**ЗМІСТ**

[ВСТУП 2](#_Toc137206964)

[1 ЕЛЕКТРИЧНА КОМПАНІЯ ДТЕК В УКРАЇНІ 4](#_Toc137206965)

[1.1 Загальні відомості 4](#_Toc137206966)

[1.2 Сучасний стан компанії 5](#_Toc137206967)

[2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ 8](#_Toc137206968)

[2.1 Сайт ДТЕК 8](#_Toc137206969)

[2.2 Телеграм-бот ДТЕК 13](#_Toc137206970)

[2.3 Гаряча лінія постачальника електроенергії 21](#_Toc137206971)

[2.4 Інформаційні джерела 23](#_Toc137206972)

[3 ВИБІР ТЕХНОЛОГІЧНОГО РІШЕННЯ 26](#_Toc137206973)

[3.1 Месенджер Telegram як фронтенд платформа 26](#_Toc137206974)

[3.2 Мова програмування Python 28](#_Toc137206975)

[3.3 Система контролю версіями Git 31](#_Toc137206976)

[3.4 Реляційні бази даних 33](#_Toc137206977)

[3.5 Документно-орієнтовані бази даних 35](#_Toc137206978)

[3.6 Система управління базами даних MongoDB 38](#_Toc137206979)

[4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 41](#_Toc137206980)

[4.1 Технічне завдання на розробку програмного забезпечення 41](#_Toc137206981)

[4.2 Проектування інтерфейсу користувача 44](#_Toc137206982)

[4.3 Алгоритм рішення задач 56](#_Toc137206983)

[ВИСНОВКИ 68](#_Toc137206984)

[ПЕРЕЛІК ПОСИЛАНЬ 69](#_Toc137206985)

# ВСТУП

Відключення електроенергії є поширеним явищем у багатьох регіонах України, що спричиняє незручності, порушення повсякденної діяльності та потенційні ризики для різних верств населення. Метою роботи є вирішення цих проблем та пом'якшення негативних наслідків відключень електроенергії для міст, що охоплюють різні аспекти соціальної, економічної, інфраструктурної та громадської безпеки.

З соціальної точки зору, перебої в електропостачанні порушують системи зв'язку, ізолюючи людей від екстрених служб, сім'ї та друзів. Крім того, втрата електроенергії ускладнює доступ до соціальних мереж і розважальних платформ, посилюючи соціальну ізоляцію. Відсутність систем освітлення, камер спостереження та сигналізації під час відключень також може становити загрозу безпеці, особливо вночі.

З економічної точки зору, відключення електроенергії призводить до значних збитків для бізнесу та промисловості. Виробничі процеси перериваються, що призводить до затримок, пошкодження обладнання та фінансових втрат. Бізнес, орієнтований на надання послуг, включаючи ресторани, готелі та роздрібні магазини, відчуває зниження доходів через закриття та зменшення потоку клієнтів. Транспортні системи також стикаються з перебоями, що впливають на ланцюги поставок і спричиняють логістичні проблеми.

Інфраструктура значною мірою залежить від електроенергії, і відключення електроенергії може мати серйозні наслідки. Критично важливі системи, такі як транспортні мережі, заклади охорони здоров'я та водопостачання, залежать від безперебійного живлення. Перебої в електропостачанні порушують роботу світлофорів, метро та залізничних мереж, спричиняючи затори, затримки та загрозу безпеці. Лікарні потребують безперервного живлення для життєво важливого обладнання та догляду за пацієнтами, що робить перебої в електропостачанні небезпечними для життя. Водоочисні станції, насосні станції та очисні споруди борються за забезпечення чистої води та підтримання санітарних умов.

Для вирішення цих проблем було запропоновано розробити додаток. Метою цього додатку є завчасне та зручне сповіщення громадян про відключення електроенергії, тим самим покращуючи готовність та реагування на такі події. Використовуючи сучасні технології, додаток може надавати сповіщення та оновлення в режимі реального часу, що дозволить користувачам вжити необхідних заходів обережності та мінімізувати вплив відключень електроенергії на їхнє повсякденне життя.

Використовуючи платформи обміну миттєвими повідомленнями, додаток може охопити широке коло користувачів, включаючи мешканців, підприємства і навіть постачальників комунальних послуг.

# 1 ЕЛЕКТРИЧНА КОМПАНІЯ ДТЕК В УКРАЇНІ

## 1.1 Загальні відомості

ДТЕК – провідна енергетична компанія України, визнана за свою значну роль в енергетичному секторі країни. Заснований у 2005 році, ДТЕК перетворився на провідний вертикально інтегрований енергетичний холдинг, що об'єднує низку дочірніх підприємств і працює в усіх ланках енергетичного ланцюжка.

Основною метою ДТЕК є забезпечення надійного та сталого енергопостачання України. Як вертикально інтегрована компанія, ДТЕК здійснює різні види діяльності, пов'язані з виробництвом, розподілом і постачанням електроенергії, а також розвідкою, видобутком і переробкою вугілля, природного газу та відновлюваних джерел енергії.

Компанія володіє декількома вугільними шахтами, що дозволяє їй забезпечувати стабільне постачання вугілля для виробництва електроенергії. Вугільні активи ДТЕК включають шахти з видобутку енергетичного та коксівного вугілля, стратегічно розташовані в різних регіонах України. Такий диверсифікований вугільний портфель дозволяє компанії задовольняти різні енергетичні потреби і обслуговувати різні сектори, такі як генерація електроенергії, промислове виробництво і теплоенергетика.

ДТЕК також інвестує значні кошти у відновлювані джерела енергії, приділяючи особливу увагу вітровій та сонячній енергетиці. Компанія експлуатує кілька вітрових і сонячних електростанцій по всій Україні, сприяючи переходу країни до більш сталого енергетичного балансу. Завдяки цим інвестиціям ДТЕК прагне зменшити свій вуглецевий слід, підвищити енергоефективність і просувати більш чисті альтернативні джерела енергії.

Окрім вугільної та відновлюваної енергетики, ДТЕК займається видобутком і розподілом природного газу. Компанія займається розвідкою та видобутком запасів природного газу, використовуючи свій досвід і ресурси для зміцнення енергетичної незалежності та безпеки України. Забезпечуючи диверсифікований енергетичний портфель, ДТЕК відіграє важливу роль у зменшенні залежності країни від зовнішніх джерел енергії.

ДТЕК управляє розгалуженою мережею розподілу електроенергії, постачаючи її побутовим, комерційним і промисловим споживачам. Інфраструктура компанії включає лінії електропередач, підстанції та інші розподільчі активи, що забезпечують ефективне та надійне постачання електроенергії по всій території України.

Компанія прагне впроваджувати сучасні технології та кращі практики у своїй діяльності, сприяти підвищенню енергоефективності та екологічної стійкості. Компанія активно впроваджує інноваційні рішення, такі як «розумні мережі» і діджиталізація, для оптимізації виробництва і споживання енергії, зниження втрат і підвищення загальної операційної ефективності.

Як соціально відповідальне підприємство, ДТЕК також фокусується на ініціативах корпоративної соціальної відповідальності, розвитку громад та захисті навколишнього середовища. Компанія сприяє розвитку позитивних відносин з місцевими громадами, реалізуючи проєкти, орієнтовані на розвиток освіти, охорони здоров'я та соціального захисту, мінімізуючи при цьому вплив своєї діяльності на навколишнє середовище.

## 1.2 Сучасний стан компанії

ДТЕК Мережі розвиває бізнес з розподілу електроенергії й експлуатації електромереж в Києві, Київській, Дніпропетровській, Донецькій та Одеській областях. Підприємства обслуговують всього 5,6 млн клієнтів [2].

У 2018 році в рамках реформи ринку електроенергії завершив процедуру анбандлінгу, внаслідок чого на базі «Київенерго», «Дніпрообленерго» та «Донецькобленерго» були створені оператори систем розподілу – «ДТЕК Київські електромережі», «ДТЕК Дніпровські електромережі» і «ДТЕК Донецькі електромережі».

У Новій стратегії до 2030 року ДТЕК зобов'язався трансформувати бізнес в більш екологічний, ефективний і технологічний, який у своїй діяльності керується принципами ESG. Мета ДТЕК – досягти вуглецевої нейтральності до 2040 року.

Також ДТЕК інтегрував 12 Цілей сталого розвитку ООН у свою ESG-стратегію.

Загалом ДТЕК бере участь у наступних ініціативах:

* глобальний договір ООН;
* valuable 500 (міжнародна соціальна ініціатива);
* бізнес без бар'єрів (ініціатива Першої Леді);
* women's Empowerment Principles.

В 2018 році була створена дирекція з інновацій – Innovation DTEK, а з початку 2019 року в ДТЕК почала працювати програма цифрової трансформації компанії під назвою MODUS.

З самого початку російського вторгнення Група ДТЕК разом з НАК «Нафтогаз України» та НЕК «Укренерго» ініціювали проєкт Stop Bloody Energy, в рамках якого вони закликають західні компанії припинити співпрацю з Росією у сфері енергетики. За допомогою цієї ініціативи українські компанії хочуть показати міжнародній спільноті, хто з представників світового бізнесу все ще продовжує працювати з Росією, незважаючи на війну в Україні. На сайті проєкту зібрана інформація про газові, вугільні, нафтосервісні та машинобудівельні компанії.

Ініціатива була публічно підтримана Офісом Президента України та Президентом України Володимиром Зеленським, на чиїх сторінках у соціальних мережах були опубліковані відеоролики з посиланням на проєкт.

Проєкт публічно підтримали міністр закордонних справ України Дмитро Кулеба, радник голови Офісу Президента України Михайло Подоляк, посол України в Німеччині Андрій Мельник, а також відомі українські футболісти, телеведучі та артисти.

18 травня фахівці ДТЕК на 100% повернули світло у Київську область, звільнену від російських військових 2 квітня 2022 року. Електроенергія, високовольтними та розподільчими мережами, що належать компанії, була подана у 600 населених пунктів в домівки 260 тисяч родин. Роботи були виконані за 45 днів, а не 60, як було анонсовано компанією раніше. Група ДТЕК інвестувала 300 млн грн. у відновлення зруйнованих війною електричних мереж Київської області. Енергетики ДТЕК щодня відновлюють електропостачання у Донецькій області незважаючи на постійні бойові дії.

З 24 лютого 2022 р. загинув 51 співробітник Групи ДТЕК, було поранено 99 людей. Для вшанувати героїчної роботи працівників енергетичної сфери, які щодня ризикують своїм життям, Група ДТЕК ініціювала проєкт «Герої енергетичного фронту». На сайті проєкту зібрані історії енергетиків, які продовжують героїчно працювати в умовах війни.

Група ДТЕК безкоштовно надає електроенергію установам критичної інфраструктури – державним та комунальним медичним установам, військовим та силовим структурам. За чотири місяці Група ДТЕК надала безкоштовну електроенергію на 160 млн грн близько 100 закладам. Зокрема: Інститут травматології та ортопедії АМН України, Дніпропетровська обласна клінічна лікарня ім. І.І.Мечникова, Дніпропетровський обласний центр екстреної медицини та медицини катастроф, Курахівська міська лікарня тощо.

# 2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

## 2.1 Сайт ДТЕК

Через нещодавні удари по об'єктах енергетичної інфраструктури України виникали аварійні відключення електроенергії, що спонукало владу до швидкого відновлення обладнання та забезпечення стабільного електропостачання. Щоденна потужність виробництва енергії зменшилася, що призвело до необхідності скорочення споживання електроенергії. Для цього було запроваджено віялові відключення електроенергії як захід для зменшення навантаження на енергосистему та запобігання проблемам синхронізації, перебоям у частоті та тривалим перебоям у роботі системи. Віялові відключення передбачають вибіркове відключення споживачів від електромережі почергово, з використанням розподільчих пристроїв для управління навантаженням. Такі відключення є крайнім заходом для підтримки цілісності енергосистеми. Причинами віялових відключень можуть бути несподіваний або нетиповий піковий попит, тимчасова недоступність пікової генерації, недостатня пропускна здатність електростанцій або ліній електропередач.

Враховуючи участь ДТЕК у розподілі електроенергії та експлуатації електромереж у Києві, компанія стала ключовим суб'єктом в управлінні цими перебоями в електропостачанні. Для забезпечення ефективної комунікації та підтримки кінцевих споживачів ДТЕК розробив різні сервіси, в тому числі офіційний вебсайт [2]. Сайт є інформаційним порталом для клієнтів та мешканців міста, де можна знайти відповіді на питання, пов'язані з електромережами, записатися в сервісні центри, оплатити рахунки, укласти договори та зняти показання лічильників. Він є зручним та сучасним інструментом як для приватних осіб, так і для бізнесу.

У відповідь на виклики останнього часу на сайті з'явився спеціальний розділ, який містить актуальну інформацію про стан енергосистеми та графік віялових відключень. У разі виявлення розбіжностей між графіком та фактичним виконанням, користувачі можуть подати заявку або отримати доступ до номера гарячої лінії безпосередньо з цієї ж сторінки, що забезпечить оперативну допомогу та ефективне вирішення будь-яких проблемних питань.

На рис. 2.1 зображено вигляд вкладки з інформацією про відключення.

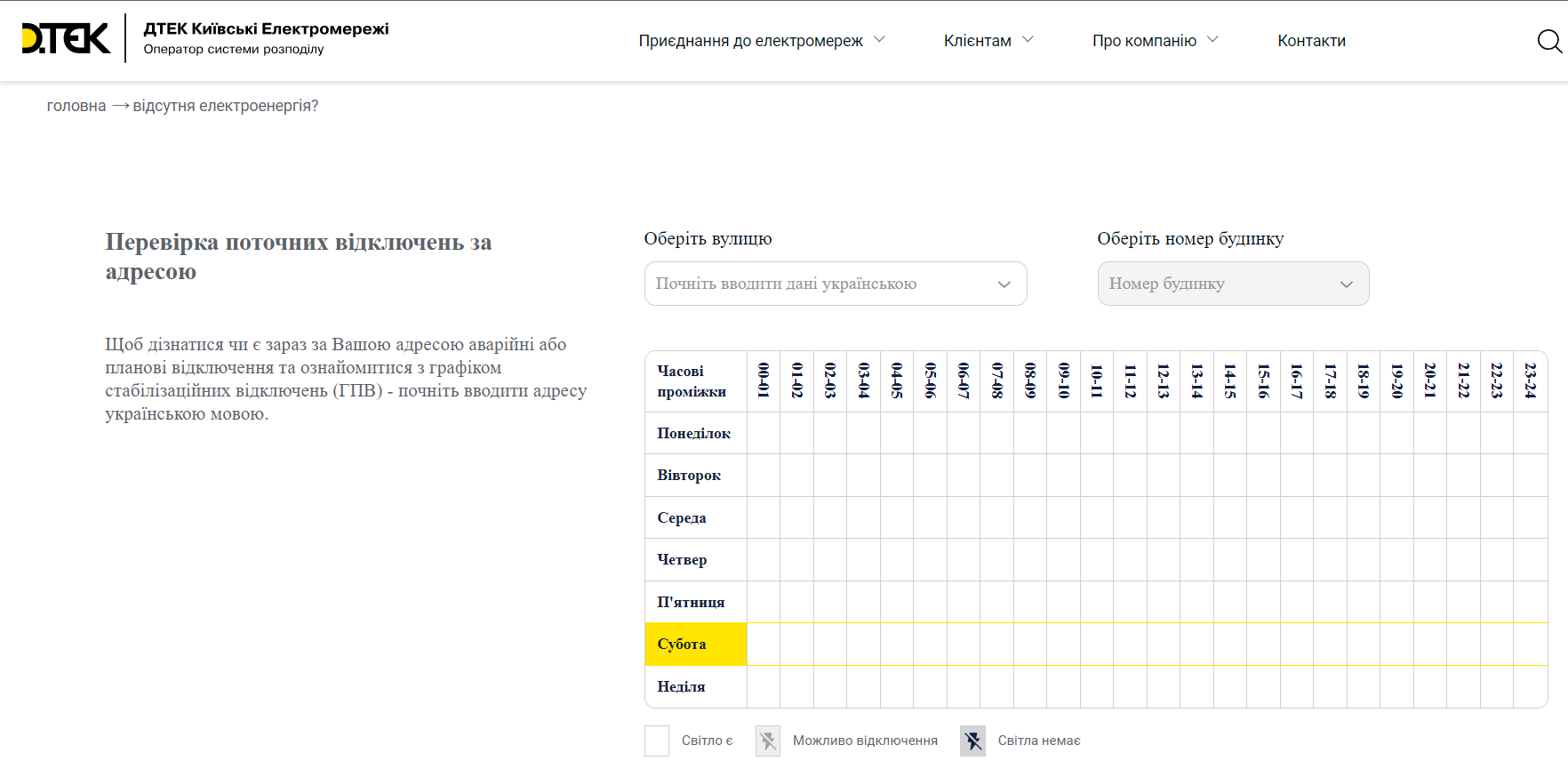


Рисунок 2.1 – Вигляд сторінки інформації про відключення у місті

(https://www.dtek-kem.com.ua/ua/shutdowns)

На сайті зручно представлена таблиця, де кожен рядок відповідає дню тижня, а стовпчики – різним годинам доби. Щоб визначити, чи є аварійні або планові відключення за адресою користувача, йому потрібно ввести свою адресу українською мовою. На рис. 2.2 показано приклад введення адреси вулиці на сайті.

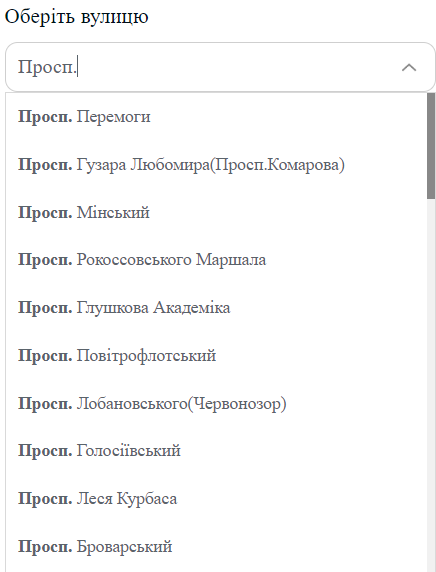


Рисунок 2.2 – Приклад введення вулиці на сайті

(https://www.dtek-kem.com.ua/ua/shutdowns/streets.png)

Варто зазначити, що вебсайт містить зручну функцію, яка підказує можливу адресу, коли користувач починає вводити інформацію. Ця функція є дуже корисною, оскільки економить час, враховуючи час, необхідний для того, щоб щоразу вводити повну адресу будинку. Крім того, введення тексту на мобільних пристроях схильне до помилок, що робить функцію автоматичного підказування ще більш цінною. Назви вулиць не завжди ідеально відповідають офіційним реєстрам, і дані можуть бути застарілими. Покладаючись на єдине сховище, таке як вебсайт, користувачі можуть забезпечити точну ідентифікацію назв своїх вулиць і уникнути будь-яких можливих помилок. Після введення адреси будинку, потрібно увести його номер. Сайт і для номеру будинку також пропонує всі можливі варіанти.

Після введення всіх необхідних даних вигляд сайту оновлюється, та таблиця заповнюється даними відключень на поточний тиждень. На рис. 2.3 зображено вигляд сайту відключень після введення всіх необхідних даних.

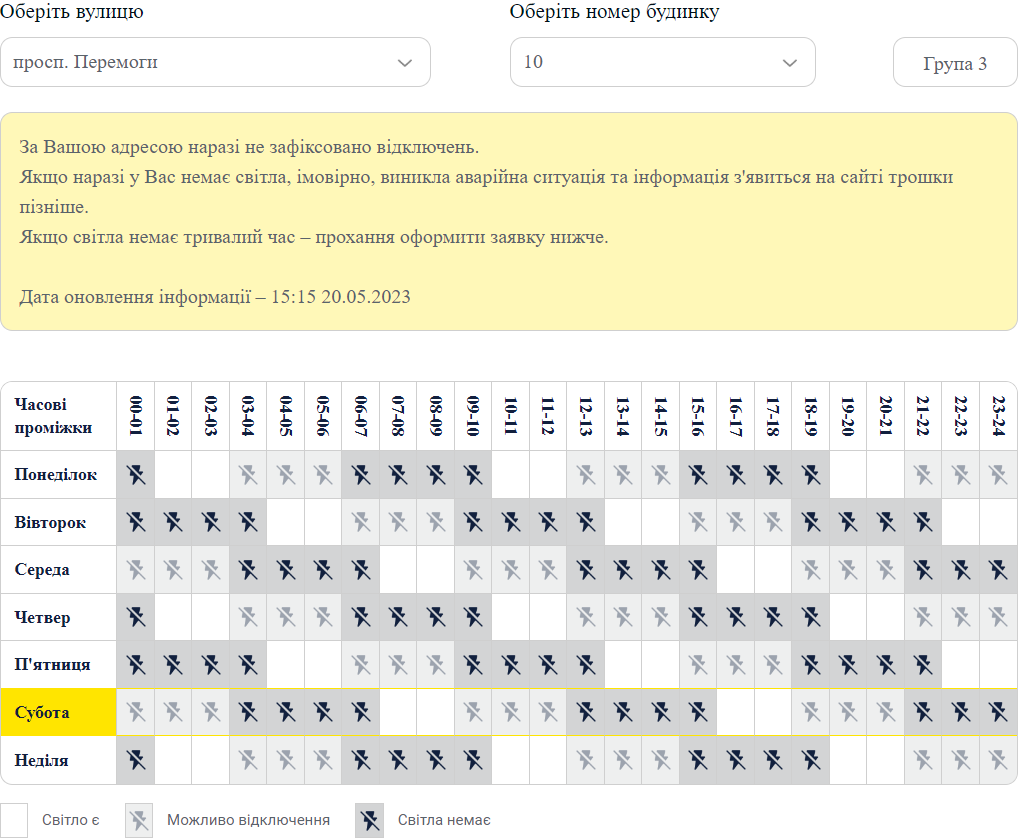


Рисунок 2.3 – Вигляд сайту після вводу всіх даних

(https://www.dtek-kem.com.ua/ua/shutdowns)

Вебсайт про відключення надає повний та актуальний опис відключень електроенергії, стабілізаційних відключень та орієнтовний час відновлення. Дані представлені у добре структурованій таблиці з маркуванням, яка представляє різні «групи», на які поділено місто. Користувачі можуть легко визначити свою групу, звернувшись до верхнього правого кута вебсайту. Кожна будівля у місті належить до певної групи, пронумерованої від 1 до 3.

У таблиці відображається статус відключення та очікуваний час відновлення електропостачання для кожної групи. Вона слугує надійним орієнтиром для користувачів для визначення тривалості потенційних відключень у їхньому районі. Ця інформація регулярно оновлюється енергопостачальником для забезпечення точності та надійності.

На додаток до таблиці, на сайті також представлена поточна інформація, що стосується конкретно обраної будівлі. Користувачі можуть отримати доступ до додаткової інформації про стан відключення їхнього будинку та орієнтовний час відновлення. Ця персоналізована інформація надходить безпосередньо від енергопостачальника, що гарантує її надійність.

Для подальшого підвищення прозорості на сайті вказана дата останнього оновлення інформації. Це гарантує, що користувачі знають про актуальність даних і можуть приймати обґрунтовані рішення на основі найсвіжішої доступної інформації.

Загалом, вебсайт про відключення пропонує зручний інтерфейс, що надає цінну та актуальну інформацію про відключення, стабілізаційні відключення та час відновлення електропостачання для різних груп населення міста. Включення персоналізованих даних про будівлі та чіткі вказівки на надійність даних підвищують ефективність сайту в наданні допомоги користувачам під час перебоїв в електропостачанні.

Вебсайт містить цінну інформацію про перебої в роботі, але має деякі недоліки, які впливають на роботу користувачів:

* повільне або нестабільне завантаження може розчарувати користувачів. Такі фактори, як перевантажені сервери, поганий інтернет-зв'язок та неефективне кодування можуть спричинити затримки;
* вимога повного введення адреси для доступу до розкладу відключень є незручною. Зберігання даних користувачів для автоматичного визначення відповідної інформації спростило б цей процес;
* користувачі повинні запам'ятовувати конкретні дати відключень на весь тиждень, що може бути складно. Впровадження послуги завчасного нагадування дозволить надсилати сповіщення на мобільні пристрої користувачів, щоб вони були готові до відключень та надзвичайних ситуацій.

## 2.2 Телеграм-бот ДТЕК

Компанія ДТЕК надає сервіс інформування клієнтів та мешканців міста завдяки Телеграму, розробивши в цій екосистемі бота. На рис. 2.4 зображено зовнішній вигляд бота ДТЕК.

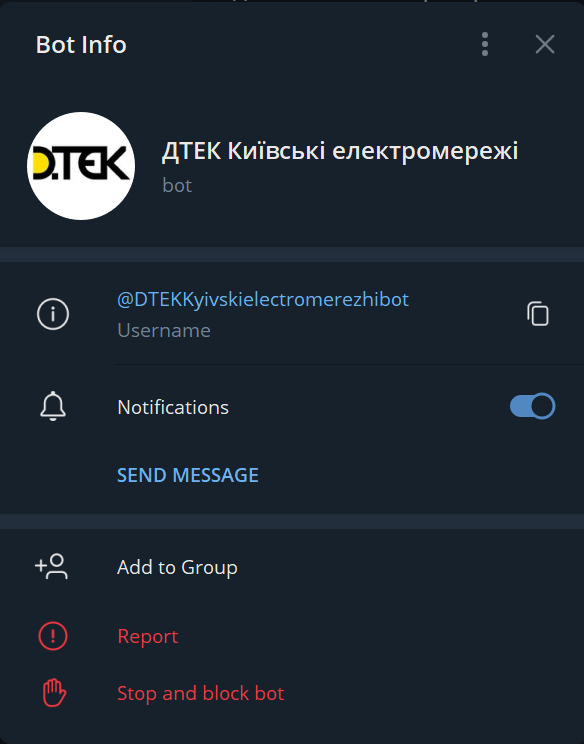


Рисунок 2.4 – Зовнішній вигляд бота ДТЕК

Ідентифікаційне ім'я бота, розробленого в екосистемі Telegram, – «DTEKKyivskielectromerezhibot». Користувачі можуть отримати прямий доступ до бота за посиланням з мобільних пристроїв або персональних комп'ютерів [3].

На рис. 2.5 зображено стартове вікно початку роботи з сервісом.

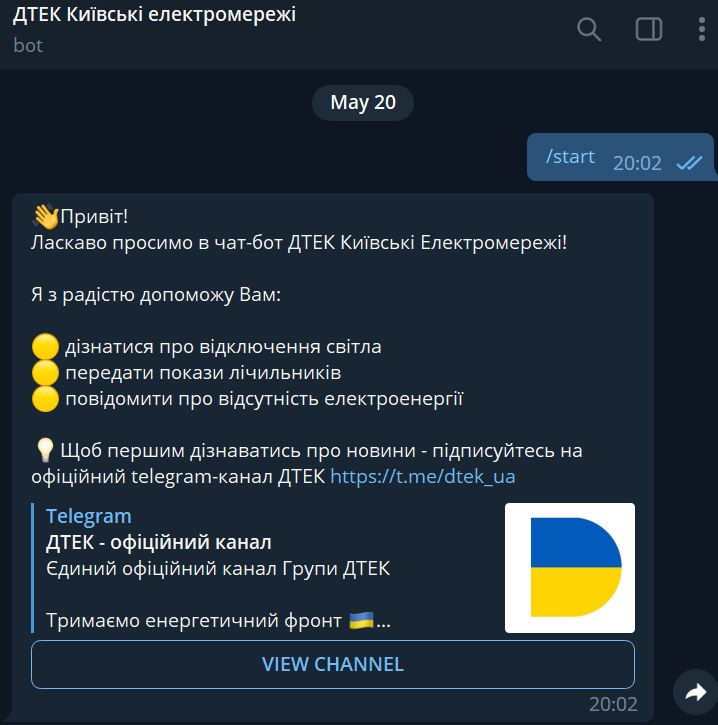


Рисунок 2.5 – Стартове вікно сервісу

Після ініціалізації чату командою /start, бот вітає користувача, надає інформацію про свої можливості, та надає посилання на офіційний чат ДТЕК.

На рис. 2.6 зображено результат команди «Почати».

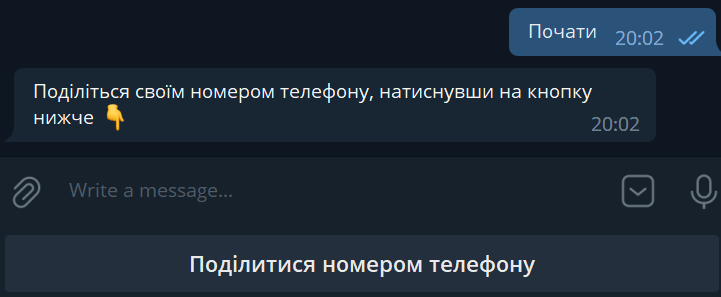


Рисунок 2.6 – Результат команди «Почати»

Для початку роботи з ботом необхідно поділитися номером, прив’язаним до аккаунту користувача.

На рис. 2.7 зображено результат команди «Поділитися номером телефону».

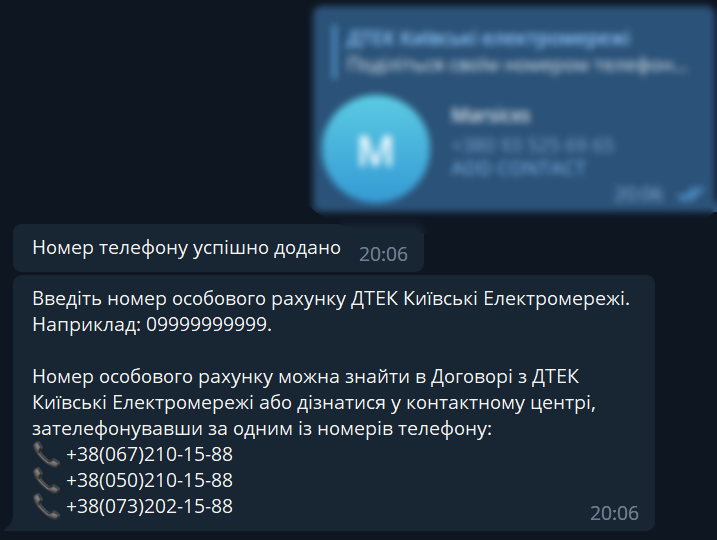


Рисунок 2.7 – Результат команди «Поділитися номером телефону»

Після виконання команді «Поділитися номером телефону» бот повідомляє про успішну операцію та запитує ввести номер особового рахунку ДТЕК. Також бот надає інформацію де можна знайти номер. Якщо особисто дізнатися номер не вдається, користувачу пропонується зателефонувати до контактного центру за наданими номерами.

На рис. 2.8 зображено результат виконаних кроків старту роботи з ботом.

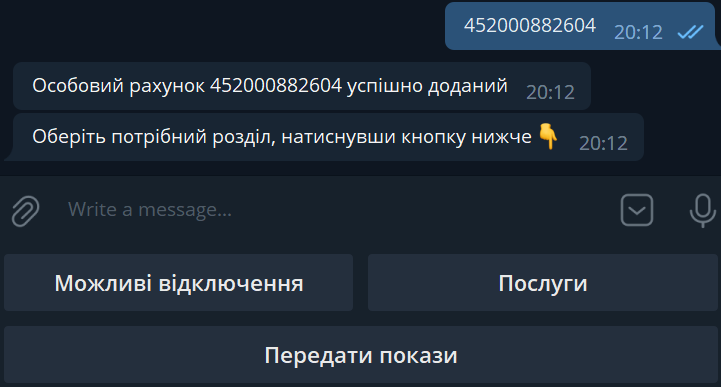


Рисунок 2.8 – Результат виконання всіх кроків з старту боту

Після завершення всіх кроків, необхідних для користування сервісом бот відображає панель з кнопками управління, розташованими в нижній частині екрана, що забезпечує зручну навігацію та взаємодію. За допомогою цих кнопок користувачі можуть отримати доступ до різних функцій та отримувати сповіщення про необхідні дії.

На рис. 2.9 зображено результат команди «Можливі відключення».

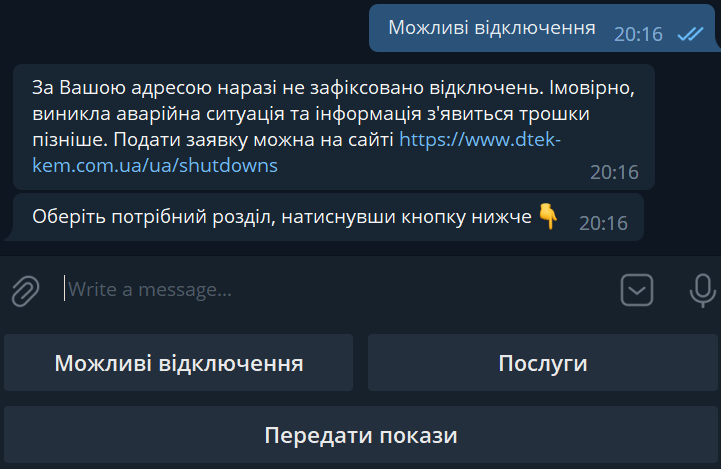


Рисунок 2.9 – Результат команди «Можливі відключення»

Команда, доступна користувачам, надає цінну інформацію про можливі перебої в електропостачанні. Ця інформація подається у звичайному текстовому форматі, з детальною інформацією про наявність електроенергії, причини відключень та рекомендації щодо усунення аварійних ситуацій.

Такий текстовий формат має певні переваги. Користувачі отримують стислий огляд поточної ситуації, що дозволяє їм швидко зрозуміти стан енергопостачання. Оновлення в режимі реального часу дозволяє енергопостачальнику оперативно інформувати громадян про аварійні ситуації та надавати більш детальні описи, оскільки телеграм-повідомлення мають високий ліміт на кількість символів.

Однак такий підхід має свої обмеження. Оскільки інформація надається виключно у текстовому вигляді, без жодних ілюстрацій чи візуалізацій, користувачу може бути незручно сприймати інформацію. Дані обмежуються поточною ситуацією, що обмежує можливості користувачів передбачити потенційні перебої в електропостачанні або вжити необхідних заходів заздалегідь.

Щоб усунути ці обмеження, включення візуальних елементів або додаткових функцій може покращити користувацький досвід. Інтеграція графічних зображень, таких як діаграми або схеми, може забезпечити візуальне зображення зон відключення або очікуваної тривалості. Крім того, надання розкладу або прогнозу майбутніх відключень дасть можливість користувачам планувати свою діяльність відповідно до них.

Поєднуючи текстову інформацію з візуальними зображеннями та перспективними графіками, користувачі матимуть більш повне уявлення про ситуацію з енергопостачанням. Це дозволить їм краще планувати свою щоденну або щотижневу діяльність і мінімізувати перебої, спричинені відключенням електроенергії.

На рис. 2.10 зображено результат команди «Передати покази».

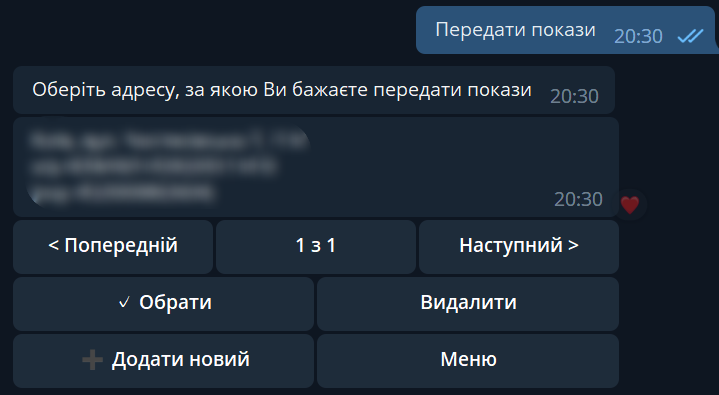


Рисунок 2.10 – Результат команди «Передати покази»

На відміну від сайту, перевага Telegram-бота полягає в тому, що в ньому зберігається інформація про користувача, наприклад, номер телефону та номер особового рахунку, пов'язані з договором з ДТЕК. Це дає змогу боту надсилати користувачам повідомлення про адреси будинків, які були раніше додані в чат за допомогою їхніх особистих кабінетів.

Крім того, бот надає зручний та зрозумілий інтерфейс для керування особистими записами в базі даних. Відправивши команду «Передати покази», користувачі отримують доступ до функціональної панелі, на якій відображаються різні опції. Натискання відповідних кнопок на цій панелі дозволяє користувачеві переглянути наявні записи. Кожен запис містить основні дані, такі як адреса будинку та особистий номер запису.

Бот також пропонує гнучкість у зміні або видаленні записів. Користувачі можуть видалити певний запис, натиснувши кнопку «Видалити», гарантуючи, що їхні записи залишаються актуальними і точними. Крім того, користувачі мають можливість додавати нові записи, натиснувши на кнопку «Додати новий», що дозволяє їм зручно розширювати свою базу даних керованих адрес.

Надаючи ці функції, Telegram-бот спрощує користувачам управління особистими записами. Він підвищує зручність, дозволяючи користувачам переглядати, редагувати та оновлювати свої записи в інтерфейсі чату, гарантуючи, що інформація залишається актуальною та релевантною. Такий спрощений підхід дозволяє користувачам зберігати точні дані та ефективно керувати пов'язаними з ними адресами з легкістю.

На рис. 2.11 зображено результат команди «Обрати» меню «Передати покази».

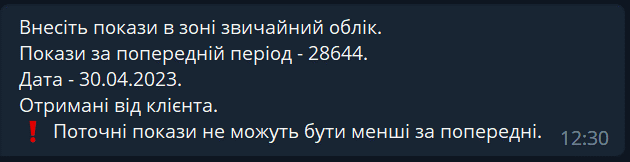


Рисунок 2.11 – Результат команди «Обрати» меню «Передати покази»

Після того, як користувач вибрав певний запис, він має можливість натиснути кнопку «Вибрати», що спонукає бота відобразити відповідну інформацію. Ця інформація включає в себе дані про дату останньої передачі показань та відповідний обсяг цих показань.

Крім того, бот бере на себе відповідальність за інформування користувачів про правила та необхідний формат подання нових показань. Надаються чіткі інструкції, які допомагають користувачеві точно зрозуміти належну процедуру передачі своїх показань.

Передача показань лічильників електроенергії через Інтернет пропонує значну зручність як для споживачів, так і для постачальників енергії. Цей сучасний метод усуває необхідність ручного зняття показань лічильників та паперового документообігу, приносячи численні переваги всім учасникам процесу.

Перш за все, передача показань через Інтернет економить цінний час та зусилля споживачів. Замість того, щоб призначати зустрічі або вручну записувати показання лічильників, вони можуть зручно передавати показання, не виходячи з дому чи офісу. Всього за кілька кліків користувачі можуть передавати дані в електронному вигляді, оминаючи необхідність фізичного відвідування або ручного введення даних.

Крім того, передача даних через Інтернет забезпечує більшу точність показань лічильників. Людські помилки, часто пов'язані з ручними записами, можна мінімізувати або повністю усунути. Це зменшує ймовірність розбіжностей або суперечок при виставленні рахунків, надаючи споживачам більш точну та надійну інформацію про використання енергії.

Для постачальників енергії передача показань лічильників через Інтернет спрощує їхню роботу та підвищує ефективність. Автоматизований збір даних усуває потребу в ручному введенні даних і зменшує адміністративні витрати. Це дозволяє постачальникам енергії більш ефективно розподіляти свої ресурси, зосереджуючись на наданні надійних послуг та вирішенні інших важливих аспектів своєї діяльності.

На рис. 2.12 зображено результат команди «Послуги».

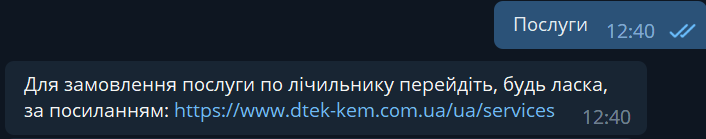


Рисунок 2.12 – Результат команди «Послуги»

Команда послуги відповідає за відображення актуальної інформації про послуги, які надає компанія ДТЕК. Бот надсилає посилання на сайт, де можна замовити послуги по лічильнику.

## 2.3 Гаряча лінія постачальника електроенергії

Гаряча лінія постачальника електроенергії слугує важливим засобом отримання інформації про графік відключень у місті та стан електропостачання. Звернувшись на гарячу лінію, споживачі можуть безпосередньо поспілкуватися з компетентними представниками, які можуть надати актуальну та точну інформацію.

Однією з основних переваг використання гарячої лінії є можливість отримувати інформацію про заплановані відключення в режимі реального часу. Представники на гарячій лінії мають доступ до останньої інформації та можуть інформувати користувачів про будь-які заплановані відключення електроенергії в їхньому районі. Це допомагає клієнтам залишатися в курсі подій та вжити необхідних заходів для мінімізації незручностей, спричинених відключеннями. Гаряча лінія слугує надійним джерелом для отримання інформації про поточний стан електропостачання в конкретному регіоні. Користувачі можуть запитувати про будь-які перебої в електропостачанні, аварійні ситуації або несподівані відключення. Представники гарячої лінії можуть надати оновлену інформацію щодо очікуваного часу відновлення, що дозволяє клієнтам планувати свою діяльність відповідно.

Перевагою гарячої лінії є безпосередня взаємодія з кваліфікованим персоналом, який може відповісти на будь-які конкретні запитання чи проблеми. Клієнти можуть отримати роз'яснення щодо причин відключення електроенергії, запросити додаткову інформацію про поточні роботи з технічного обслуговування або повідомити про будь-які проблеми, з якими вони можуть зіткнутися. Такий персоналізований підхід гарантує, що користувачі отримують відповіді, враховуючи їхні конкретні обставини.

Крім того, такий спосіб є цінним під час надзвичайних ситуацій. У випадку складних погодних умов, пошкодження інфраструктури або масових відключень електроенергії, гаряча лінія може надати клієнтам критично важливу інформацію та рекомендації. Це допомагає керувати очікуваннями, забезпечувати безпеку та сприяти швидкому відновленню роботи.

Таким чином, використання гарячої лінії постачальника електроенергії пропонує користувачам прямий та надійний засіб доступу до інформації про графік відключень та стан електропостачання. Оновлення інформації в режимі реального часу, персоналізована допомога та можливість відповісти на конкретні запитання роблять гарячу лінію цінним ресурсом для споживачів, які шукають своєчасну та точну інформацію щодо своїх послуг з електропостачання.

Хоча використання гарячої лінії, наданої постачальником електроенергії, може бути корисним у багатьох відношеннях, існує також кілька потенційних недоліків, пов'язаних з цим каналом зв'язку:

* обмежена доступність гарячих ліній. Вони можуть мати обмежений час роботи, що може бути незручним для споживачів, які потребують допомоги поза встановленими часовими рамками. Це обмеження може створювати проблеми, особливо під час надзвичайних ситуацій або ситуацій, що потребують негайної підтримки;
* велика кількість дзвінків і час очікування. Під час пікових періодів або масових відключень гарячі лінії можуть стикатися зі сплеском кількості дзвінків, що призводить до тривалого часу очікування. Це може призвести до розчарування клієнтів, які шукають своєчасну інформацію або допомогу;
* гарячі лінії переважно покладаються на усне спілкування, а це означає, що користувачі можуть не мати доступу до наочних посібників чи ілюстрацій для кращого розуміння складних питань чи інструкцій. Відсутність візуальної інформації може бути недоліком при спробі зрозуміти технічні деталі або кроки з усунення неполадок;
* існує ймовірність непорозуміння. Усне спілкування по телефону іноді може призвести до непорозумінь або неправильного тлумачення, особливо при передачі конкретних деталей або інструкцій. Це може призвести до надання невірної інформації або неналежного реагування на запити клієнтів;
* якість обслуговування клієнтів на гарячій лінії може бути непостійною. Рівень допомоги може відрізнятися залежно від того, хто саме відповідає на дзвінок. У той час як одні представники можуть бути обізнаними, корисними та здатними ефективно вирішувати проблеми, інші можуть не мати необхідного досвіду або надавати непослідовну допомогу.

## 2.4 Інформаційні джерела

Вторинні джерела інформації, такі як вебсайти, статті та платформи на кшталт Telegram-каналів, стали важливими інструментами для мешканців, які шукають актуальну інформацію про відключення електроенергії та поточний стан електромереж у своєму місті. Ці джерела пропонують ефективний і доступний спосіб залишатися в курсі подій, що дозволяє людям відповідно планувати свою діяльність і мінімізувати вплив потенційних відключень.

Вторинні джерела інформації, такі як вебсайти, статті та платформи на кшталт Telegram-каналів, стали важливими інструментами для мешканців, які шукають актуальну інформацію про відключення електроенергії та поточний стан електромереж у своєму місті. Ці джерела пропонують ефективний і доступний спосіб залишатися в курсі подій, що дозволяє людям відповідно планувати свою діяльність і мінімізувати вплив потенційних відключень.

Статті, опубліковані в засобах масової інформації або спеціалізованих енергетичних виданнях, також слугують цінними додатковими джерелами. Вони надають інформацію про стан електромережі, в тому числі про нещодавні відключення, прогрес у відновленні електропостачання та поточні проблеми, з якими стикаються енергетичні компанії. Ці статті дають уявлення про ширший контекст питань енергопостачання та висвітлюють відповідні події, ініціативи чи заходи, що впроваджуються для підвищення стійкості енергосистеми.

Платформи обміну миттєвими повідомленнями, такі як Telegram, набули популярності як канали поширення оперативної інформації, в тому числі новин, пов'язаних з відключеннями електроенергії. Телеграм-канали, якими керують енергорозподільчі компанії або місцеві органи влади, в режимі реального часу сповіщають про відключення, орієнтовний час відновлення та іншу важливу інформацію. Підписавшись на ці канали, мешканці отримують своєчасні сповіщення на свої пристрої, що дозволяє їм скоригувати плани, вжити необхідних запобіжних заходів або підготувати резервні джерела живлення, якщо це необхідно.

Ці джерела інформації мають кілька переваг для мешканців. Вони забезпечують зручність, надаючи доступ до інформації з дому або в дорозі за допомогою смартфонів, ноутбуків або планшетів. Крім того, користувачі можуть персоналізувати попередження або сповіщення на основі свого місцезнаходження або конкретних потреб в електроенергії, забезпечуючи отримання цільових оновлень, що стосуються їхньої місцевості або ситуації.

Важливо знати про недоліки, щоб приймати обґрунтовані судження та рішення на основі інформації, отриманої з цих джерел.

Можливість отримання неточної або недостовірної інформації. Не всі вебсайти, статті або телеграм-канали проходять ретельну перевірку фактів або верифікацію. Як наслідок, існує ризик натрапити на застарілу, оманливу або навіть неправдиву інформацію. Користувачі повинні проявляти обережність і перевіряти інформацію з кількох авторитетних джерел, щоб переконатися в її точності.

Не завжди може бути забезпечено необхідний контекст або повне розуміння певного питання. Вебсайти, статті або телеграм-канали часто зосереджуються на конкретних аспектах або подіях, що може обмежувати широту доступної інформації. Як наслідок, користувачі можуть пропустити важливі деталі або не зрозуміти ширшого значення наданої інформації.

Навіть якщо джерела прагнуть надавати актуальну інформацію, може виникнути затримка між подією та її публікацією або розповсюдженням через ці канали. Ця затримка може бути проблематичною, особливо в ситуаціях, коли оновлення інформації в режимі реального часу має вирішальне значення. Користувачі повинні усвідомлювати, що покладання виключно на вторинні джерела може не забезпечити найактуальнішу та своєчасну інформацію.

# 3 ВИБІР ТЕХНОЛОГІЧНОГО РІШЕННЯ

## 3.1 Месенджер Telegram як фронтенд платформа

Telegram, популярний додаток для обміну повідомленнями, став чимось більшим, ніж просто інструментом комунікації. Він перетворився на комплексну платформу, яка надає розробникам широкі можливості для створення та розгортання функціональних чат-ботів. Завдяки своєму відкритому API та надійній інфраструктурі, Telegram дав можливість приватним особам та організаціям використовувати месенджер як фронтенд-платформу для створення інтерактивних та інтелектуальних ботів [5].

Інфраструктура Telegram побудована на розподіленій мережі серверів, розташованих у різних регіонах. Така структура забезпечує високу доступність і мінімальну затримку, що дозволяє безперешкодно і швидко взаємодіяти між користувачами та чат-ботами. Розподілений характер інфраструктури також додає додатковий рівень безпеки, що робить Telegram надійною платформою для розробки ботів.

В основі інфраструктури Telegram лежить API ботів. Цей API дозволяє розробникам легко створювати чат-ботів і керувати ними. Він надає простий інтерфейс для надсилання та отримання повідомлень, обробки взаємодії з користувачами та доступу до різних функцій Telegram. Bot API підтримує зв'язок як на основі HTTP, так і на основі WebSocket, пропонуючи гнучкість і масштабованість для розробки ботів.

Однією з ключових особливостей, які роблять Telegram привабливою платформою для створення чат-ботів, є підтримка вбудованого режиму. Завдяки інлайн-режиму розробники можуть легко інтегрувати своїх ботів у розмови користувачів. Замість того, щоб ініціювати окремий чат з ботом, користувачі можуть просто згадати ім'я користувача бота у своїх розмовах і запустити його дії. Такий спрощений підхід покращує користувацький досвід і заохочує більш природну та інтерактивну взаємодію з ботами.

Крім того, Telegram підтримує мультимедійні та інтерактивні елементи в чат-ботах. Розробники можуть використовувати такі функції, як кнопки, вбудовані клавіатури та меню для створення цікавих користувацьких інтерфейсів. Ці елементи дозволяють користувачам взаємодіяти з ботом через інтуїтивно зрозумілі та візуально привабливі інтерфейси, покращуючи загальний користувацький досвід.

Платформа дозволяє розробникам створювати кастомні клавіатури та реалізовувати запити на зворотний виклик. Кастомні клавіатури надають користувачам заздалегідь визначені опції, спрощуючи процес взаємодії. Запити зворотного виклику дозволяють розробникам ефективно обробляти взаємодію користувачів з вбудованими кнопками, забезпечуючи динамічну та чуйну поведінку бота.

Telegram також надає широку підтримку сторонніх інтеграцій та API, що дозволяє розробникам використовувати зовнішні сервіси та ресурси у своїх чат-ботах. Ця можливість інтеграції дозволяє ботам отримувати дані із зовнішніх джерел, виконувати складні обчислення або взаємодіяти з іншими додатками, що робить їх більш потужними та універсальними.

Ще одним важливим аспектом інфраструктури Telegram є її спрямованість на конфіденційність та безпеку. Telegram пропонує наскрізне шифрування як для індивідуальних, так і для групових розмов, гарантуючи, що дані користувачів залишаються конфіденційними. Такий акцент на конфіденційності зробив Telegram кращим вибором для користувачів, які цінують захист даних, тим самим підвищуючи довіру до чат-ботів, створених на цій платформі, і довіру до них.

## 3.2 Мова програмування Python

Python – популярна мова програмування, відома своєю простотою, читабельністю та великими бібліотеками, що робить її чудовим вибором для створення Telegram-ботів. Порівняно з іншими мовами програмування, Python має низку переваг, які роблять її найкращим варіантом для розробки ботів. Однак він також має кілька недоліків, які слід враховувати. Розглянемо обидва аспекти.

Переваги використання Python для створення Telegram-бота:

* має чистий і зрозумілий синтаксис, що полегшує його вивчення і розуміння для початківців. Читабельність мови дозволяє розробникам писати чистий код, який легко підтримувати, зменшуючи ймовірність помилок і покращуючи загальний процес розробки;
* Python може похвалитися великою екосистемою бібліотек і фреймворків, які надають готові інструменти та функціональність, що робить розробку ботів більш ефективною. Зокрема, для розробки Telegram-ботів такі бібліотеки, як python-telegram-bot і Telethon, надають комплексні API та функції, що спрощують процес взаємодії з API Telegram-ботів;
* має чудову документацію, з великою кількістю ресурсів, доступних в Інтернеті. Крім того, Python має велику та активну спільноту розробників, які завжди готові допомогти та надати підтримку. Це означає, що якщо розробник зіткнеться з будь-якими проблемами під час розробки Telegram-бота, він може легко знайти допомогу від спільноти;
* широка підтримка бібліотек Python дозволяє розробникам використовувати існуючі модулі для різних завдань. Наприклад, такі бібліотеки, як BeautifulSoup та requests, можна використовувати для вебскрепінгу та взаємодії з вебсервісами. Це забезпечує швидку розробку і зменшує необхідність винаходити велосипед;
* python є кросплатформенною мовою, що означає, що один і той самий код може працювати на різних операційних системах без значних модифікацій. Це особливо корисно для розробки ботів для Telegram, оскільки боти повинні працювати на серверах або локальних машинах, щоб взаємодіяти з API Telegram.

Недоліки використання Python для створення Telegram-бота:

* хоча Python є універсальною мовою, вона може бути не найпродуктивнішим вибором для певних сценаріїв. Оскільки Python є інтерпретованою мовою, вона може бути повільнішою порівняно з компільованими мовами, такими як C++ або Go. Однак для більшості випадків використання Telegram-ботів продуктивності Python більш ніж достатньо;
* глобальне блокування інтерпретатора (GIL) в Python може обмежувати здатність мови ефективно використовувати кілька ядер, що впливає на масштабованість для високопаралельних додатків. Однак для більшості Telegram-ботів це обмеження не є суттєвою проблемою, оскільки взаємодія з Telegram API в основному базується на обробці запитів користувачів;
* якщо необхідно створити Telegram-бота з мобільним користувацьким інтерфейсом, Python може бути не найкращим вибором. Хоча Python підтримує фреймворки для розробки мобільних додатків, такі як Kivy та BeeWare, вони використовуються рідше, ніж нативні мови, такі як Java або Swift.

При розробці Telegram-бота на Python поєднання бібліотек Selenium, BeautifulSoup, Multithreading та Mongoengine утворює потужний стек технологій для реалізації потрібного функціоналу. Кожна бібліотека слугує певній меті та робить свій внесок у різні аспекти функціональності бота, що робить їх ефективною комбінацією.

Selenium, інструмент автоматизації, дозволяє розробникам програмно керувати браузерами. Він корисний при взаємодії з вебсайтами, які не мають публічного API або потребують взаємодії на основі JavaScript. За допомогою Selenium можна витягувати інформацію з вебсайтів, взаємодіяти з сервісами та виконувати дії, які потребують інтерфейсу.

BeautifulSoup, бібліотека Python для розбору HTML та XML документів, доповнює Selenium, витягуючи необхідні дані з вебсторінок. Вона спрощує процес вилучення, надаючи простий у використанні інтерфейс. Поєднуючи Selenium і BeautifulSoup, можна витягувати певну інформацію з вебсторінок і використовувати її у своєму Telegram-боті.

Багатопоточність підвищує швидкість реакції бота в чаті, дозволяючи запускати кілька потоків одночасно в рамках одного процесу. Це допомагає обробляти кілька запитів користувачів одночасно. Завдяки багатопоточності можна ефективно обробляти команди користувачів, отримувати дані із зовнішніх джерел і виконувати трудомісткі операції, гарантуючи, що бот зможе впоратися з одночасною взаємодією без затримок і блокувань.

Mongoengine, бібліотека об'єктно-документного маркування (ODM) для MongoDB, полегшує взаємодію з базами даних MongoDB. Вона спрощує такі завдання, як створення, оновлення та пошук даних. Використовуючи Mongoengine, ви можете зберігати та отримувати налаштування користувача, кешувати дані та зберігати будь-яку важливу інформацію про ботів та користувачів у базі даних MongoDB.

Поєднавши ці бібліотеки, можна створити надійний стек для Telegram-бота. Selenium автоматизує вебвзаємодію, BeautifulSoup аналізує та витягує дані, багатопотоковість обробляє одночасні запити, а Mongoengine забезпечує безшовну інтеграцію з базою даних MongoDB. Ця комбінація дозволяє створити бота з різноманітними функціональними можливостями, включаючи вебскрепінг, маніпуляції з даними та ефективну взаємодію з користувачами.

## 3.3 Система контролю версіями Git

Git – це широко розповсюджена розподілена система контролю версій (VCS), яка зробила революцію в управлінні проєктами з розробки програмного забезпечення. Розроблений Лінусом Торвальдсом у 2005 році, Git став стандартом де-факто для відстеження змін у вихідному коді та спільної роботи над проєктами з розробки програмного забезпечення. Він надає надійний та ефективний спосіб керувати версіями коду, розгалужувати та об'єднувати зміни коду, а також співпрацювати з іншими розробниками [9].

Однією з ключових переваг Git є його розподілена природа. Кожен розробник, який працює над проєктом, має повну копію сховища, включаючи всю історію змін. Це означає, що розробники можуть працювати автономно і незалежно, що робить його ідеальним для розподілених команд або ситуацій з обмеженим підключенням до мережі. Git дозволяє розробникам фіксувати зміни в локальному репозиторії, а потім переміщувати їх у віддалений репозиторій, коли вони будуть готові поділитися ними з іншими.

Git працює на основі концепції гілок, які є незалежними напрямками розробки. Розробники можуть створювати гілки для роботи над новими функціями, виправленнями помилок або експериментів, не впливаючи на основну кодову базу. Це дозволяє здійснювати паралельну розробку і полегшує управління декількома завданнями одночасно. Гілки можна об'єднати назад в основну кодову базу, коли зміни будуть завершені і протестовані.

Ще однією потужною особливістю Git є його здатність обробляти злиття коду. Коли кілька розробників працюють над однією і тією ж кодовою базою і вносять зміни одночасно, Git може розумно об'єднати ці зміни разом. У випадках, коли виникають конфлікти, Git надає інструменти для їх вирішення вручну. Це гарантує, що зміни від різних розробників будуть об'єднані плавно і без помилок.

Git також надає повний набір інструментів для перевірки історії кодової бази. Розробники можуть легко переглядати попередні комміти, порівнювати різні версії файлів і відстежувати еволюцію коду в часі. Це допомагає зрозуміти, чому були прийняті ті чи інші рішення, відстежити походження помилок та відкотити зміни, якщо це необхідно.

На рис. 3.1 зображено візуалізацію CI/CD патерну розробки.

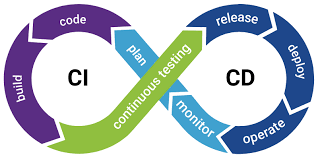


Рисунок 3.1 – Візуалізація CI/CD патерну розробки

(https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRs06WZ7cmn2LOH2z-liycjWe2Vfg51niXz-u9yqDPs7Q&s)

CI/CD (Continuous Integration/Continuous Deployment – безперервна інтеграція/безперервне розгортання) – це підхід до розробки програмного забезпечення, який фокусується на автоматизації та впорядкуванні процесу створення, тестування та розгортання змін у програмному забезпеченні. Git, як система контролю версій, відіграє важливу роль у реалізації робочих процесів CI/CD [6].

У CI/CD розробники часто об'єднують свої зміни коду у спільний репозиторій. Git надає такі функції, як розгалуження, злиття та pull-запити, щоб полегшити цей процес. Розробники створюють гілки для незалежної роботи над своїми змінами, фіксують їх у локальному сховищі, а потім переносять ці зміни до центрального сховища.

Як тільки зміни коду потрапляють до центрального репозиторію, система CI, інтегрована з Git, автоматично запускає збірку та тестування змін. Система CI отримує останній код, компілює його і запускає автоматизовані тести, щоб переконатися, що зміни є дійсними. Інтеграція Git з різними тестовими фреймворками дозволяє безперешкодно виконувати тести.

Якщо тести проходять успішно, зміни можуть бути злиті в основну гілку. Можливості розгалуження та об'єднання Git дозволяють контролювати просування змін коду в різних гілках, що представляють різні середовища розгортання (наприклад, розробка, стадія, виробництво). Це дозволяє створювати окремі конфігурації та сценарії розгортання для кожного середовища.

Крім того, Git відіграє важливу роль на етапі безперервного розгортання. Після того, як зміни коду пройдуть необхідні тести, система CI/CD може автоматично запустити конвеєр розгортання. Цей конвеєр отримує останній код зі сховища, пакує додаток і розгортає його в цільовому середовищі. Можливості Git з контролю версій та чітка історія змін забезпечують надійність та контроль процесу розгортання.

## 3.4 Реляційні бази даних

Реляційні бази даних – це тип системи управління базами даних (СКБД), яка організовує дані в таблиці, що складаються з рядків і стовпців. Вони засновані на реляційній моделі, яку представив Едгар Ф. Кодд у 1970-х роках. Ця модель представляє дані та їхні зв'язки за допомогою набору таблиць, де кожна таблиця представляє сутність, а кожен рядок – конкретний екземпляр або запис цієї сутності [7].

Основним поняттям реляційної бази даних є поняття зв'язку між таблицями. Зв'язки встановлюються за допомогою ключів, які є унікальними ідентифікаторами для кожного рядка в таблиці. Первинний ключ однозначно ідентифікує кожен запис у таблиці, тоді як зовнішні ключі встановлюють зв'язки між таблицями, посилаючись на первинний ключ іншої таблиці. Це дає змогу відображати складні зв'язки між різними об'єктами, а також шукати та маніпулювати даними на основі цих зв'язків.

Реляційні бази даних мають ряд переваг, таких як

* дані в реляційних базах даних структуровані, із заздалегідь визначеними схемами, які визначають таблиці, стовпці та зв'язки між ними. Така структура забезпечує чіткий та організований спосіб зберігання та управління даними;
* реляційні бази даних забезпечують цілісність даних шляхом застосування таких обмежень, як унікальність первинного ключа, посилальна цілісність через зовнішні ключі та правила перевірки даних. Ці обмеження гарантують, що дані залишаються точними та узгодженими;
* реляційні бази даних надають стандартизовану мову запитів, зазвичай SQL (мова структурованих запитів), яка дозволяє користувачам отримувати, фільтрувати та агрегувати дані у гнучкий спосіб. Це полегшує виконання складних запитів і створення звітів на основі даних;
* бази даних призначені для обробки великих обсягів даних і можуть масштабуватися вертикально (додаючи більше ресурсів на один сервер) або горизонтально (розподіляючи дані між декількома серверами), щоб задовольнити зростаючі потреби в даних;
* система дотримується властивостей ACID (Atomicity, Consistency, Isolation, Durability – атомарність, узгодженість, ізоляція, довговічність), забезпечуючи надійну обробку транзакцій. Властивості ACID гарантують, що модифікації даних відбуваються за принципом «все або нічого», зберігаючи узгодженість і цілісність даних.

Однак, реляційні бази даних також мають деякі обмеження, такі як

* заздалегідь визначені схеми реляційних баз даних можуть бути обмежуючими при роботі з даними, які погано вписуються в жорстку структуру. Модифікація схеми вимагає ретельного планування і може зайняти багато часу;
* міркування щодо продуктивності. Зі збільшенням обсягу даних у реляційних базах даних можуть виникати проблеми з продуктивністю, особливо при обробці складних запитів, що включають кілька таблиць і великі набори даних. Для підтримки хорошої продуктивності необхідні правильні методи індексування та оптимізації запитів;
* проблеми з масштабуванням. Хоча реляційні бази даних можуть масштабуватися, горизонтальне масштабування на декількох серверах може бути складним і може вимагати застосування передових технологій, таких як «шардеризація» або розбиття бази даних на розділи;
* реплікація та відмовостійкість. Забезпечення високої доступності та відмовостійкості реляційних баз даних часто вимагає додаткових заходів, таких як реплікація баз даних, що ускладнює архітектуру системи.

Незважаючи на ці обмеження, реляційні бази даних залишаються широко використовуваним і важливим інструментом у багатьох галузях. Вони особливо підходять для структурованих даних з чітко визначеними взаємозв'язками і пропонують надійне і перевірене рішення для зберігання, пошуку та управління даними.

## 3.5 Документно-орієнтовані бази даних

Документно-орієнтовані бази даних, також відомі як сховища документів, – це тип NoSQL (нереляційних) баз даних, які зберігають і отримують дані у форматі документів. На відміну від традиційних реляційних баз даних, які організовують дані в таблиці з фіксованими схемами, документно-орієнтовані бази даних призначені для обробки неструктурованих або напівструктурованих даних у вигляді документів [8].

У документно-орієнтованій базі даних документ є автономною одиницею даних, яка складається з пар ключ-значення або ключ-масив, де значення або масиви можуть містити різні типи даних, такі як рядки, числа, масиви або навіть вкладені документи. Документи зазвичай представлені у популярних форматах, таких як JSON (JavaScript Object Notation) або BSON (Binary JSON), які дозволяють легко серіалізувати та десеріалізувати їх.

Однією з ключових переваг документно-орієнтованих баз даних є їхня гнучкість і відсутність схем. На відміну від реляційних баз даних, які вимагають заздалегідь визначених схем і жорсткої структури таблиць, сховища документів дозволяють розробникам зберігати різні типи документів в одній колекції без застосування фіксованої схеми. Така гнучкість особливо корисна в сценаріях, де дані є динамічними або часто змінюються, оскільки дозволяє легше еволюціонувати схеми та адаптуватися до вимог, що змінюються.

Документно-орієнтовані бази даних чудово справляються з напівструктурованими та ієрархічними даними, такими як каталоги товарів, профілі користувачів, публікації в блогах або стрічки соціальних мереж. Ці бази даних надають потужні можливості запитів, дозволяючи розробникам знаходити документи на основі різних критеріїв, включаючи значення полів, діапазони або навіть складні вкладені умови. За допомогою вторинних індексів сховища документів можуть ефективно знаходити документи на основі певних атрибутів, забезпечуючи швидкий і гнучкий доступ до даних.

Ще однією значною перевагою документно-орієнтованих баз даних є горизонтальна масштабованість. Ці бази даних призначені для горизонтального масштабування шляхом розподілу даних між кількома серверами або кластерами, що дозволяє їм обробляти великі обсяги даних і високі навантаження на трафік. Завдяки поділу та реплікації даних сховища документів можуть забезпечити високу доступність, відмовостійкість і здатність справлятися з величезними навантаженнями.

Сховища документів також сприяють швидкій розробці, оскільки вони добре узгоджуються з сучасними фреймворками веброзробки та API, які працюють з даними JSON. Розробники можуть безпосередньо зіставляти документи з об'єктами в коді своїх додатків, спрощуючи рівень доступу до даних і зменшуючи невідповідність між додатком і базою даних.

Однак, документно-орієнтовані бази даних можуть не підходити для всіх випадків використання. Оскільки їм не вистачає підтримки транзакцій між кількома документами, вони можуть бути не найкращим вибором для сценаріїв, які вимагають суворих гарантій ACID (атомарність, узгодженість, ізольованість, довговічність) або складних реляційних операцій. Крім того, хоча сховища документів надають потужні можливості запитів, вони можуть не працювати так ефективно, як спеціалізовані бази даних для певних типів складних запитів або агрегацій.

У контексті документно-орієнтованих баз даних JSON зазвичай використовується як формат даних для зберігання та пошуку документів. JSON забезпечує гнучкий і широко підтримуваний формат для представлення напівструктурованих або ієрархічних даних, що робить його природним вибором для документно-орієнтованих баз даних.

Документно-орієнтовані бази даних зберігають дані в документах, і ці документи, як правило, представлені у вигляді об'єктів JSON. Кожен документ містить пари ключ-значення, де ключами є рядки, а значеннями можуть бути рядки, числа, булеві функції, масиви, вкладені об'єкти або навіть нуль. Це дозволяє представляти складні структури даних в межах одного документа.

Документно-орієнтовані бази даних використовують гнучкість JSON для зберігання та пошуку документів без застосування фіксованої схеми. На відміну від реляційних баз даних, де таблиці мають заздалегідь визначену структуру, сховища документів дозволяють документам у колекції мати різні поля та структури. Така динамічна схема дозволяє легко адаптуватися до мінливих вимог до даних і сприяє швидкій розробці.

Простота, читабельність і сумісність JSON з різними мовами програмування та платформами роблять його ідеальним вибором для представлення та обміну даними в документно-орієнтованих базах даних.

## 3.6 Система управління базами даних MongoDB

MongoDB – популярна і потужна система керування базами даних (СКБД), що належить до категорії NoSQL (нереляційних) баз даних. Вона призначена для обробки великих обсягів неструктурованих і напівструктурованих даних, що робить її добре придатною для сучасних додатків, які мають справу з різноманітними форматами даних і складними моделями даних. Гнучкість, масштабованість і продуктивність MongoDB сприяли її широкому впровадженню в організаціях усіх розмірів [4].

Однією з ключових особливостей MongoDB є її документно-орієнтована модель даних. Замість використання таблиць і рядків, як у традиційних реляційних базах даних, MongoDB зберігає дані у гнучких JSON-подібних документах. Ці документи можуть мати різну структуру, що дозволяє розробникам зберігати, отримувати та оновлювати дані у спосіб, близький до мови програмування, яка використовується в їхньому додатку. Така гнучкість усуває необхідність у складних об'єднаннях і дозволяє легко модифікувати схеми даних відповідно до вимог програми.

Ще однією перевагою MongoDB є її здатність до горизонтального масштабування на декількох серверах і розподілу даних між кластерами машин. Вона використовує концепцію шардингу, коли дані розділяються і розподіляються між різними шардами, забезпечуючи високу доступність і ефективний розподіл даних. Така горизонтальна масштабованість робить MongoDB ідеальним вибором для роботи з великомасштабними додатками з високим трафіком і вимогливими робочими навантаженнями.

MongoDB також надає надійну підтримку реплікації, забезпечуючи довговічність та відмовостійкість даних. Вона дозволяє створювати набори реплік, які є самовідновлюваними кластерами екземплярів MongoDB. У наборі реплік декілька вузлів зберігають ідентичні копії даних, що забезпечує автоматичне обхідне копіювання у разі відмови вузла. Така надмірність забезпечує високу доступність і надійність даних, що робить MongoDB придатною для критично важливих додатків.

Розробники можуть взаємодіяти з MongoDB за допомогою мови запитів, яка базується на JavaScript. Вона пропонує багатий набір можливостей для створення запитів, включаючи фільтрацію, сортування, агрегацію та геопросторові запити. Крім того, MongoDB надає повний набір драйверів для різних мов програмування, що полегшує інтеграцію з різними стеками додатків.

Крім того, MongoDB пропонує широкий спектр функцій для підвищення продуктивності розробників та управління даними. Ці функції включають індексування для ефективного пошуку даних, підтримку повнотекстового пошуку, потужний фреймворк агрегації для комплексного аналізу даних, підтримку транзакцій для атомарних операцій, а також широкі можливості ведення журналів і моніторингу для оптимізації продуктивності.

Розробка Telegram-бота на Python за допомогою MongoDB має кілька переваг:

* гнучка схема MongoDB добре узгоджується з розробкою Telegram-ботів, дозволяючи працювати з різними типами даних без попередньо визначених структур або міграцій;
* драйвер PyMongo забезпечує безшовну інтеграцію, пропонуючи багатофункціональний API для CRUD-операцій і складних запитів;
* обробка даних у реальному часі за допомогою потоків змін MongoDB дозволяє миттєво реагувати на події, підвищуючи інтерактивність;
* MongoDB має активну спільноту розробників, велику кількість документації та зручні інструменти, такі як MongoDB Compass та Atlas;
* широко використовувана авторитетними організаціями, MongoDB є надійним і перевіреним вибором для Python Telegram-ботів виробничого рівня.

Хоча існують й інші варіанти баз даних для розробки Telegram-ботів на Python, гнучкість, масштабованість, можливості роботи в режимі реального часу, інтеграція з Python та потужна підтримка спільноти роблять MongoDB привабливим вибором.

# 4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 4.1 Технічне завдання на розробку програмного забезпечення

Метою цієї роботи є розробка Telegram-бота на Python, який слугуватиме надійним та ефективним засобом сповіщення користувачів про можливі відключення електроенергії в місті. Відключення електроенергії може спричинити незручності та перебої в роботі приватних осіб та підприємств, а своєчасне інформування має вирішальне значення для планування та вжиття необхідних запобіжних заходів.

Цілі, поставлені при створенні програмного забезпечення:

* телеграм-бот надаватиме користувачам сповіщення в режимі реального часу про можливі відключення електроенергії в місті. Бот збиратиме інформацію про відключення з відповідних джерел та оперативно доставлятиме її підписникам;
* користувачі матимуть можливість налаштовувати свої уподобання щодо сповіщень на основі групи, яка їх цікавить;
* додаток забезпечить точність і надійність інформації про відключення. Він інтегруватиметься з авторитетними джерелами, такими як енергопостачальні компанії або офіційні урядові канали, для отримання перевірених даних;
* бот буде проактивно моніторити стан електромереж та завчасно повідомляти користувачів про будь-які планові роботи або заплановані відключення. Ця функція дозволить користувачам відповідно планувати свою діяльність та мінімізувати незручності;
* Telegram-бот матиме зручний інтерфейс для полегшення взаємодії. Користувачі зможуть легко підписуватися, скасовувати підписку та змінювати свої вподобання. Крім того, бот може пропонувати додаткові функції, такі як поширені запитання або поради щодо ефективного вирішення проблем, пов'язаних з відключенням електроенергії;
* конфіденційність користувачів та безпека даних будуть надзвичайно важливими. Telegram-бот дотримуватиметься найкращих галузевих практик захисту даних, гарантуючи, що особиста інформація користувачів обробляється безпечно та за їхньою згодою.

Телеграм-бот обслуговує різноманітну аудиторію, включаючи як звичайних мешканців міста, так і підприємців. Він адресований окремим особам, сім'ям та підприємствам, які шукають своєчасну та достовірну інформацію щодо можливих відключень електроенергії в їхньому районі.

Очікувані вигоди від використання додатку:

* користувачі можуть легко отримувати інформацію про відключення електроенергії через Telegram-бот. Замість того, щоб вручну перевіряти різні джерела або вдаватися до традиційних методів, таких як телефонні дзвінки чи відвідування вебсайтів, вони отримують сповіщення безпосередньо у своєму додатку Telegram. Такий спрощений підхід усуває необхідність постійного моніторингу та забезпечує безпроблемний досвід;
* автоматизуючи процес сповіщення, Telegram-бот економить користувачам цінний час і зусилля. Їм більше не потрібно активно шукати інформацію або чекати на оновлення від комунальних служб. Бот спрощує всю процедуру сповіщення, оперативно доставляючи актуальну інформацію;
* Telegram-бот прагне надавати достовірну та точну інформацію з перевірених каналів. Інтегруючись з авторитетними організаціями, такими як енергопостачальні компанії або державні установи, відповідальні за управління електромережами, бот пропонує користувачам перевірені дані про відключення. Це дає можливість користувачам приймати обґрунтовані рішення та планувати відповідно до них;
* сповіщення від бота в режимі реального часу інформують користувачів про відключення електроенергії в міру їх виникнення. Оперативно передаючи інформацію про відключення, бот дозволяє користувачам вжити необхідних заходів обережності, вивчити альтернативи, якщо це необхідно, та мінімізувати потенційні незручності. Сповіщення про заплановані відключення допомагають користувачам ефективно організувати свою діяльність;
* підприємці можуть використовувати Telegram-бот, щоб отримувати сповіщення про відключення, які впливають на їхній бізнес. Це дозволяє їм планувати свою діяльність, спілкуватися з клієнтами та забезпечувати мінімальний збій у наданні послуг. Своєчасні сповіщення про перебої допомагають бізнесу зменшити потенційні фінансові втрати та підтримувати високий рівень задоволеності клієнтів.

Функціонал та можливості Telegram-бота:

* управління за допомогою попередньо визначених команд, які можна налаштувати в спеціальному чаті для конфігурації бота.
* користувачі можуть взаємодіяти з ботом, надсилаючи певні команди для запуску різних дій та отримання інформації;
* користувачі можуть легко запросити у бота графік відключень електроенергії;
* зручно представляє графік відключень у вигляді зображення в додатку Telegram, що дозволяє користувачам переглядати його без особливих зусиль;
* користувачі можуть гнучко обирати графік відключень електроенергії для бажаної групи;
* надає список доступних груп, що дозволяє користувачам вибрати конкретну групу, яка їх цікавить;
* зареєстровані користувачі отримують проактивні попередження про заплановані відключення;
* оперативно повідомляє зареєстрованих користувачів про майбутні відключення електроенергії в обраній ними групі;
* зареєстровані користувачі можуть легко змінити групу, для якої вони отримують завчасні нагадування;
* віддавши певну команду, користувачі можуть оновити свої налаштування і швидко перейти до іншої групи;
* дозволяє користувачам вказати за скільки часу вони бажають отримати попередження про можливе відключення;
* підтримує кілька мов, що дозволяє користувачам взаємодіяти з ботом мовою, якій вони надають перевагу. Усі відповіді, повідомлення та сповіщення локалізуються та перекладаються відповідно до мовних уподобань користувача.

Telegram забезпечує узгоджену роботу з повідомленнями на всіх платформах. Незалежно від того, чи користуєтесь ви настільним комп'ютером, смартфоном або планшетом, користувачі можуть розраховувати на надійний обмін повідомленнями, безперебійну синхронізацію та легке спілкування.

## 4.2 Проектування інтерфейсу користувача

Дизайн інтерфейсу користувача (UI) – це процес створення візуального макета та взаємодії програмного додатку, вебсайту або будь-якого іншого цифрового продукту. Він зосереджений на проектуванні елементів і компонентів, які користувачі бачать і з якими взаємодіють на екрані, з метою створення зручного, візуально привабливого та інтуїтивно зрозумілого інтерфейсу.

На рис. 4.1 зображено профіль боту.

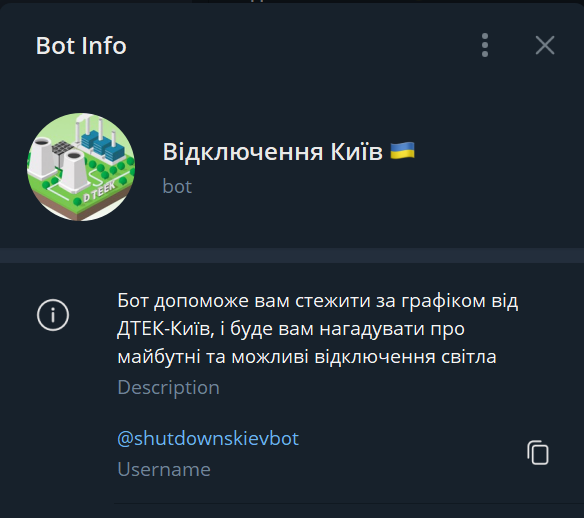


Рисунок 4.1 – Профіль боту

Профіль бота в Telegram – це набір інформації, який надає користувачам огляд і деталі про конкретного бота. Він слугує орієнтиром для користувачів, які взаємодіють з ботом, і допомагає їм зрозуміти його призначення, можливості та іншу важливу інформацію. Ось основні поля та інформація, які містяться в профілі бота в Telegram:

* назва бота – у цьому полі відображається назва бота, яка допомагає користувачам ідентифікувати його та посилатися на нього;
* ім'я користувача є унікальним ідентифікатором бота і використовується у форматі «@botusername» для згадування або взаємодії з ботом у Telegram;
* зображення профілю представляє візуальну ідентичність бота. Це може бути логотип, іконка або будь-яке зображення, що відображає призначення бота або його брендинг;
* опис надає короткий огляд функціональності, особливостей та призначення бота. Він допомагає користувачам зрозуміти, для чого призначений бот;
* команди – у цьому полі перераховані доступні команди, які користувачі можуть надсилати боту для виконання певних дій або отримання доступу до певних функцій. Кожна команда зазвичай починається з прямого слешу (/), за яким слідує ключове слово команди;
* про бота, розділ містить додаткову інформацію про бота, наприклад, про його розробника або команду розробників, контактну інформацію, вебсайт або будь-яку іншу важливу інформацію, яка може знадобитися користувачам;

Ці елементи в сукупності дають користувачам повне уявлення про бота та його можливості. Користувачі можуть звернутися до профілю бота, щоб вивчити доступні команди, дізнатися про політику конфіденційності та отримати доступ до контактної інформації, коли це необхідно.

На рис. 4.2 зображено вигляд повідомлення старту.

Повідомлення є відповіддю на команду /start у Telegram-боті. Воно містить вступне слово та пояснення призначення і функціоналу бота.

Повідомлення починається з привітання та ідентифікує бота як Київський бот з відключень. У ньому зазначено, що бот може надавати графік та сповіщення про майбутні відключення в місті. Інформація взята з офіційного сайту ДТЕК [2].

Щоб переглянути графік, користувач може скористатися командою /view із зазначенням номера групи або просто /view для перегляду власних відключень. Команда /info доступна для отримання інформації про налаштування користувача.

Далі у повідомленні пояснюється, як налаштувати сповіщення про можливі перебої у роботі. Воно містить покрокові інструкції.

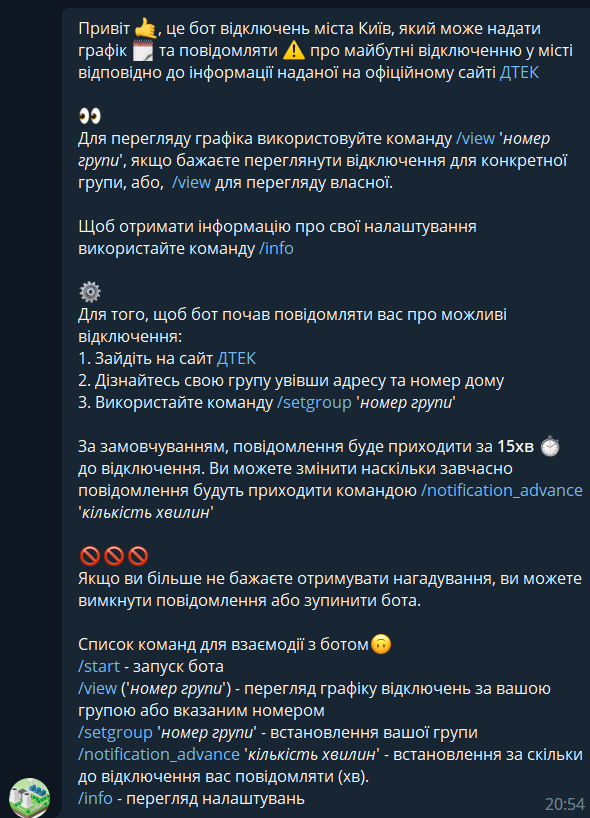


Рисунок 4.2 – вигляд повідомлення старту

Наприкінці повідомлення зазначено, що якщо користувач більше не бажає отримувати нагадування, він може вимкнути сповіщення або зупинити бота. У ньому також перераховані доступні команди для взаємодії з ботом.

Загалом, повідомлення інформує користувача про призначення бота, про те, як отримати доступ до розкладу відключень, як налаштувати сповіщення, а також надає варіанти керування сповіщеннями.

На рис. 4.3 зображено повідомлення невдалої спроби перегляду розкладу.

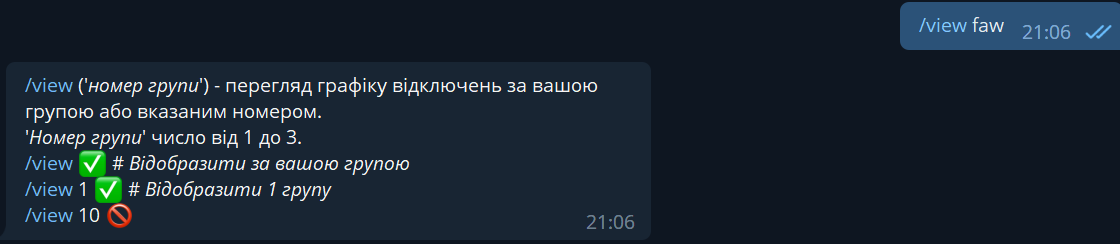


Рисунок 4.3 – Повідомлення невдалої спроби перегляду розкладу

Рекомендований формат команди для перегляду розкладу відключень наступний:

'/view', за яким слідує параметр 'номер групи', що вказує на конкретну групу або номер групи, для якої потрібно переглянути розклад. «Номер групи» має бути числовим значенням в діапазоні від 1 до 3.

Щоб пояснити правильне використання, у повідомленні наведено два приклади:

'/view' – Цей приклад демонструє, як відобразити розклад відключень для вашої групи. Використовуючи команду без зазначення номера групи, система покаже розклад, що відповідає призначеній групі користувача

'/view 1' – Цей приклад демонструє правильне використання для відображення розкладу відключень для певної групи, позначеної цифрою 1. Якщо ввести номер групи після команди, система згенерує розклад саме для цієї групи.

Однак варто зазначити, що спроба отримати доступ до групового розкладу з невірним номером призведе до помилки. Повідомлення демонструє про це за допомогою прикладу '/view 10', де число 10 означає невірний номер групи.

Надаючи чіткі інструкції, а також практичні приклади та візуальні підказки, повідомлення має на меті допомогти користувачам правильно використовувати команду «Старт», щоб отримати доступ до розкладу відключень для бажаних груп.

На рис. 4.4 зображено повідомлення вдалої спроби перегляду групи.



Рисунок 4.4 – Повідомлення вдалої спроби перегляду групи

Коли користувач правильно вводить команду, бот негайно реагує, надсилаючи повідомлення, яке містить не лише зображення графіка відключень, але й детальний підпис, що вказує на конкретну групу, до якої відноситься зображення.

Повідомлення, надіслане ботом, міститиме вкладену картинку, що відображає графік відключень. Зображення призначене для візуально доступного та інформативного представлення графіка, що дозволяє користувачам легко зрозуміти деталі запланованих відключень.

На рис. 4.5 зображено приклад вигляду зображення графіку відключень.

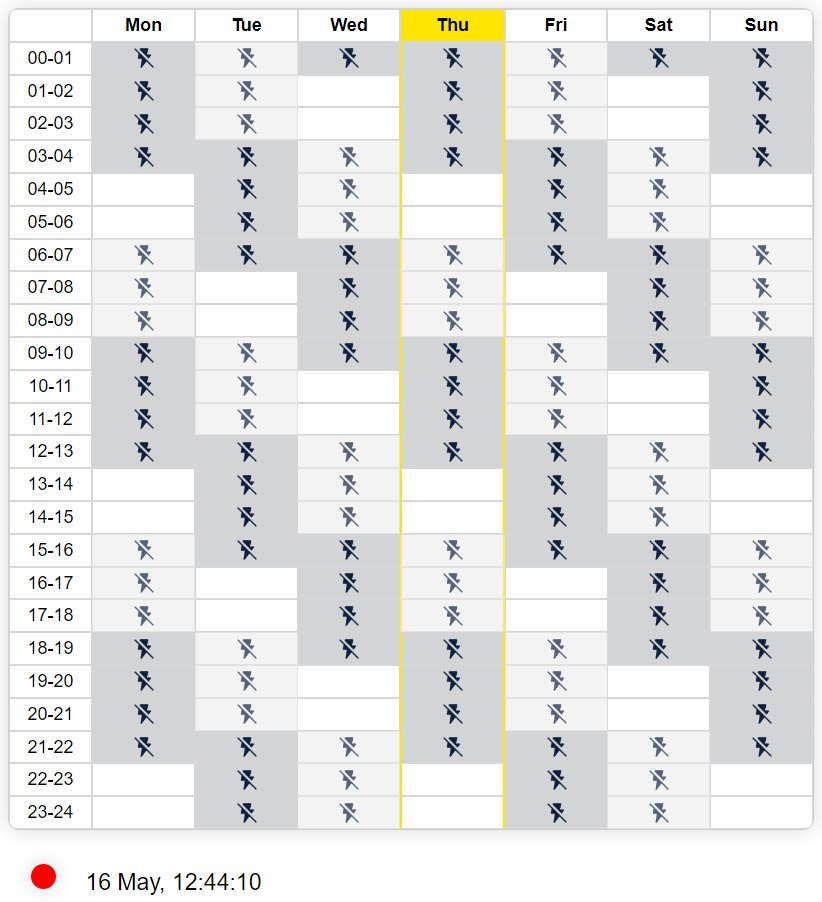


Рисунок 4.5 – Приклад вигляду графіку відключень

Це зображення оформлене у вигляді таблиці, стовпці якої відповідають дням тижня, а рядки – різним годинам протягом доби.

Структура таблиці дуже схожа на представлення графіка відключень на сайті ДТЕК, що забезпечує послідовність і звичність для користувачів. Таке узгодження з макетом сайту дозволяє користувачам легко орієнтуватися та інтерпретувати інформацію, що відображається на зображенні.

Для швидкого визначення поточного дня в таблиці відповідна колонка підсвічується яскравим жовтим кольором. Ця візуальна підказка дозволяє користувачам швидко знаходити розклад на поточний день і планувати відповідно до нього.

У кожній комірці таблиці використовуються стислі зображення, що відображають різні стани, пов'язані з доступністю електроенергії. Ці зображення слугують візуальними індикаторами та надають користувачам цінну інформацію з першого погляду. Зокрема, є три типи зображень:

* наявність електроенергії – електроенергія доступна протягом відповідного часового інтервалу;
* можливе відключення електроенергії – потенційна можливість тимчасового відключення електроенергії протягом зазначеного часу;
* відсутність електроенергії – повна відсутність електроенергії протягом зазначеного періоду часу.

Використовуючи ці візуальні зображення, користувачі можуть швидко зрозуміти стан електроенергії в різні години кожного дня, що дає їм змогу приймати обґрунтовані рішення і відповідно адаптувати свої плани.

Внизу таблиці надається додаткова інформація для забезпечення точності та надійності даних. Вона включає дату останнього оновлення даних, що інформує користувачів про свіжість інформації. Крім того, включено індикатор, який відображається червоним або зеленим кольором, залежно від актуальності даних. Червоний індикатор означає, що дані можуть бути застарілими, тоді як зелений індикатор запевняє користувачів, що дані є оновленими.

На рис. 4.6 зображено повідомлення невдалої спроби встановлення групи користувача.

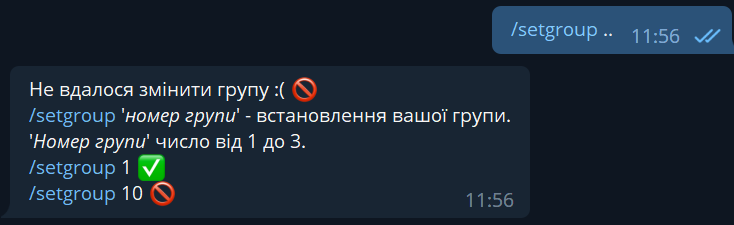


Рисунок 4.6 – Повідомлення невдалої спроби встановлення групи користувача

Це повідомлення відображається користувачеві, коли він намагається змінити свою групу, але вводить неправильний синтаксис команди.

Повідомлення повідомляє про неможливість змінити групу, надаючи стислий опис: «Не вдалося змінити групу :( ».

Щоб допомогти користувачеві правильно використовувати команду зміни групи, повідомлення містить інструкції про те, як встановити свою групу за допомогою команди «/setgroup». У ньому зазначено, що до команди слід додати параметр «номер групи», щоб вказати бажану групу.

Крім того, у повідомленні уточнюється, що «номер групи» має бути числовим значенням у діапазоні від 1 до 3. Ця інформація допомагає користувачам зрозуміти допустимий діапазон параметрів, доступних для вибору їхньої групи.

Для ілюстрації правильного використання наведено два приклади:

* /setgroup 1 – Цей приклад демонструє правильний синтаксис для встановлення групи під номером 1. Дотримуючись цього формату, користувач може успішно оновити призначення своєї групи.
* /setgroup 10 – На відміну від цього, цей приклад демонструє невірний номер групи, а саме номер 10.

Надаючи чіткі інструкції, а також приклади правильного і неправильного використання команд, повідомлення має на меті допомогти користувачам правильно змінити призначення групи, одночасно інформуючи їх про будь-які синтаксичні помилки, з якими вони могли зіткнутися.

На рис. 4.7 зображено повідомлення вдалої спроби зміни групи користувача.



Рисунок 4.7 – Повідомлення вдалої спроби зміни групи користувача

Це повідомлення відображається, щоб повідомити користувача про успішну зміну групи.

Повідомлення повідомляє про позитивний результат: «Групу успішно змінено!». Це чітке і стисле повідомлення підтверджує, що запитувана зміна групи була успішно оброблена.

Надаючи це повідомлення, повідомлення запевняє користувача, що бажану групу було успішно оновлено.

На рис. 4.8 зображено повідомлення невдалої спроби змінити час попередження.

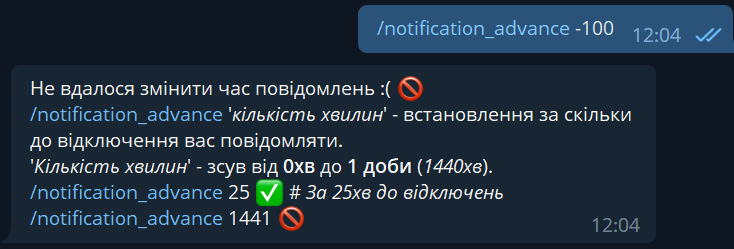


Рисунок 4.8 – Повідомлення невдалої спроби змінити час попередження

Це повідомлення відображається для інформування користувача про невдалу спробу змінити час сповіщення для попереджень, пов'язаних з відключеннями.

Повідомлення містить короткий опис невдалої спроби: «Не вдалося змінити час сповіщення :( ».

Щоб допомогти користувачеві правильно налаштувати час сповіщень, повідомлення містить інструкцію про те, як встановити бажаний час за допомогою команди «/notification\_advance». У ньому зазначено, що параметр «Кількість хвилин» має бути включений до команди, щоб вказати бажаний зсув перед відключенням для отримання сповіщень.

Крім того, повідомлення уточнює допустимий діапазон для параметра «Кількість хвилин». У ньому зазначено, що зсув може становити від 0 хвилин до 1 дня, що відповідає 1440 хвилинам.

Для ілюстрації правильного використання наведено два приклади:

* /notification\_advance 25 – демонструє правильний синтаксис для встановлення випередження сповіщення за 25 хвилин до відключення. Дотримуючись цього формату, користувач може успішно оновити час сповіщення;
* /notification\_advance 1441 – демонструє невірний ввід, де вказано зміщення на 1441 хвилину. Це перевищує максимально допустиме зміщення 1440 хвилин.

На рис. 4.9 зображено повідомлення вдалої спроби зміни часу повідомлень.



Рисунок 4.9 – Повідомлення вдалої спроби зміни часу повідомлень

Це повідомлення відображається, щоб проінформувати користувача про успішну зміну часу сповіщення.

Повідомлення повідомляє про позитивний результат: «Час успішно змінено! ». Це чітке і стисле повідомлення підтверджує, що запитувана зміна часу сповіщення була успішно оброблена.

Повідомлення запевняє користувача, що бажана зміна часу сповіщення була точно реалізована в системі.

Команда «info» має два варіанти відповіді користувачу, залежно від його запису у системі. На рис. 4.10 зображено перший приклад повідомлення команди «/info».

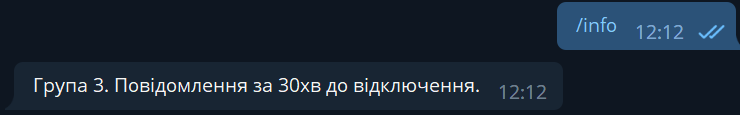


Рисунок 4.10 – Перший приклад повідомлення команди «/info»

Повідомлення команди «/info» надає відповідну інформацію про групу користувача і час сповіщення перед відключенням.

У цьому конкретному прикладі у повідомленні зазначено: «Група 3. Сповіщення за 30 хвилин до відключення». Це коротке повідомлення містить дві ключові частини інформації:

* група – вказує, що користувач належить до групи 3. Це вказує на призначену групу користувача в системі;
* час сповіщення – користувач отримає сповіщення за 30 хвилин до того, як відбудеться відключення. Це вказує на часовий інтервал, за який користувач буде попереджений перед будь-яким запланованим відключенням електроенергії.

Завдяки наданню цієї інформації за допомогою командного повідомлення «/info», користувачі можуть швидко отримати доступ до деталей про свою групову приналежність і час надходження сповіщень.

На рис. 4.11 зображено другий приклад повідомлення команди «/info».

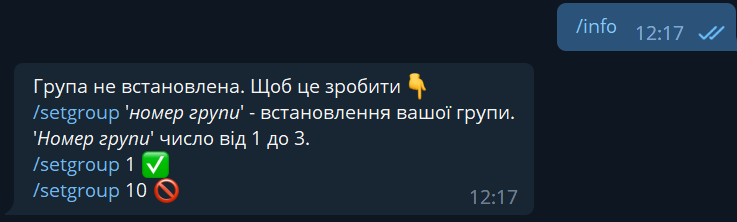


Рисунок 4.11 – Другий приклад повідомлення команди «/info»

У цьому контексті повідомлення команди «/info» відображається, якщо користувач ще не приєднався до групи. Це повідомлення має на меті підказати користувачеві, як встановити його належність до групи.

Повідомлення інформує користувача про те, що групу ще не створено, і вказує на необхідність виконати певну дію, щоб виправити це. Щоб допомогти користувачеві встановити групу, у повідомленні наведено інструкції щодо використання команди «/setgroup».

Надаючи чіткі інструкції та приклади, повідомлення допомагає користувачам зрозуміти, як правильно встановити свою групову приналежність. Воно заохочує їх використовувати команду «/setgroup», підкреслюючи при цьому допустимий діапазон номерів груп.

## 4.3 Алгоритм рішення задач

Розробка Telegram-бота вимагає впровадження добре продуманого алгоритму вирішення проблем. Цей алгоритм має бути здатним ефективно збирати та обробляти дані з надійних джерел, аналізувати їх для виявлення потенційних відключень електроенергії та генерувати своєчасні сповіщення, які можуть бути надіслані користувачам через платформу Telegram.

Цей розділ присвячений проектуванню та розробці алгоритму вирішення проблем для Telegram-бота, спрямованого на сповіщення користувачів про можливі відключення електроенергії в їхньому місті. У ньому розглядаються ключові міркування та виклики, пов'язані зі створенням алгоритму, який може ефективно і точно визначати потенційні відключення електроенергії, гарантуючи, що користувачі отримають своєчасні та релевантні сповіщення. Алгоритм охоплюватиме різні етапи, включаючи збір даних, попередню обробку, аналіз та генерацію повідомлень, кожен з яких відіграє життєво важливу роль у загальній функціональності Telegram-бота.

На рис. 4.12 зображено діаграму послідовності роботи бота.

Бот працює по безперервному циклу, що нагадує нескінченну бігову доріжку. Модуль Puller відповідає за запит оновлень з сервера Telegram. Він робить це, взаємодіючи з «BotAPI», проміжним інтерфейсом, який з'єднує бота з сирим API Telegram, який в кінцевому підсумку з'єднується з серверами.

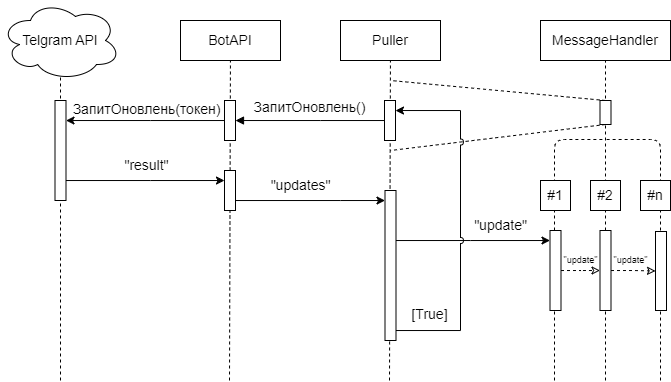


Рисунок 4.12 – Діаграма послідовності роботи бота

Bot API безпосередньо виконує запити до сервера, використовуючи технологію Long Poll. Довге опитування – це техніка, яка широко використовується у веброзробці та системах зв'язку в реальному часі, включаючи Telegram API [10]. Вона сприяє ефективному обміну даними між клієнтом і сервером, забезпечуючи оновлення в режимі реального часу без необхідності постійних клієнтських запитів.

У випадку з Telegram API тривале опитування дозволяє обмінюватися повідомленнями та сповіщеннями в реальному часі в додатку Telegram. Коли користувач взаємодіє зі своїм клієнтом Telegram, наприклад, надсилає повідомлення або приєднується до групи, сервери Telegram повинні негайно повідомити клієнта про ці події. Тривале опитування полегшує цей процес комунікації.

Після тривалого періоду очікування сервер відповідає запитуваною інформацією, яка може включати різноманітні деталі про дії різних клієнтів, такі як нові повідомлення, зміни до попередніх повідомлень або дії в поточних чатах. Сервер надає цю інформацію у вигляді JSON-файлу. Якщо ніяких дій не відбулося, сервер повертає порожній файл.

API бота обробляє ці вхідні дані і передає їх додатку для подальшої обробки повідомлення системою.

Оброблений результат запиту на оновлення від Telegram повертається до модуля Puller. Цей модуль займається подальшим розподілом оновлень від сервера до додатку, розміщуючи повідомлення в черзі відповідно до порядку їх надходження.

Для реалізації функціональності черги модуль Puller використовує вбудований модуль Queue в Python. Цей модуль пропонує безпечний для потоків підхід до реалізації черг, які дотримуються принципу First-In-First-Out (FIFO). Він надає набір класів і методів, які дозволяють декільком потокам безпечно взаємодіяти зі спільною чергою.

Безпека потоків у модулі Queue досягається завдяки використанню блокувального об'єкту. Коли потік має намір виконати операцію над чергою, він спочатку отримує блокування. Це блокування гарантує, що тільки один потік може отримати доступ до черги в будь-який момент часу. Інші потоки, які намагатимуться отримати доступ до черги, поки вона заблокована, будуть заблоковані і чекатимуть, поки блокування не буде знято.

Використання черги для передачі оновлень іншим компонентам дає кілька переваг в управлінні та ефективній обробці вхідних даних. По-перше, черга забезпечує збереження порядку оновлень, обробляючи їх у порядку надходження. Це має вирішальне значення для збереження цілісності повідомлень і точної послідовності подій.

По-друге, черга надає механізми синхронізації для обробки оновлень від декількох компонентів або потоків одночасно. Використовуючи методи синхронізації, такі як блокування або семафори, черга забезпечує безпеку потоків і запобігає конфліктам при доступі та обробці спільних оновлень.

Крім того, черга дозволяє асинхронно обробляти оновлення. По мірі надходження оновлень вони можуть бути негайно додані до черги, не чекаючи на завершення обробки попередніх оновлень. Це дозволяє іншим компонентам або потокам споживати оновлення з черги у власному темпі, сприяючи паралельній обробці та оптимізації продуктивності системи.

Користувачі черги, в якій зберігаються повідомлення від користувачів, представлені екземплярами функцій з модуля «MessageHandler». Ці екземпляри виконуються паралельно і конкурентно на окремих потоках, що дозволяє ефективно обробляти повідомлення користувачів. Кількість потоків спочатку визначається конфігурацією програми, яка враховує доступні системні ресурси і прагне досягти балансу між паралельністю і використанням ресурсів.

Використовуючи паралельну обробку, система може обробляти кілька повідомлень користувачів одночасно, покращуючи швидкість реагування та загальну продуктивність. Однак, важливо вирішувати потенційні проблеми, які виникають при одночасному та конкурентному виконанні, такі як умови гонки або конфлікти при доступі до спільних ресурсів.

Щоб забезпечити безпеку потоків і запобігти неузгодженості даних або конфліктам, слід впровадити відповідні механізми синхронізації, такі як блокування або семафори. Ці механізми контролюють доступ до спільних ресурсів, забезпечуючи скоординоване та впорядковане виконання функцій обробника повідомлень.

Обмежена кількість потоків, визначена початковою конфігурацією програми, відіграє вирішальну роль в оптимізації продуктивності системи. Це запобігає виснаженню ресурсів і дозволяє ефективно використовувати доступні системні ресурси. Точна кількість потоків може змінюватися залежно від таких факторів, як апаратні можливості хостингового середовища та вимоги робочого навантаження програми.

Як тільки обробники повідомлень починають виконувати відповідну бізнес-логіку, цикл отримання оновлень від користувачів повторюється. Цей циклічний процес забезпечує безперервний потік оновлень, дозволяючи спілкуватися в режимі реального часу і безперешкодно взаємодіяти між користувачами і системою.

На рис. 4.13 зображено схему архітектури бази даних додатку.

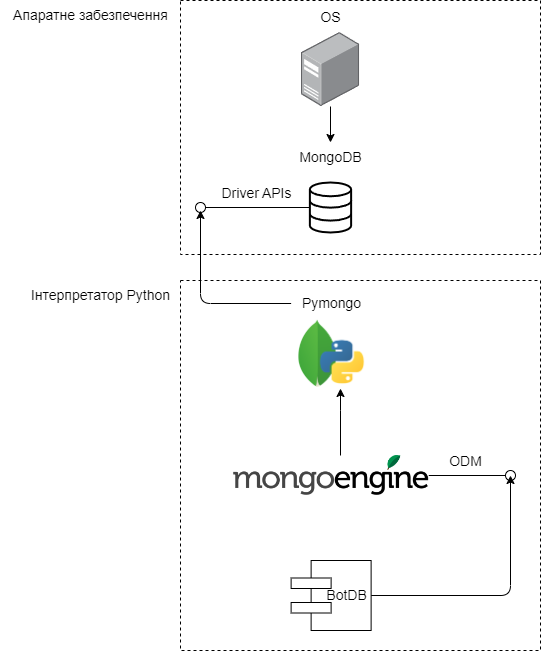


Рисунок 4.13 – Схема архітектури бази даних додатку

MongoDB надає API для Python, який дозволяє розробникам взаємодіяти з базами даних MongoDB за допомогою мови програмування Python. Цей API, який називається PyMongo, є офіційним Python-драйвером MongoDB і пропонує повний набір функцій і класів для полегшення роботи з базами даних.

PyMongo підтримує різні функції та можливості, що дозволяє легко підключатися до MongoDB, маніпулювати даними та виконувати складні запити.

Для того, щоб додаток мав змогу взаємодіяти з базою даних, використовуючи інтеграцію MongoDB спільно з ODM MongoEngine, створено модуль «BotDB», в якому реалізовані схеми документів User, DisconSchedule та функції доступу до цих документів відповідно.

На рис. 4.14 зображено фрагмент коду, що реалізує схему документа User.

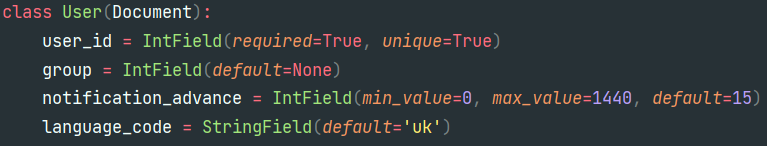


Рисунок 4.14 – Фрагмент коду реалізації схеми документа User

Код представляє собою опис схеми документа User, який має наступні поля:

* user\_id – цілочисельне поле, яке представляє ідентифікатор користувача в Telegram і може бути використане для унікальної ідентифікації користувача в додатку;
* group – Необов'язкове цілочисельне поле, яке зберігає інформацію про групу користувача. За замовчуванням користувач не належить до жодної групи;
* notification\_advance – цілочисельне поле, яке зберігає кількість хвилин до відключення електроенергії, за які користувач повинен бути сповіщений. Має мінімальне значення 0 (негайне сповіщення) і максимальне значення 1440 (1 день). За замовчуванням встановлено значення 15 хвилин;
* language\_code – рядкове поле, в якому зберігається код бажаної мови користувача, що використовується для локалізації розмови між ботом і користувачем. За замовчуванням встановлено значення 'uk' (українська).

На рис. 4.15 зображено фрагмент коду, що реалізує схему документа DisconSchedule.



Рисунок 4.15 – Фрагмент коду реалізації схеми документа DisconSchedule

Вбудований документ DisconDay представляє розклад відключень електроенергії на один день. Він має наступні поля:

* timetable – поле зі списком рядків, в якому зберігається розклад відключень електроенергії на день.

Вбудований документ DisconWeek представляє графік відключення електроенергії на тиждень, що складається з декількох днів. Він має наступне поле:

* days – поле списку вбудованих документів типу DisconDay, що зберігає розклад відключень електроенергії для кожного дня тижня.

Документ DisconSchedule представляє загальний графік відключення електроенергії для кожної групи в місті. Він має наступні поля:

* groups – поле списку вбудованих документів типу DisconWeek, в якому зберігаються графіки відключень електроенергії для кожної групи. Це поле є обов'язковим і має бути унікальним;
* last\_update – поле типу DateTimeField, яке зберігає дату і час останнього оновлення розкладу відключень електроенергії. Це поле є обов'язковим.

На рис. 4.16 зображено архітектуру додатку.

Основний модуль бота складається з декількох логічних підмодулів: Puller, MessageHandler та Browser. Його роль полягає в управлінні загальною поведінкою бота шляхом створення та ініціалізації необхідних об'єктів, створення додаткових потоків для обробки вхідних повідомлень і запуску нескінченного робочого циклу.

Підмодуль Puller відповідає за запит та оперативне розміщення отриманих з сервера оновлень у загальну чергу. Це дозволяє різним частинам програми отримувати доступ до одних і тих самих даних. Черга слугує каналом зв'язку, зберігаючи оновлення для подальшої обробки. Модуль Puller отримує дані за допомогою відкритого методу BotAPI get\_updates. Метод отримує параметр offset, що вказує на запитуване повідомлення, і параметр timeout, що визначає максимальний час очікування відповіді.

Підмодуль MessageHandler обробляє повідомлення в черзі на основі їх порядку. Він відповідає за комунікацію з користувачем, надсилання повідомлень у чат і реалізує командний функціонал для керування поведінкою бота.

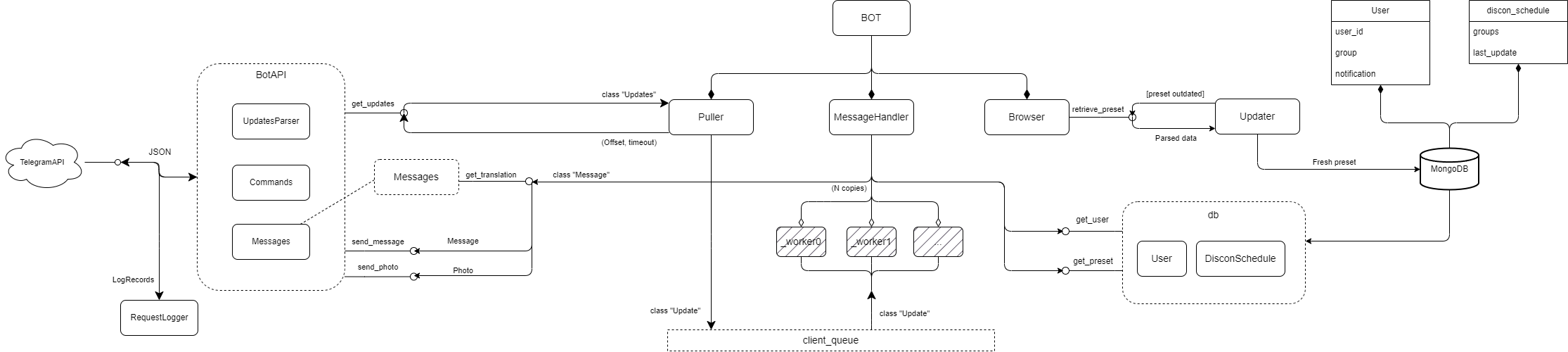


Рисунок 4.16 – Архітектура додатку

Повідомлення та зображення надсилається завдяки методам BotAPI send\_message та send\_photo відповідно. В названі модулі вбудований локалізатор, реалізований бібліотекою Gettext. Таким чином на рівні інтерфейсу між ботом та Telegram API вбудовано автоматичне перекладання для обраної мови користувача. Під час ініціалізації модуль створює кілька копій «worker» методів, кожен з яких працює в окремому потоці.

Підмодуль Browser використовує бібліотеку Selenium та вебдрайвер Chrome для реалізації функцій, пов'язаних з браузером. Цей модуль створює два екземпляри вебдрайвера. Перший екземпляр отримує доступ до вебсайту ДТЕК, аналізує вміст сторінки та отримує щотижневі дані про відключення електроенергії для кожної групи в місті. Таким чином реалізується відкритий метод retrieve\_preset, який також використовується іншими модулями, такими як Updater та db.

Другий екземпляр вебдрайвера відображає дані, отримані від першого екземпляра. Він перетворює дані в HTML-код за допомогою бібліотеки Pandas, налаштовує вигляд вікна, встановлюючи необхідну роздільну здатність і вмикаючи повноекранний режим, і робить скріншот графіка відключень, представленого у вигляді таблиці. Ці скріншоти використовуються модулем MessageHandler, що дозволяє йому надсилати зображення користувачеві за запитом.

Модуль Updater відіграє важливу роль у моніторингу своєчасності розкладу відключень. Він досягає цього, використовуючи можливості модуля Браузер для отримання та збереження оновленого розкладу. Дані, отримані з браузера, потім зберігаються в базі даних.

Модуль Db слугує інтерфейсом для взаємодії програми з базою даних. Він використовує бібліотеку mongoengine для створення статичних схем документів. Модуль відповідає за зберігання схем User і DisconSchedule і надає методи доступу до даних, такі як get\_user і get\_preset. Цей модуль дозволяє програмі зберігати важливу інформацію про користувачів, зокрема їхню групу, мову, якій надається перевага, мітки часу повідомлень і дані про перебої у роботі.

Модуль BotAPI слугує мостом між сирим API Telegram та програмою. Він реалізує методи-обгортки для Telegram, зокрема get\_updates, send\_message, send\_photo та set\_my\_commands. Метод get\_updates відповідає за отримання оновлень від користувачів бота. Він обробляє початковий JSON-документ, який повертає Telegram, виконує валідацію і на основі отриманих даних створює об'єкт Updates, який представляє собою результат виконання команди. Методи send\_message і send\_photo мають схожий функціонал. Вони конвертують текстовий параметр методу, локалізуючи його на мову користувача-одержувача перед відправкою повідомлення або фотографії. На рис. 4.17 зображено результат команди set\_my\_commands.

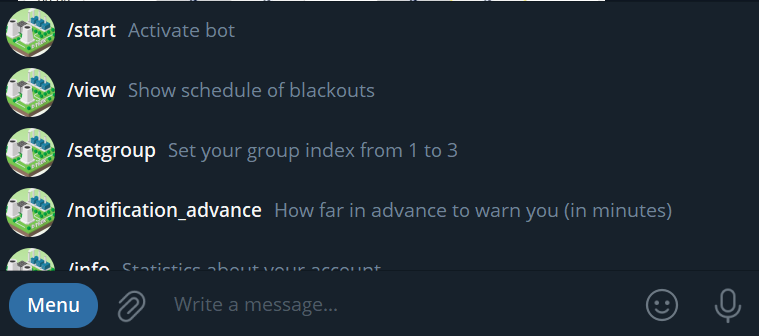


Рисунок 4.17 – Результат команди set\_my\_commands

Метод set\_my\_commands дозволяє визначити команди, які будуть відображатися безпосередньо в чаті з користувачем. Цей метод не тільки перераховує всі доступні команди для взаємодії з ботом, але й надає короткий опис функціоналу кожної команди.

Для забезпечення прозорості та відстеження зв'язку між програмою та Telegram API кожен запит логується за допомогою модуля RequestLogger та бібліотеки logging. Модуль включає в себе консольний обробник для зручного відображення логів під час розробки, а також файловий обробник, який зберігає об'єкти логу у файл під час роботи програми. Цей механізм логування забезпечує запис взаємодії програми з Telegram API для цілей налагодження та аналізу.

# ВИСНОВКИ

Дана кваліфікаційна робота присвячена розробці та впровадженню чат-бота для Telegram, спрямованого на вирішення проблеми інформування мешканців міста про стан електромереж. Метою проєкту є створення зручної та ефективної платформи, яка забезпечує оновлення та доступ до актуальної інформації під час відключень електроенергії, як у періоди стабілізації, так і в надзвичайних ситуаціях. Завдяки ретельному аналізу існуючих джерел, тематичних досліджень та використанню сучасних технологій, чат-бот був успішно розроблений та впроваджений.

Оцінка чат-бота показала його ефективність у точній та ефективній обробці запитів користувачів. Відгуки, отримані від користувачів, свідчать про високий рівень задоволеності функціоналом чат-бота.

Крім того, це дослідження є внеском у ширшу сферу розробки чат-ботів та їх застосування у сфері державних послуг. Успішне впровадження чат-бота Telegram для відключення електроенергії демонструє потенціал використання технології чат-ботів для вирішення різноманітних міських проблем та покращення обслуговування громадян. Ідеї та висновки цього дослідження мають цінність для майбутніх проєктів, спрямованих на використання систем чат-ботів для подібних цілей.

Однак важливо визнати певні обмеження та визначити сфери для подальшого вдосконалення. Ефективність чат-бота значною мірою залежить від якості та точності джерел даних, які він використовує, що може створювати проблеми, коли він стикається з неповною або застарілою інформацією. Крім того, розширення можливостей чат-бота шляхом включення таких функцій, як сповіщення про відключення через альтернативні канали та інтеграція з іншими системами розумного міста, може значно підвищити його загальну ефективність.

# ПЕРЕЛІК ПОСИЛАНЬ

1. DTEK – Wikiwand [Електронний ресурс] // – Режим доступу: <https://www.wikiwand.com/en/DTEK>
2. Планові і аварійні відключення [Електронний ресурс] // – Режим доступу: <https://www.dtek-kem.com.ua/ua/shutdowns>
3. Telegram: Contact @DTEKKyivskielectromerezhibot [Електронний ресурс] // – Режим доступу: <https://t.me/DTEKKyivskielectromerezhibot>
4. MongoDB: The Developer Data Platform [Електронний ресурс] // – Режим доступу: <https://www.mongodb.com/>
5. Telegram APIs [Електронний ресурс] // – Режим доступу: <https://core.telegram.org/>
6. CI/CD – Wikiwand [Електронний ресурс] // – Режим доступу: <https://www.wikiwand.com/en/CI/CD>
7. Relational database [Електронний ресурс] // – Режим доступу: <https://www.wikiwand.com/en/Relational_database>
8. Document-oriented database [Електронний ресурс] // – Режим доступу: <https://www.wikiwand.com/en/Document-oriented_database>
9. Git – Wikiwand [Електронний ресурс] // – Режим доступу: <https://www.wikiwand.com/en/Git>
10. What is HTTP Long Polling? [Електронний ресурс] // – Режим доступу: [https://www.pubnub.com/blog/http-long-polling](https://www.pubnub.com/blog/http-long-polling/)