name__ == "__main__":
    create_table()
    print("Бот запущено.")
    bot.polling(none_stop=True)
```

create_table() — гарантує, що таблиця у БД буде створена перед запуском.

bot.polling(none_stop=True) — запускає нескінченну обробку повідомлень від Telegram.

print("Бот запущено.") — технічне підтвердження у консолі.

Цей бот призначений для зручного збереження показань електролічильника, перегляду попередніх значень, обчислення боргу на основі фіксованого тарифу та симуляції оплати. Користувач починає взаємодію з командою /start, після чого йому пропонується вибрати послугу — "Лічильник світла". У меню користувач може надати нові показання, переглянути попередні, дізнатись суму боргу або імітувати оплату. Для надання показань необхідно ввести номер телефону у форматі 0XXXXXXXXX, після чого бот запитує поточні показання. Дані зберігаються у локальній базі SQLite із унікальним записом для кожної комбінації телефон + місяць + послуга. При обчисленні боргу бот порівнює поточне та попереднє значення, множить різницю на тариф (у випадку світла — 4.32 грн) і виводить суму. Також передбачена можливість повторного старту через кнопку "Почати спочатку", яка очищує сесію користувача. Бот використовує бібліотеки telebot, sqlite3, datetime, dateutil та re для своєї роботи, запускається через polling, автоматично створює таблицю в базі при старті. Це просте, але надійне рішення для побутових користувачів, яке легко адаптувати до інших комунальних послуг.

Інтеграція Telegram-бота з реальними системами обліку електроенергії можлива кількома способами залежно від типу лічильника, доступу до даних

і технічних можливостей користувача чи організації. Один із сучасних напрямів — підключення до так званих «розумних» лічильників (smart meters), які мають вбудовані модулі зв’язку (Wi-Fi, GSM, Zigbee) та можуть надавати дані через спеціалізовані API або протоколи обміну (наприклад, DLMS, MQTT, Modbus). Якщо виробник або енергокомпанія надає публічний доступ до API, можна інтегрувати ці дані напряму в Telegram-бота: за допомогою Python-бібліотек, таких як requests, бот може періодично або на запит користувача отримувати актуальні показання лічильника без потреби ручного введення.

Ще один популярний варіант — отримання даних з особистих кабінетів енергокомпаній, таких як YASNO, Обленерго тощо. Якщо в таких кабінетах є відкритий API — інтеграція буде швидкою і стабільною. Якщо API не надається, можлива реалізація через автоматизований парсинг сторінок (scraping), проте це менш надійно, потребує авторизації і часто суперечить умовам користування. У побутових чи офісних умовах також можливий варіант інтеграції через завантаження файлів — деякі системи обліку можуть експортувати показання у вигляді CSV, Excel або JSON. Такі файли можна періодично завантажувати або надсилати Telegram-боту, і він автоматично зчитуватиме звідти потрібну інформацію.

У промислових системах, де використовуються SCADA-рішення або протоколи на кшталт Modbus TCP/RTU, показання з лічильників можна зчитувати через окремі скрипти або сервери й надсилати до бота. У такому випадку Telegram-бот є лише інтерфейсом виведення або підтвердження дій, тоді як основна логіка обміну йде через локальні мережі або контролери.

Для реалізації повноцінної інтеграції потрібно мати доступ до джерела даних (API, фізичний порт пристрою, акаунт у системі обліку) та проміжне програмне забезпечення, яке буде зчитувати ці дані і передавати їх боту. У багатьох випадках це буде окремий Python-скрипт або серверна служба, яка працює паралельно з ботом. Telegram-бот у цьому ланцюгу виконує функцію

комунікації з користувачем — запит, вивід інформації, підтвердження, історія тощо.

Застосування Telegram-ботів для автоматизації обліку комунальних послуг, зокрема електроенергії, стає все більш популярним завдяки зручності та можливості інтеграції з різними системами. Продовжимо детальніше розглядати можливості, які можуть бути реалізовані для такої автоматизації, а також потенційні проблеми і виклики.

Інтеграція з «розумними» лічильниками (Smart Meters)

Як було зазначено, розумні лічильники з вбудованими модулями зв'язку дозволяють отримувати дані без необхідності ручного введення показань. Це може значно зекономити час і знизити ймовірність помилок при введенні інформації. Такі лічильники можуть передавати дані через різні протоколи, що дає можливість для автоматичного моніторингу споживання електроенергії.

API підключення - найбільш зручним способом інтеграції є використання відкритих API або протоколів, які надає виробник лічильника або енергопостачальна компанія. Завдяки цьому можна здійснювати автоматичний збір показань без участі користувача. Наприклад, лічильники можуть передавати дані через MQTT або DLMS протоколи, що дозволяє створювати систему, яка збиратиме ці показання безпосередньо через бот.

Проблеми та рішення - інколи можуть виникнути проблеми з налаштуванням зв'язку або синхронізацією між лічильником і Telegram-ботом. У таких випадках необхідно забезпечити належну технічну підтримку та моніторинг для виявлення помилок у з'єднанні або передачі даних.

Інтеграція з особистими кабінетами енергопостачальних компаній

Багато енергетичних компаній пропонують онлайн-сервіси, де користувачі можуть переглядати свої показники споживання електроенергії. Такі сервіси часто мають зручний інтерфейс для перевірки нарахувань, оплати та історії споживання. Для Telegram-бота можна організувати інтеграцію з цими сервісами через API або через автоматизоване отримання даних.

API від енергопостачальника - якщо компанія надає API для отримання даних про споживання електроенергії, інтеграція з ботом буде достатньо простою. Наприклад, за допомогою бібліотек Python, таких як requests, можна періодично або за запитом користувача отримувати інформацію про споживання електроенергії і на основі цих даних здійснювати розрахунок заборгованості.

Парсинг веб-сторінок (Web Scraping) - якщо API немає, можна здійснювати парсинг веб-сторінок особистих кабінетів. Проте цей підхід має свої недоліки: сторінки можуть змінюватися, що потребує регулярного оновлення коду, а також парсинг може бути заблокований, якщо компанія має заходи захисту від автоматизованого доступу. Це менш стабільний підхід порівняно з використанням API, але все ж може бути корисним варіантом.

Завантаження файлів для обліку показань

Ще один варіант автоматизації збору даних — це завантаження файлів, що містять дані про споживання, наприклад, у форматах CSV, Excel або JSON. Деякі енергетичні компанії надають можливість експорту даних у таких форматах. Бот може бути налаштований для автоматичного завантаження та аналізу цих файлів.

Імпорт даних з файлів - за допомогою Python-бібліотек, таких як pandas або openpyxl, можна організувати процес обробки файлів. Бот може отримувати ці файли через повідомлення в чаті або за допомогою API, що дозволяє автоматизувати введення даних.

Зручність для користувачів - такий варіант є зручним для користувачів, яким не хочеться вручну вводити дані, але вони все ж мають змогу контролювати процес, завантажуючи файл, який містить всі необхідні показники.

Розширення функціоналу Telegram-бота для комунальних послуг

Що стосується додаткових можливостей для Telegram-бота, можна реалізувати кілька корисних функцій:

Нагадування про оплату - бот може надсилати користувачам автоматичні нагадування про необхідність сплатити за спожиту електроенергію. Це можна налаштувати через відкладене повідомлення, яке буде надсилатись користувачу в конкретний день кожного місяця або після досягнення певного порогу боргу.

Історія платежів та споживання - бот може зберігати історію платежів і даних про споживання. Користувачі зможуть переглядати не лише поточні показники, а й аналізувати зміни в споживанні за різні періоди.

Пошук оптимальних тарифів - бот може також надавати рекомендації щодо оптимальних тарифів для користувача на основі його споживання електроенергії. Це може бути корисним, якщо енергетична компанія пропонує різні варіанти тарифів, які варто порівняти.

Підтримка багатьох послуг - замість того, щоб обмежуватися лише електроенергією, бот може бути налаштований для обслуговування і інших комунальних послуг, таких як газ, вода або опалення. Це дозволить створити універсальний інструмент для обліку всіх побутових платежів.

Проблеми безпеки та конфіденційності

При обробці персональних даних користувачів, таких як номер телефону, показники споживання, інформація про платежі, необхідно враховувати вимоги до безпеки та конфіденційності. Telegram надає можливість шифрування повідомлень, але важливо також правильно налаштувати базу даних, щоб уникнути витоків інформації.

Шифрування бази даних - для забезпечення безпеки можна використовувати шифрування бази даних, де зберігаються всі показники та дані користувачів. Це допоможе уникнути несанкціонованого доступу до чутливої інформації.

Підтвердження операцій - для важливих операцій, таких як платіж, можна додавати додаткові підтвердження через код або через двофакторну автентифікацію, що підвищить рівень захисту.

Telegram-бот для обліку електроенергії та інших комунальних послуг є потужним інструментом для спрощення процесів моніторингу та оплати. Інтеграція з різними системами збору даних, автоматизація розрахунків та додавання нових функцій дозволяє зробити таке рішення універсальним і зручним для користувачів. Водночас важливо враховувати можливі технічні та безпекові проблеми, що можуть виникнути при інтеграції з реальними системами обліку та обробки даних.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТУ

Розроблений чат-бот для передачі показань лічильника електроенергії орієнтований на вирішення однієї з повсякденних побутових потреб населення - швидке, зручне та безпомилкове надсилання даних до енергопостачальної компанії. Бот має простий і зрозумілий інтерфейс, що не потребує додаткових установок чи реєстрацій, працює на платформі Telegram, що робить його доступним для широкої аудиторії користувачів, зокрема людей старшого віку.

Цільова аудиторія бота - домогосподарства, які передають показання лічильників щомісяця, особливо мешканці багатоквартирних будинків, приватного сектору, а також літні люди, яким важлива простота користування та зручність доступу. Потенційними користувачами також можуть стати керуючі компанії або ОСББ, які можуть централізовано передавати дані за кількох споживачів.

У перспективі розвитку можна передбачити розширення функціоналу - інтеграцію з іншими сервісами ЖКГ (газ, вода), можливість надсилання повідомлень про тарифи або зміни умов оплати, а також підключення функцій оплати рахунків безпосередньо через бот. Введення особистого кабінету з архівом показань та рахунків також зробить бот ще кориснішим. Проєкт має потенціал для комерційної реалізації через партнерство з енергопостачальними компаніями, впровадження брендованих рішень для кожного регіону або навіть національного масштабу. Такий бот може стати надійним та звичним інструментом цифрової комунікації між споживачами і постачальниками послуг.

У порівнянні з аналогічними рішеннями, що існують на ринку, розроблений чат-бот вирізняється поєднанням простоти, функціональності та зручності. Наприклад, деякі українські енергокомпанії, як-от Yasno (ДТЕК), мають власні боти у Viber або Telegram, однак функціональність часто обмежується лише прийомом показань без збереження історії чи розрахунку боргу. Закордонні приклади, як-от британський бот від Octopus Energy або німецькі рішення в межах smart home-інтеграцій, зазвичай орієнтовані на

технологічно підготовленого користувача, пропонуючи складніші інтерфейси або обов’язкову прив’язку до особистого кабінету.

Перевагою представленого рішення є інтуїтивний сценарій взаємодії, який не вимагає авторизації чи складних налаштувань. Запам’ятовування попередніх показань, автоматичний підрахунок боргу, а також можливість сплати через Google Pay - це функціональність, що зазвичай відсутня у більшості аналогів. Це робить бот особливо привабливим для літніх користувачів та домогосподарств, які прагнуть мінімізувати час на комунальні операції.

Таким чином, розроблений чат-бот є конкурентоспроможним рішенням, яке об’єднує доступність, функціональність та потенціал масштабування. У перспективі він може стати зразком для подібних ініціатив як в Україні, так і за її межами.

Окрім згаданого Yasno, в Україні існують інші приклади чат-ботів для передачі показань лічильників, однак більшість із них мають обмежену функціональність або потребують попередньої реєстрації в особистому кабінеті. Наприклад:

My Kyivstar Бот - хоч і не є спеціалізованим рішенням для ЖКГ, але через нього можна передавати показання за деякими сервісами, інтегрованими з оператором. Проте інтерфейс орієнтований більше на просунутих користувачів, і він не забезпечує збереження історії чи розрахунків безпосередньо в боті.

Чат-бот “Мій Миколаївгаз” у Viber - дозволяє передати показання газового лічильника, однак обов’язково вимагає номер особового рахунку та верифікацію даних, що ускладнює використання для людей літнього віку або тих, хто не зберігає такі дані під рукою.

Єдиний комунальний кабінет у Дніпрі - має Telegram-бота, який інтегрований із міським сервісом. Хоча він дозволяє передавати показання води, газу, світла, функціонал часто обмежений лише відправленням значень, без додаткової аналітики чи спрощеного введення.

Vodokanal.bot (наприклад, у Запоріжжі чи Львові) - орієнтований виключно на передачу показань за воду. Він зручний для базових операцій, але не забезпечує інтеграцію з іншими послугами чи можливість онлайн-оплати.

На фоні цих прикладів розроблений чат-бот вирізняється гнучкістю та дружнім підходом до користувача. Його ключова перевага — універсальність та адаптивність, що дозволяє не лише передавати показання, а й поступово створювати комплексний цифровий інструмент для щомісячних комунальних операцій. Такий підхід вигідно позиціонує проєкт у ніші «розумного» управління побутом, орієнтованого на зручність і доступність для всіх категорій населення.

Варто акцентувати увагу на стратегічних можливостях масштабування та впровадження додаткових сервісів. Зокрема, інтеграція чат-бота з державними або муніципальними платформами (наприклад, «Дія», «Єдиний державний вебпортал електронних послуг») відкриє нові горизонти співпраці та дозволить створити єдиний цифровий простір для взаємодії з постачальниками житлово-комунальних послуг.

У майбутньому бот може перетворитися на персонального асистента користувача з питань ЖКГ, який не лише фіксує дані, а й попереджає про настання термінів оплати, аналізує динаміку споживання ресурсів, радить способи зменшення витрат, або навіть рекомендує тарифи та послуги на основі реального споживчого профілю.

Окрім цього, інтеграція з системами розумного дому (IoT) дозволить передавати показання автоматично, без втручання людини, що особливо корисно для зайнятих користувачів або управителів нерухомості. А запровадження функцій багатомовної підтримки (українська, англійська, російська, польська тощо) зробить бот доступним не лише для громадян України, а й для іноземців, які орендують житло або ведуть бізнес на території країни.

Ще одним важливим напрямком розвитку є аналітика даних споживання в розрізі регіонів або типів житла — це може бути цікаво як для користувачів,

так і для постачальників послуг або органів місцевої влади з метою оптимізації витрат, прогнозування навантажень на мережу тощо.

Таким чином, почавши як простий засіб для передачі показань, бот має усі шанси еволюціонувати у потужний, інтегрований інструмент цифрової взаємодії між мешканцем і державою в контексті житлово-комунальних послуг.

Розвиток чат-бота для передачі показань лічильників має значний потенціал для трансформації у повноцінну цифрову платформу управління житлово-комунальними послугами. Одним із перспективних напрямів є інтеграція з банківськими та фінтех-сервісами, що дозволить реалізувати миттєву оплату рахунків безпосередньо через бот за допомогою Google Pay, Apple Pay або онлайн-банкінгу. Це не лише підвищить зручність для користувачів, а й зменшить кількість прострочених платежів. Крім того, бот може перетворитися на єдину точку доступу до всіх ЖКГ-послуг, де користувачі зможуть переглядати історію переданих показань, отримувати нагадування про оплату, контролювати витрати та керувати кількома об'єктами одночасно. Для юридичних осіб, ОСББ та керуючих компаній бот стане зручним інструментом централізованого збору даних, формування звітів і моніторингу сплати платежів, а також каналом оперативної комунікації з мешканцями. Завдяки накопиченим даним споживання можна реалізувати розширену аналітику - бот зможе попереджати про аномально високе споживання, прогнозувати витрати, формувати персоналізовані рекомендації щодо енергозбереження. У контексті екологічних ініціатив бот також може інформувати користувачів про енергоефективні програми, допомагати розрахувати «вуглецевий слід» домогосподарства та стимулювати зменшення споживання через рейтинги чи гейміфікацію. Як доповнення до Telegram-бота можливий запуск мобільного застосунку з розширеним інтерфейсом, офлайн-доступом до історії показань та push-нотифікаціями. Ще одним стратегічним напрямом є партнерство з органами місцевого самоврядування та державними структурами. Інтеграція з платформами на кшталт «Дія» чи муніципальними

системами дозволить створити єдиний цифровий простір для взаємодії громадян з постачальниками послуг. Таким чином, бот може еволюціонувати з простого сервісу в багатофункціонального персонального асистента з питань ЖКГ, який допомагає не лише передавати дані та оплачувати рахунки, а й аналізує споживання, попереджає про ризики, оптимізує витрати і забезпечує прозорість у відносинах між мешканцем і державою.

Знайти його можна за посиланням: t.me/meter_d_bot.

ВИСНОВКИ

У ході розробки чат-бота для передачі показань лічильників електроенергії було створено зручний, доступний та ефективний інструмент для користувачів, що дозволяє швидко і безпомилково передавати показання до енергопостачальної компанії. Система реалізована на платформі Telegram, що робить її доступною для широкого кола користувачів, зокрема для людей старшого віку. Основною мовою розробки став Python, а для інтеграції з різними сервісами та базами даних використовувалися відповідні бібліотеки та API. Особливу увагу було приділено інтуїтивно зрозумілому інтерфейсу, який не вимагає додаткової реєстрації чи складних налаштувань.

У теоретичній частині роботи було проведено аналіз існуючих рішень на ринку чат-ботів для передачі показань лічильників та інших комунальних послуг. Розглянуто переваги та недоліки доступних сервісів, а також визначено ключові функціональні вимоги до чат-бота, серед яких – простота використання, можливість автоматичного збереження історії показань, підрахунок боргу та оплата рахунків через інтегровані платіжні системи. Також було обґрунтовано вибір інструментів розробки, зокрема використання Telegram як основної платформи, що забезпечує простоту доступу та зручність для користувачів.

У процесі проєктування було сформульовано основні функціональні та нефункціональні вимоги до системи, а також спроєктовано архітектуру чат-бота, алгоритми обробки запитів та взаємодії з користувачем. Реалізовано основні функції, зокрема: введення та збереження показань, автоматичний підрахунок боргу, інтеграція з платіжними системами для сплати рахунків через Google Pay, а також надання сповіщень про зміни тарифів чи нові повідомлення від енергопостачальника. Особливістю розробленого рішення є його адаптивність та гнучкість, що дозволяє з часом додавати нові функції та інтегрувати додаткові сервіси, такі як передача показань за газ чи воду.

Практичне тестування бота здійснювалось за участі користувачів різного віку та технічної підготовки. Тестування підтвердило, що система

ефективно виконує всі необхідні функції, є зручною для використання та забезпечує високий рівень точності передачі показань. Користувачі відзначили зручність простоти інтерфейсу, можливість швидко отримати підтвердження успішної передачі показань і сплати рахунків, а також корисність автоматичних сповіщень.

Порівняння розробленого бота з існуючими рішеннями виявило ряд переваг, зокрема: відсутність необхідності в реєстрації, збереження історії показань, автоматичний підрахунок боргу, а також інтеграція з платіжними системами. Одним із головних недоліків є необхідність стабільного інтернет-з'єднання для коректної роботи бота та можливі обмеження у функціональності при інтеграції з іншими сервісами ЖКГ.

Таким чином, виконана робота охоплює повний цикл створення чат-бота: від ідеї та проєктування до реалізації, тестування та аналізу ефективності. Розроблений чат-бот є прикладом використання сучасних технологій для вирішення повсякденних побутових проблем, зокрема спрощення передачі показань лічильників, оптимізації процесу сплати комунальних послуг і покращення взаємодії між споживачами та енергопостачальними компаніями. У перспективі проект може бути доповнений новими функціями, такими як інтеграція з іншими комунальними сервісами (газ, вода), додаванням персональних кабінетів для користувачів та можливістю автоматичного зчитування показань за допомогою пристроїв IoT.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Міністерство енергетики України. У Міністерстві енергетики проаналізували кількість та стан лічильників в ЖКГ. UA Energy. URL: https://ua-energy.org/uk/posts/u-minenerhetyky-proanalizuvaly-kilkist-ta-stan-lichylnykiv-v-zhkh?utm_source=chatgpt.com (дата звернення: 22.04.2025).

2. Волинські новини. Волинь першою в Україні може запровадити SMART-лічильники. Волинь News. URL: https://www.volynnews.com/news/all/volyn-pershoiu-v-ukrayini-mozhe-zaprovadyty-SMART-lichylnyky/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

3. PAUCI. Унікальний проєкт для підвищення ефективності обліку енергоресурсів в Україні. PAUCI. URL: https://pauci.org/ua/news/324?utm_source=chatgpt.com (дата звернення: 22.04.2025).

4. Elmisto. Сама інформація про лічильники енергоресурсів для автоматизованого обліку. Elmisto. URL: https://elmisto.com.ua/ua/a428460-sama-informatsiya-pro.html?utm_source=chatgpt.com (дата звернення: 22.04.2025).

5. 24 канал. Показники лічильників світла: коли українцям передавати дані. 24 TV. URL: https://24tv.ua/economy/pokazniki-lichilnikiv-svitla-koli-ukrayintsyam-peredavati-dani_n2436570?utm_source=chatgpt.com (дата звернення: 22.04.2025).

6. Черкаська хімволокно. Відтепер передати показники лічильників можливо за допомогою Telegram-Bot! Черкаська хімволокно. URL: https://khimvolokno.com.ua/news-post/vidteper-peredaty-pokaznyky-lichylnykiv-mozhlyvo-za-dopomogoyu-telegram-bot.html?utm_source=chatgpt.com (дата звернення: 22.04.2025).

7. DTEK. DTEK Grids запустив чат-бот для 12 млн жителів України. DTEK. URL: https://dtek.com/en/media-center/news/dtek-seti-zapustil-chat-bot-

dlya-12-mln-zhiteley-ukrainy/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

8. GitHub. sl4wa/poweron: Telegram bot for power outages in Lviv, Ukraine. GitHub. URL: https://github.com/sl4wa/poweron?utm_source=chatgpt.com (дата звернення: 22.04.2025).

9. GitHub. DimChig/SvitloBot: Serving 135,000+ users in 1,100+ households. GitHub. URL: https://github.com/DimChig/SvitloBot?utm_source=chatgpt.com (дата звернення: 22.04.2025).

10. YASNO. Офіційний сайт YASNO. URL: https://yasno.com.ua/news/b2b_news/telegram_bot?utm_source=chatgpt.com (дата звернення: 22.04.2025).

11. Чернігівобленерго. Telegram. Чернігівобленерго. URL: https://chernihivoblenergo.com.ua/in_telegram?utm_source=chatgpt.com (дата звернення: 22.04.2025).

12. Українська енергетична біржа. Українська енергетична біржа запустила Telegram-бот. Українська енергетична біржа. URL: https://www.ueex.com.ua/eng/presscenter/news/ukrainian-energy-exchange-launches-telegram-bot/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

13. DTEK. DTEK Grids запустив чат-бот для 12 млн жителів України. DTEK. URL: https://dtek.com/en/media-center/news/dtek-seti-zapustil-chat-bot-dlya-12-mln-zhiteley-ukrainy/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

14. Interfax Ukraine. DTEK планує реалізувати проект смарт-мереж в Київській області на суму 145 млн доларів США. Interfax Ukraine. URL: https://en.interfax.com.ua/news/economic/918464.html?utm_source=chatgpt.com (дата звернення: 22.04.2025).

15. Interfax Ukraine. DTEK представляє концепцію розвитку смарт-мереж на прикладі Київської області. Interfax Ukraine. URL:

https://en.interfax.com.ua/news/economic/926295.html?utm_source=chatgpt.com (дата звернення: 22.04.2025).

16. DTEK. Передати показники лічильника легко та швидко можна через чат-бот DTEK Дніпровські електромережі. DTEK. URL: https://grids.dtek.com/en/media-center/press/peredati-pokazi-lichilnika-legko-ta-shvidko-mozhna-cherez-chat-bot-dtek-dniprovski-elektromerez/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

17. DTEK. Обережно фейки! DTEK Київські регіональні електромережі попереджає про TG канали з неправдивою інформацією про відключення світла. DTEK. URL: https://grids.dtek.com/en/media-center/press/oberezhno-feyki-dtek-kiivski-regionalni-elektromerezhi-poperedzhae-pro-tg-kanali-z-nepravdivoyu-informatsieyu-pro-vidklyuchennya-svitl/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

18. EBA. Українські газовики створили власні смарт-лічильники: кому і коли їх встановлюватимуть. European Business Association. URL: https://eba.com.ua/en/ukrayinski-gazovyky-stvoryly-vlasni-smart-lichylnyky-komu-i-koly-yih-vstanovlyuvatymut/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

19. DTEK. 70% клієнтів DTEK Київські регіональні електромережі надають перевагу чат-боту й особистому кабінету. DTEK. URL: https://grids.dtek.com/en/media-center/press/70-klientiv-dtek-kiivski-regionalni-elektromerezhi-nadayut-perevagu-chat-botu-y-osobistomu-kabin/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

20. New York Post. "The Ultimate AI, ChatGPT & Python Programming Bundle Is 80% Off." New York Post. URL: https://nypost.com/2025/01/03/shopping/the-ultimate-ai-chatgpt-amp-python-programming-bundle-is-80-off/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

21. The Guardian. "I learned the language of computer programming in my 50s – here's what I discovered." The Guardian. URL:

https://www.theguardian.com/technology/article/2024/aug/31/learning-computer-programming-language-coding-devil-stack-andrew-smith?utm_source=chatgpt.com (дата звернення: 22.04.2025).

22. W3Schools. "Python Tutorial." W3Schools. URL: https://www.w3schools.com/PYTHON/default.asp?utm_source=chatgpt.com (дата звернення: 22.04.2025).

23. GeeksforGeeks. "Python Programming Language Tutorial." GeeksforGeeks. URL: https://www.geeksforgeeks.org/python-programming-language-tutorial/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

24. Programiz. "Python Programming Guide." Programiz. URL: https://www.programiz.com/python-programming/guide?utm_source=chatgpt.com (дата звернення: 22.04.2025).

25. PYnative. "Python Basics." PYnative. URL: https://pynative.com/python/basics/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

26. W3Schools. "Python Introduction." W3Schools. URL: https://www.w3schools.com/python/python_intro.asp?utm_source=chatgpt.com (дата звернення: 22.04.2025).

27. W3Schools. "SQL Tutorial". URL: https://www.w3schools.com/sql/ (дата звернення: 22.04.2025).

28. GeeksforGeeks. "SQL Tutorial". URL: https://www.geeksforgeeks.org/sql-tutorial/ (дата звернення: 22.04.2025).

29. TutorialsPoint. "SQL Database Tutorial". URL: https://www.tutorialspoint.com/sql/index.htm (дата звернення: 22.04.2025).

30. SQLZoo. "SQL Tutorials". URL: https://sqlzoo.net/ (дата звернення: 22.04.2025).

31. Database Journal. "SQL Queries and Database Optimization". URL: https://www.databasejournal.com/ (дата звернення: 22.04.2025).

32. Khan Academy. "Intro to SQL: Querying and managing databases". URL: https://www.khanacademy.org/computing/computer-programming/sql (дата звернення: 22.04.2025).

33. Stack Overflow. "SQL Query Examples". URL: https://stackoverflow.com/questions/tagged/sql (дата звернення: 22.04.2025).

34. IBM. "SQL Performance Optimization". URL: https://www.ibm.com/docs/en/db2 (дата звернення: 22.04.2025).

35. Microsoft Docs. "SQL Server Documentation". URL: https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15 (дата звернення: 22.04.2025).

36. PostgreSQL Documentation. "PostgreSQL: Documentation". URL: https://www.postgresql.org/docs/ (дата звернення: 22.04.2025).

37. Інтерфакс-Україна. "Україна готова до інтеграції в європейську енергосистему – Міністр енергетики". URL: https://en.interfax.com.ua/news/economic/918464.html?utm_source=chatgpt.com (дата звернення: 22.04.2025).

38. ДТЕК. "ДТЕК Запустив новий сервіс для зручного обліку споживання електроенергії". URL: https://dtek.com/en/media-center/news/dtek-seti-zapustil-chat-bot-dlya-12-mln-zhiteley-ukrainy/?utm_source=chatgpt.com (дата звернення: 22.04.2025).

39. Ukrenergo. "Стан енергетичної системи України та події останніх тижнів". URL: https://ua.energy/uk/posts/stan-enerhetychnoyi-systemy-ukrayiny (дата звернення: 22.04.2025).

40. Енергоатом. "Ядерна енергетика України: стан та майбутнє". URL: https://www.energoatom.com.ua/ua/news/2019/12/17/ (дата звернення: 22.04.2025).

41. НКРЕКП. "Регулювання енергетики в Україні: законодавчі зміни". URL: https://www.nerc.gov.ua/ (дата звернення: 22.04.2025).