

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики



ЧАТ-БОТ ДЛЯ ЖИТЛОВО-КОМУНАЛЬНОГО ГОСПОДАРСТВА

Курсова робота

за спеціальністю „Інженерія програмного забезпечення” 121

Керівник курсової роботи
кандидат технічних наук
Ковалюк Тетяна Володимирівна

Виконала студентка
Мітіньова А.С.

Київ – 2020

Календарний план виконання курсової роботи

Тема: Чат-бот для житлово-комунального господарства.

Календарний план виконання роботи:

| № п/п | Назва етапу курсової роботи | Термін виконання етапу | Примітка |
|----------|---|------------------------------|----------|
| 1. | Отримання завдання на курсову роботу. | 14.11.2020 | |
| 2. | Дослідження предметної області. | 30.12.2020 | |
| 3. | Огляд створених чат-ботів для даної області. | 05.02.2020 | |
| 4. | Визначення проблеми та постановка задачі. | 07.03.2020 | |
| 5. | Визначення функціональних вимог до чат-бота. | 10.03.2020 | |
| 6. | Проектування бази даних | 12.04.2020 | |
| 7. | Розробка програмних інтерфейсів та тестування. | 25.04.2020 | |
| 9. | Написання пояснювальної роботи. | 06.05.2020 | |
| 10. | Створення слайдів для доповіді та написання доповіді. | 07.05.2020 | |
| 11. | Аналіз отриманих результатів з керівником, написання доповіді та попередній захист курсової роботи. | 09.05.2020 | |
| 13. | Захист курсової роботи. | 18.05.2020 | |

Студентка _____ Мітіньова А. С. _____

Керівник _____ Ковалюк Т.В. _____

Зміст

| | |
|---|-----------|
| ВСТУП | 4 |
| 1 ПРЕДМЕТНА ОБЛАСТЬ “ЖИТЛОВО-КОМУНАЛЬНЕ ГОСПОДАРСТВО” | 5 |
| 1.1 Опис предметної області “Житлово-комунальне господарство” | 5 |
| 1.2 Проблеми житлово-комунального господарства в Україні | 6 |
| 2 ОГЛЯД СТВОРЕНИХ ЧАТ-БОТІВ ДЛЯ ЖКГ | 8 |
| 2.1 Продукт “МІЙ ДІМ. Чат-бот для ЖКГ” | 8 |
| 2.2 Продукт “CityBot Назар” | 12 |
| 3 ВИЗНАЧЕННЯ ПРОБЛЕМИ – ПОСТАНОВКА ЗАДАЧІ | 13 |
| 4 РОЗРОБКА ЧАТ-БОТУ | 15 |
| 4.1 Проектування та розробка бази даних | 15 |
| 4.2 Створення програмних інтерфейсів бота | 27 |
| ВИСНОВКИ | 34 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 35 |

ВСТУП

Метою цієї роботи було створення чат-бота для житлово-комунального господарства.

Першим кроком для досягнення мети стало вивчення предметної області, для якої потрібно було створити чат-бот. Потрібно було зрозуміти процеси та проблеми в житлово-комунальному господарстві. У ході вивчення предметної області “Житлово-комунальне господарство” виявилось, що вона дуже комплексна та складна, має велике поле різних проблем, які варто вирішувати.

Другим кроком стало знайомство та огляд вже створених раніше чат-ботів для житлово-комунального господарства. Було досліджено функціональні можливості, які вони надають та оцінено наскільки вони зручні для користування.

Третім кроком стало визначення конкретної проблеми, тобто постановка задачі, на якій потрібно було сфокусуватися, щоб створити якісний чат-бот.

Останнім кроком стала розробка самого чат-боту, а саме огляд технологій які підходять для реалізації застосунку, побудова ER-моделі, а також формування висновків, які можна зробити з проробленої роботи.

1 ПРЕДМЕТНА ОБЛАСТЬ “ЖИТЛОВО-КОМУНАЛЬНЕ ГОСПОДАРСТВО”

1.1 Опис предметної області “Житлово-комунальне господарство”

Житлово-комунальне господарство – це сукупність галузей, мета яких – забезпечити життя та роботу населення. Житлово-комунальне господарство відповідає за утримання житлових та допоміжних приміщень у належному стані, тобто здійснення ремонту, реставрації, санітарного контролю будівель, а також проведення каналізацій, прокладання водопровідних труб і т.д. Також ця галузь забезпечує населення такими необхідними для комфортного життя ресурсами, як: вода, тепло, газ та електроенергія.

Фактично житлово-комунальне господарство – є великою, складною, комплексною системою, до складу якої входять такі елементи:

1. Енергетичні підприємства, а саме – електростанції та теплоелектроцентралі, електричні, теплофікаційні та газові розподільні мережі, газові заводи та опалювальні котельні;
2. Сполучення населених пунктів, а саме – громадський транспорт, а саме: метрополітен, фунікулер, трамваї, тролейбуси, автобуси, таксі, а також водний транспорт. Також дороги, тротуари, мости, наземні та підземні переходи, магістралі і т.д
3. Санітарно-технічні підприємства – підприємства водовідведення та водопостачання, житлово-експлуатаційні підприємства, підприємства, які відповідають за підтримку чистоти в населених пунктах, а також пральні, бані.

У системі житлово-комунального господарства фігурують декілька сторін – виробники, споживачі та іноді посередники. Виробники – це інфраструктури, які займаються безпосереднім виробництвом необхідних ресурсів. Споживачі – це мешканці міста або деякі підприємства. Посередники – це житлово-експлуатаційні організації, які підтримують зв'язки з виробниками

(спеціалізованими підприємствами) та займаються обслуговуванням споживачів (громадян та інших підприємств), контролем за дотриманням споживачами відповідних правил взаємодії.

1.2 Проблеми житлово-комунального господарства в Україні

Так як житлово-комунальне господарство – це сукупність різних галузей, то зрозуміло, що там багато своїх ідентичних проблем, які властиві лише цим конкретним галузям. Але все таки, можна загально описати проблеми, які притаманні більшості галузей, а саме:

1. Підприємства житлово-комунальних галузей, які надають послуги, поагно керуються та контролюються;
2. Житлово-комунальні галузі мають низьку фінансову спроможність через, наприклад:
 - Низьку платіжну дисципліну споживачів (мешканців, підприємств).
 - Низькі тарифи на роботи, що виконуються безпосередньо ОСББ (Об'єднання співвласників багатоквартирного будинку) за статтею “Утримання будинку та прибудинковій території”, а саме: поточний та капітальний ремонт будівель та внутрішньобудинкових мереж та обладнання.
3. Зношені внутрішньобудинкові мережі, тобто системи теплозабезпечення, енергозабезпечення, водопостачання та інші механізми (ліфти тощо) приміщень є застарілими та, як резкльтат, мають невисоку ефективність.
4. Незахищеність споживачів перед постачальниками комунальних послуг;
5. Недосконала та нефективна претензійно-позивна робота позивачів, судової та виконавчої державної служб;
6. Недостатньо висока кваліфікація управителів багатоквартирних будинків;
7. Недосконала взаємодія жильців та комунікації між керуючими компаніями, ОСББ та мешканцями або підприємствами.

Отже, можна зробити висновок, що житлово-комунальне господарство містить мікс різних проблем, про які потрібно думати та вирішувати.

2 ОГЛЯД СТВОРЕНИХ ЧАТ-БОТІВ ДЛЯ ЖКГ

2.1 Продукт “МІЙ ДІМ. Чат-бот для ЖКГ”

Продукт “МІЙ ДІМ. Чат-бот для ЖКГ” – це один з продуктів лінійки продуктів з програмного комплексу “МІЙ ДІМ”, до якого входять такі продукти:

1. “МІЙ ДІМ. Розрахунок комунальних послуг”;
2. “МІЙ ДІМ. Бухгалтерія ЖКГ та побудинковий облік”;
3. “МІЙ ДІМ. Єдине вікно”;
4. “МІЙ ДІМ. Розрахунок комунальних послуг (білінг)”;
5. “МІЙ ДІМ. Особистий кабінет”

Віртуальний співрозмовник (чат-бот) “МІЙ ДІМ. Чат-бот для ЖКГ” – це канал взаємодії між керуючими компаніями, об'єднанням співвласників багатоквартирного будинку (ОСББ) та мешканцями будинку. Цей чат-бот націлений на вирішення проблеми комунікації між керуючими компаніями, ОСББ та жителями.

Функціональні можливості чат-бота “МІЙ ДІМ. Чат-бот для ЖКГ”:

1. Робота застосунку 24/7 не залежно від вихідних та свят;
2. Стрічка новин, у якій користувачі будуть інформуватися про тарифи, послуги та нарахування;
3. Інформування користувачів про їх заборгованості з метою пришвидшення їх погашення;
4. Можливість отримувати квитанції в електронному вигляді на email або прямо в чат;
5. Можливість відправити показники приладів (наприклад: електролічильників) онлайн;

6. Подача заявок щодо проблемних ситуацій у користувачів з можливістю прикласти фотографію онлайн.

Даний чат-бот інтегровано в Facebook. Взаємодію з чат-ботом користувач починає з натискання кнопки “Розпочати”, після цього користувач бачить привітання та дві опції: “Почати спілкування” та “Керування розсилкою”. Опція “Керування розсилкою” дозволяє підписатися або відписатися від розсилки новин та квитанцій. Опція “Почати спілкування” надає користувачу всі його особові рахунки, які закріплені за обліковим записом користувача. Результат виконання опції “Почати спілкування” продемонстрований на Рисунку 1.

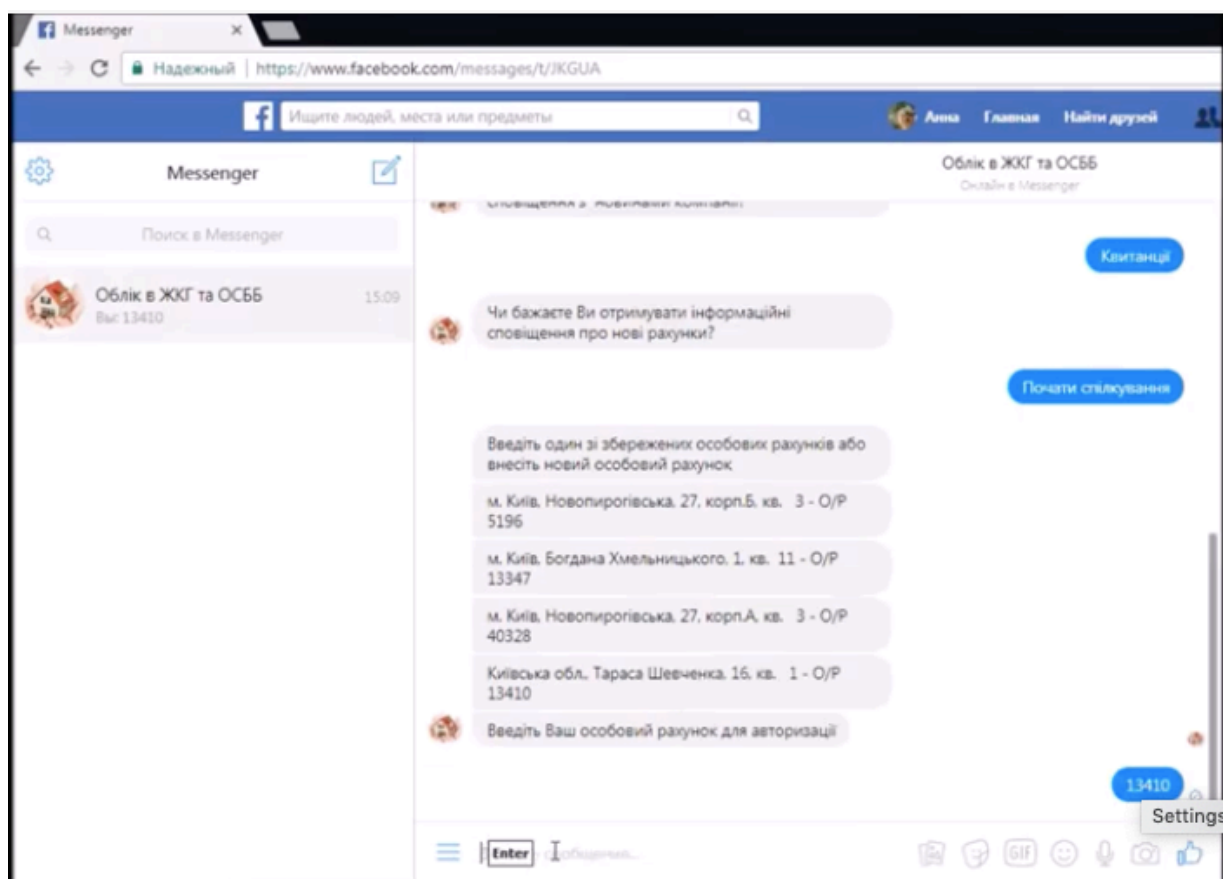


Рисунок 1 – результат виконання опції “Почати спілкування” (особові рахунки користувача)

Для авторизації користувач повинен ввести один з номерів особових рахунків зі свого облікового запису, виданих ботом. Після авторизації користувач отримує такі опції: “Розрахунки”, “Сплатити” та “Лічильники”.

Опція “Розрахунки”, у свою чергу надає такі опції: “Стан рахунку”, “Отримати квитанцію” та “Дізнатись про тарифи”. Опція “Стан рахунку” дозволяє дізнатися стан рахунку користувача, наприклад: виводить суму боргу, якщо він є та пропонує його сплатити.

Опція “Отримати квитанцію” надає можливість отримати квитанцію двома способами на email або прямо в чат (можна обрати варіант, для першого, потрібно ввести свою електронну пошту в бот). Після того, як квитанція відправлена, бот сповіщає користувача про це в повідомленні. Якщо квитанцію отримано в чаті, то бот пропонує сплатити її.

Опція “Дізнатися про тарифи” надає користувачу інформацію про тарифи в локальному повідомленні Рисунок 3.

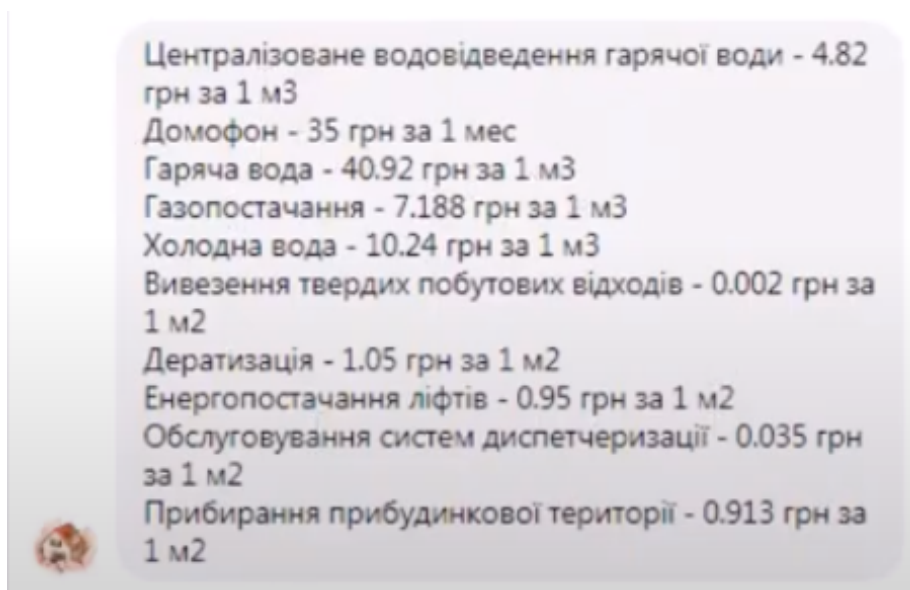


Рисунок 2 – приклад інформації, яку надає бот про тарифи

Кожне повідомлення бота супроводжується опціями, які дозволяють потрапити на опцію “Головна” та інші. Така можливість спрощує навігацію користувачу.

Опція “Лічильники” надає користувачу можливість ввести показники своїх лічильників. Після натискання користувачем опції “Лічильники” бот надає користувачу вибір комунальних послуг, для яких можна ввести показники. Після того, як користувач обирає комунальну послугу, для якої потрібно ввести

дані лічильника, бот надає інформацію про останній раз введені користувачем дані. Також бот містить перевірку даних введених користувачем, наприклад: дані не можуть бути меншими певного показника.

Опція “Сплатити” дозволяє користувачу сплатити рахунок за допомогою однієї з популярних платіжних систем (Приват, iPay та Portmone).

Дуже імовірно, що усі дані, які чат-бот отримує від користувача, записуються в базу даних для подальших розрахунків, які вже виконуються іншим окремим компонентом (продуктом) з програмного комплексу від “МІЙ ДІМ”.

Загалом, можна сказати, що “МІЙ ДІМ. Чат-бот для ЖКГ” надає зручні функції для користувача, але навігація, яку надає даний чат-бот, є не дуже зручною, бо:

1. У даному чат-боті немає єдиного меню, яке надає усі функції та постійно супроводжує користувача під час взаємодії з чат-ботом. Натомість бот розміщує пункти навігації для користувача в повідомленнях-відповідях на запити користувача і навігація “розкидана” по різних повідомленнях;
2. Бот не надає ніяких пунктів навігації для користувача, які б перекидували його на батьківське повідомлення з сусідніми в ієрархії пунктами навігації, якщо користувач обирає один з пунктів навігації у цьому батьківському повідомленні.

Як наслідок, через таку недосконалу навігацію, користувач може легко заплутатися в потоці повідомлень від бота та не знайти функцію, яка йому була потрібна.

2.2 Продукт “CityBot Назар”

Даний чат-бот – це також канал зв’язку між мешканцями, ОСББ та керуючими організаціями, але він потрібен для того, щоб мешканці могли отримувати інформацію щодо ремонтів та усунення аварій за їх адресою. Даний чат-бот інтегровано в Viber. Бот доступний лише для таких міст: Тернопіль, Дрогобич, Маріуполь та Луцьк.

Бот надає достатньо зручний інтерфейс, у якому для користувача є зручне меню. Бот дозволяє переглянути “Додвідку”, надає можливість виставити собі оцінку. У боті реалізована функція, яка дозволяє користувачам надати показники своїх лічильників – Рисунок. 3 та Рисунок. 4

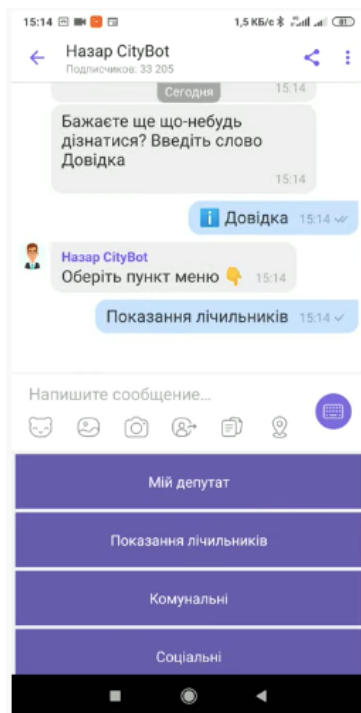


Рисунок 3 – продукт “CityBot Назар”

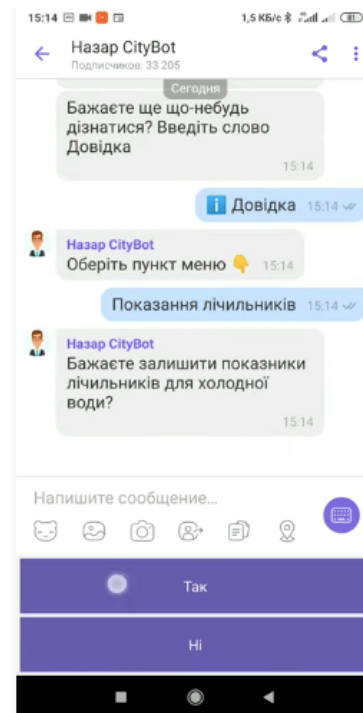


Рисунок 4 – продукт “CityBot Назар”

Загалом, бот склав приємне враження, бо у нього досить продумана навігація та присутні допоміжні інструкції для користувача, але він містить більш обмежений функціонал порівняно з продуктом “МІЙ ДІМ. Чат-бот для ЖКГ”.

3 ВИЗНАЧЕННЯ ПРОБЛЕМИ – ПОСТАНОВКА ЗАДАЧІ

Найближчою та націкавішою проблемою стала проблема взаємодії мешканців та керуючих компаній (посередників, виробників), ОСББ. Було вирішено створити чат-бот, який буде реалізувати функціонал продукта “МІЙ ДІМ. Чат-бот для ЖКГ”, бо він є серцевиною, для будь-якого подібного бота. Але бот реалізований у даній роботі буде мати певні відмінності та вдосконалення:

1. Буде вдосконалено навігацію для користувача, тобто буде створено єдине меню з усіма функціями бота, яке буде супроводжувати користувача протягом усієї його роботи з ботом. У якості зразка можна вважати навігацію в продукті “CityBot Назар”.
2. Буде здійснено спробу зробити навігацію для користувача більш інтуїтивною та зручною.
3. Бот буде реалізовано в Telegram, бо він надає готове, зручне API для взаємодії з користувачем. Можливо, Telegram є не дуже надійним для реалізації такої системи, але для створення прототипу застосунку цілком підійде;
4. Буде додано новий тип сповіщень, а саме “Борги”. Сповіщення цього типу мають містити в собі нагадування, щодо несплачених користувачем боргів.
5. Буде спроектовано свою базу даних з усіма сутностями та зв'язками;
6. Буде реалізовано свою архітектуру застосунку;
7. Для таких функцій як оплата та формування квитанції буде зроблено симуляцію. Тобто оплата буде фіксуватися лише в базі даних, а гроші зніматися не будуть. Квитанція буде відправлятися окремим повідомленням в чат, але для спрощення, вона не буде містити деякі дані реальної квитанції;
8. У користувача буде можливість самостійно додати або видалити лічильник;

9. Такі дані як: рахунки, послуги (електроенергія, холодна вода, тощо), контракти будуть братися з бази даних і їх буде завчасно згенеровано.

4 РОЗРОБКА ЧАТ-БОТУ

4.1 Проектування та розробка бази даних

4.1.1 Обраний інструментарій

Було вирішено, що база даних для чат-боту буде реляційною, а для керування нею буде обрано вільну систему керування реляційними базами даних – MySQL.

4.1.2 ER-модель

Для реалізації чат-бота необхідно було створити базу даних і тому створити ER-модель, яка продемонстрована на Рисунку 6. Для реалізації сповіщень потрібні ще додаткові зв'язки для сутності “Обліковий запис” та “Особовий рахунок”. Їх зображено на Рисунку 5.

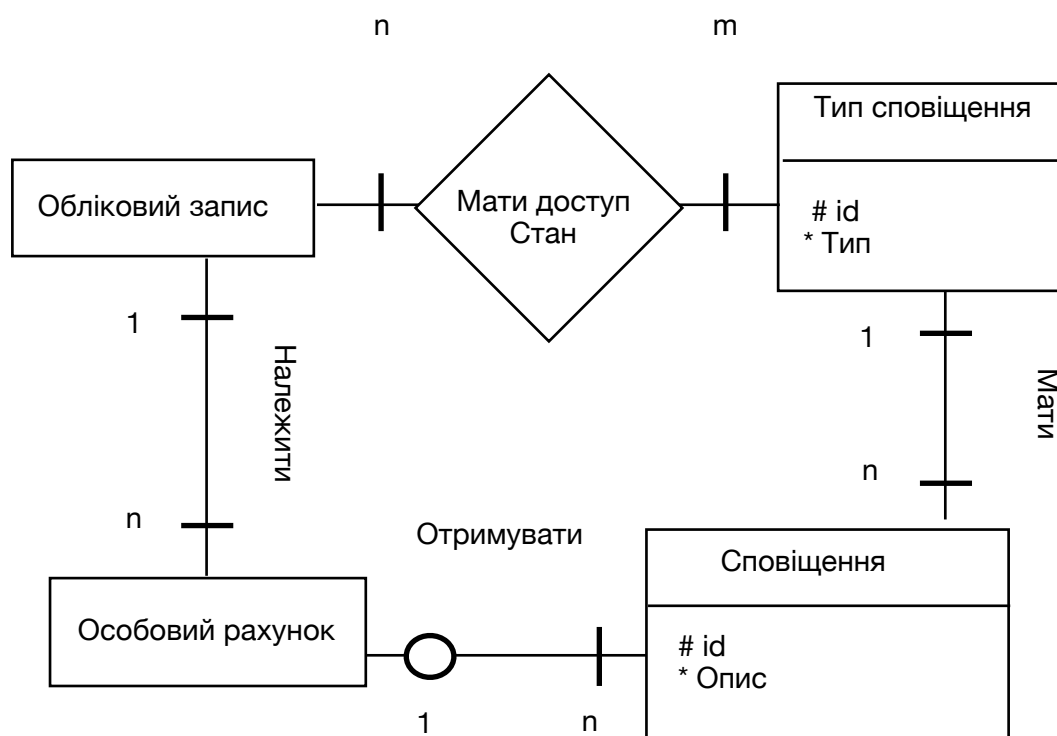


Рисунок 5 – друга частина ER-моделі

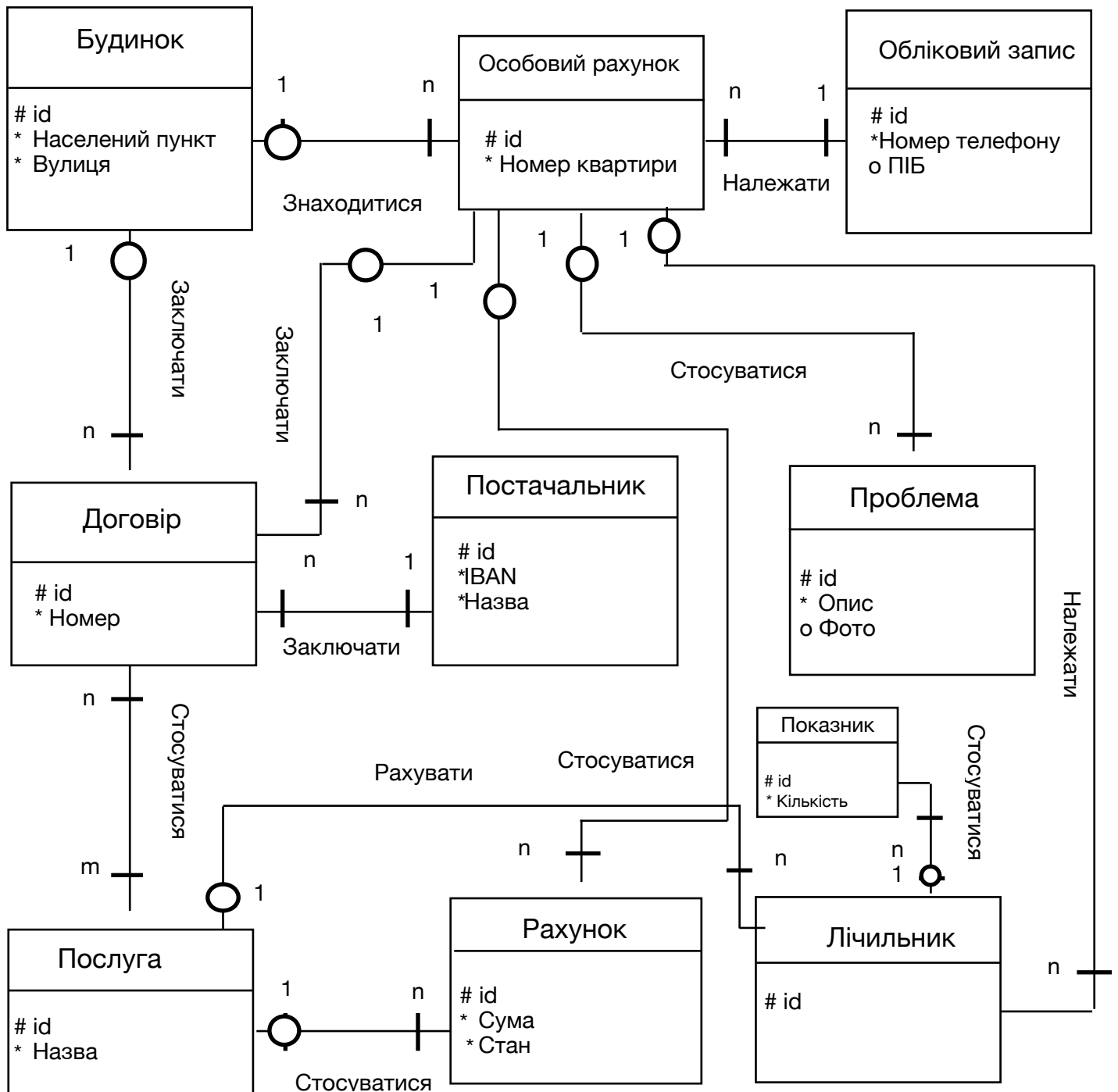


Рисунок 6 – перша частина ER-моделі

4.1.2.1 Опис сутностей

1. Сутність “Будинок”. Містить обов’язкові атрибути: “Населений пункт” та “Вулиця”. Їх буде винесено в окремі сутності, щоб не було дублювання, на ER-моделі сутності не зображені на рисунку. Сутність “Будинок” створена для того, щоб значення атрибутів не дублювалися в базі даних та для того, щоб можна було реалізувати функцію сповіщення користувачів про зміни в будинку.
2. Сутність “Особовий рахунок”. Містить обов’язковий атрибут “Номер квартири”.
3. Сутність “Обліковий запис”. Містить обов’язковий атрибут “Номер телефону” та необов’язковий – “ПІБ” (Прізвище, Ім’я, По батькові).
4. Сутність “Договір”. Потенційний ключ – може бути “Номер договору”, хоча краще, щоб це був id згенерований за допомогою MySQL, бо він може бути унікальним лише в певних межах.
5. Сутність “Постачальник”. Містить обов’язковий атрибут “IBAN” (International Bank Account Number) – це МФО + ЄДРПОУ, завдяки якому буде зрозуміло, куди будуть поступати гроші. Обов’язковий атрибут “Назва”, наприклад “Київенерго”, тощо.
6. Сутність “Проблема”. Містить обов’язковий атрибут – опис і необов’язковий атрибут – “Фото”.
7. Сутність “Послуга”. Містить обов’язковий атрибут – назва, наприклад: “Холодна вода”, “Горяча вода”, тощо.
8. Сутність “Рахунок”. Містить обов’язкові атрибути: “Сума” – вартість, яку має сплатити користувач за користування тою чи іншою послугою та “Стан” – може містити true або false, тобто може бути сплаченим або ні.
9. Сутність “Лічильник”. Містить лише потенційний ключ.

10. Сутність “Показник”. Містить лише обов’язковий атрибут “Кількість” – це показник лічильника. Також у цієї сутності буде обов’язковий атрибут – “Дата” – це буде зберігати дату, у яку користувач погасив рахунок. Екземпляр даної сутності буде автоматично створюватися кожний місяць.

Майже у всіх сутностях пристуній атрибут id, який є потенційним ключем, є унікальним та генерується за допомогою MySQL.

4.1.2.2 Опис зв’язків

1. Зв’язок “Знаходиться” між екземпляром сутності “Будинок” та екземпляром – “Особовий рахунок” є обов’язковим зв’язком зі сторони сутності “Особовий рахунок” та необов’язковим зі сторони “Будинок”, типу “один до багатьох”. Потрібний для того, щоб визначити якому будинку належить особовий рахунок та навпаки. Зв’язок з боку екземпляру сутності “Особовий рахунок” – є обов’язковим, тому що особовий рахунок не може існувати без будинку. Зв’язок з боку екземпляру сутності “Будинок” є необов’язковим, тому що цей екземпляр може бути створено раніше і без особового рахунку, далі буде пояснення щодо цього.
2. Зв’язок “Належати” між екземпляром сутності “Обліковий запис” та екземпляром – “Особовий рахунок” є обов’язковим з обох сторін, типу “один до багатьох”. Зв’язок має тип один до багатьох потрібний, щоб реалізувати можливість для користувача обслуговувати декілька особових рахунків. Зв’язок є обов’язковим, тому що ці екземпляри не можуть існувати один без одного – це не має сенсу.
3. Зв’язок “Заключати” між екземпляром сутності “Будинок” та екземпляром – “Договір” є обов’язковим з обох сторін, типу “один до багатьох”. Зв’язок має тип один до багатьох, тому що один будинок може мати декілька договорів з постачальниками, а договір може належати лише одному договору. Зв’язок є обов’язковим, бо якщо будинок немає ніяких договорів, то немає сенсу його тримати в базі даних.

4. Зв'язок “Заключати” між екземпляром сутності “Особовий рахунок” та екземпляром – “Договір” є необов'язковим зі сторони екземпляру “Особовий рахунок” та обов'язковим зі сторони – “Договір”, типу “один до багатьох”. Зв'язок потрібен, тому що користувач (мешканець) може заключати договір з постачальником напряму. Зв'язок необов'язковий зі сторони екземпляру “Особовий рахунок”, тому що мешканець необов'язково заключає договір напряму. Причина того, що зв'язок один до багатьох є аналогічна причині описаній в пункті 3.
5. Зв'язок “Стосуватися” між екземпляром сутності “Особовий рахунок” та екземпляром – “Рахунок” є необов'язковим з боку екземпляру – “Особовий рахунок” та обов'язковим з боку – “Рахунок”, типу “один до багатьох”. Зв'язок необов'язковий з боку – “Особовий рахунок”, бо на момент реєстрації, користувач може не мати рахунків. Зв'язок обов'язковий з боку – “Рахунок”, бо рахунок не може існувати без особового рахунку.
6. Зв'язок “Стосуватися” між екземпляром сутності “Особовий рахунок” та екземпляром – “Проблема” є необов'язковим з боку екземпляру – “Особовий рахунок” та обов'язковий з боку – “Проблема”, типу “один до багатьох”. Причини чому зв'язок саме такий аналогічні до причин описаних у пункті 5. Зв'язок потрібний для реалізації функції, за допомогою якої користувач зможе повідомити про проблеми.
7. Зв'язок “Належати” між екземпляром сутності “Особовий рахунок” та екземпляром – “Лічильник” є необов'язковим з боку – “Особовий рахунок” та обов'язковий з боку – “Лічильник”, типу “один до багатьох”. Зв'язок є необов'язковий з боку “Особовий рахунок”, бо мешканець є необов'язковим, бо мешканець може не мати лічильників і обов'язковим з боку – “Лічильник”, бо лічильник не може існувати в базі даних без користувача (мешканця). Зв'язок потрібний для реалізації функції, за допомогою якої у користувач може збирати показники лічильників від користувачів.

8. Зв'язок “Заклювати” між екземпляром сутності “Постачальник” та екземпляром – “Договір” є обов'язковим з обох боків, типу “один до багатьох”. Зв'язок є обов'язковим з обох боків, бо екземпляри не можуть існувати один без одного і не має особливого сенсу зберігати в базі даних постачальників, з якими ніхто не заключає договорів. Зв'язок є один до багатьох, бо постачальник може мати безліч договорів, а от один договір може відноситися лише до одного постачальника. Зв'язок потрібний для реалізації функції оплати послуги.
9. Зв'язок “Стосуватися” між екземпляром сутності “Договір” та екземпляром – “Послуга” є обов'язковим, типу “багато до багатьох”. Зв'язок є обов'язковим з обох боків, бо договір завжди обов'язково стосується певних послуг і немає сенсу без них і послуга, у свою чергу, не може бути надана без договору. Зв'язок є типу “багато до багатьох”, бо для однієї послуги може бути заключено багато різних договорів (для різних будинків) та в один договір може бути заключений для декількох послуг.
10. Зв'язок “Стосуватися” між екземпляром сутності “Послуга” та екземпляром – “Рахунок” є необов'язковим з боку екземпляра – “Послуга” та обов'язковим з боку – “Рахунок”, типу “один до багатьох”. Зв'язок є необов'язковим з боку екземпляру сутності “Послуга”, бо для послуги може бути не створено рахунків. Зв'язок є обов'язковим з боку екземпляру сутності “Рахунок”, бо рахунок не може існувати в базі даних без послуги.
11. Зв'язок “Стосуватися” між екземпляром сутності “Лічильник” та екземпляром – “Показник” є необов'язковим з боку – “Лічильник” та обов'язковим з боку – “Показник”, типу “один до багатьох”. Зв'язок є необов'язковим з боку екземпляра “Лічильник”, бо лічильник може не мати показників, а з боку екземпляру типу сутності “Показник” – є обов'язковим, бо показник не може існувати без лічильника. Зв'язок потрібен для реалізації функції, за допомогою якої користувач зможе надавати прказники лічильника.

4.1.3 Реляційна модель даних

Скорочення:

PK – Primary Key;

FK – Foreign Key.

| Settlements (для екземплярів типу сутності “Населений пункт”, була описана в 4.1.1.2) | | | | | | | |
|---|------|-----------------|---------|----------------------|-----------|-----------|------------------------------|
| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
| 1 | PK | SettlementId | integer | NOT NULL | - | - | - |
| 2 | - | Settlement Name | varchar | NOT NULL | - | - | Макс. довжина – 70 символів. |

| Streets (для екземплярів типу сутності “Вулиця”, була описана в 4.1.1.2) | | | | | | | |
|--|------|---------------|---------|----------------------|-----------|-----------|------------------------------|
| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
| 1 | PK | StreetId | integer | NOT NULL | - | - | - |
| 2 | - | StreetName | varchar | NOT NULL | - | - | Макс. довжина – 70 символів. |

| Buildings (для екземплярів типу сутності “Будівля”) | | | | | | | |
|---|------|---------------|---------|----------------------|-----------|-----------|---|
| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
| 1 | PK | BuildingId | integer | NOT NULL | - | - | - |
| 2 | FK | SettlementId | integer | NOT NULL | CASCADE | CASCADE | Якщо населений пункт видаляється/оновлюється, то і будинок видаляється/оновлюється. |

Buildings (для екземплярів типу сутності “Будівля”)

| | | | | | | | |
|----------|----|----------|---------|----------|---------|---------|--|
| 3 | FK | StreetId | integer | NOT NULL | CASCADE | CASCADE | Якщо вулиця видаляється/оновлюється, то і будинок видаляється/оновлюється. |
|----------|----|----------|---------|----------|---------|---------|--|

Accounts (для екземплярів типу сутності “Обліковий запис”)

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|----------|------|---------------|---------|----------------------|-----------|-----------|-------------------------------|
| 1 | PK | AccountId | integer | NOT NULL | - | - | - |
| 2 | - | Phone | varchar | NOT NULL | - | - | Макс. довжина – 20 символів. |
| 3 | - | UserName | varchar | NOT NULL | - | - | Макс. довжина – 120 символів. |

PersonalAccounts (для екземплярів типу сутності “Особовий рахунок”)

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|----------|------|-------------------|---------|----------------------|-----------|-----------|--|
| 1 | PK | PersonalAccountId | integer | NOT NULL | - | - | - |
| 2 | - | Flat | integer | NOT NULL | - | - | - |
| 3 | FK | BuildingId | integer | NOT NULL | CASCADE | CASCADE | Якщо будинок видаляється/оновлюється, то і особовий рахунок видаляється/оновлюється. |

PersonalAccounts (для екземплярів типу сутності “Особовий рахунок”)

| | | | | | | | |
|---|----|-----------|---------|----------|---------|---------|---|
| 4 | FK | AccountId | integer | NOT NULL | CASCADE | CASCADE | Якщо будинок видаляється/оновлюється, то і обліковий запис видаляється/оновлюється. |
|---|----|-----------|---------|----------|---------|---------|---|

Providers (для екземплярів типу сутності “Постачальник”)

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|---|------|---------------|---------|----------------------|-----------|-----------|-------------------------------|
| 1 | PK | ProviderId | integer | NOT NULL | - | - | - |
| 2 | - | IBAN | varchar | NOT NULL | - | - | Макс. довжина – 36 символів. |
| 3 | - | ProviderName | varchar | NOT NULL | - | - | Макс. довжина – 100 символів. |

Contracts (для екземплярів типу сутності “Договір”)

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|---|------|----------------|---------|----------------------|-----------|-----------|---|
| 1 | PK | ContractId | integer | NOT NULL | - | - | - |
| 2 | - | ContractNumber | varchar | NOT NULL | - | - | Макс. довжина – 100 символів. |
| 3 | FK | BuildingId | integer | NULL | CASCADE | CASCADE | Якщо будинок видаляється/оновлюється, то і договір з ним видаляється/оновлюється. |

Contracts (для екземплярів типу сутності "Договір")

| | | | | | | | |
|---|----|-------------------|---------|----------|---------|---------|--|
| 4 | FK | PersonalAccountId | integer | NULL | CASCADE | CASCADE | Якщо особовий рахунок видаляється/оновлюється, то і договір з ним видаляється/оновлюється. |
| 5 | FK | ProviderId | integer | NOT NULL | CASCADE | CASCADE | Якщо постачальник видаляється/оновлюється, то і договір з ним видаляється/оновлюється. |

Services (для екземплярів типу сутності "Послуга")

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|---|------|---------------|---------|----------------------|-----------|-----------|------------------------------|
| 1 | PK | ServiceId | integer | NOT NULL | - | - | - |
| 2 | - | ServiceId | varchar | NOT NULL | - | - | Макс. довжина – 70 символів. |

ServicesContracts (для реалізації зв'язку типу "багато до багатьох" між екземплярами сутностей типу "Договір" та – "Послуга")

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|---|------|---------------|---------|----------------------|-----------|-----------|------------------------|
| 1 | FK | ServiceId | integer | NOT NULL | CASCADE | CASCADE | - |
| 2 | FK | ContractId | integer | NOT NULL | CASCADE | CASCADE | - |
| 3 | PK | - | - | NOT NULL | - | - | ServiceId + ContractId |

Problems (для екземплярів типу сутності “Проблема”)

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|---|------|--------------------|---------|----------------------|-----------|-----------|--|
| 1 | PK | ProblemId | integer | NOT NULL | - | - | - |
| 2 | - | ProblemDescription | varchar | NOT NULL | - | - | Макс. довжина – 500 символів. |
| 3 | - | Photo | blob | NULL | - | - | - |
| 4 | FK | PersonalAccountId | integer | NOT NULL | CASCADE | CASCADE | Якщо особовий рахунок видаляється/оновлюється, то і його проблема видаляється/оновлюється. |

Bills (для екземплярів типу сутності “Рахунок”)

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|---|------|-------------------|---------|----------------------|-----------|-----------|---|
| 1 | PK | BillId | integer | NOT NULL | - | - | - |
| 2 | - | BillSum | decimal | NOT NULL | - | - | - |
| 3 | - | State | bool | NOT NULL | - | - | - |
| 4 | - | Created | date | - | - | - | - |
| 5 | FK | PersonalAccountId | integer | NOT NULL | CASCADE | CASCADE | Якщо особовий рахунок видаляється/оновлюється, то і його рахунок видаляється/оновлюється. |

Bills (для екземплярів типу сутності “Рахунок”)

| | | | | | | | |
|----------|----|-----------|---------|----------|---------|---------|---|
| 6 | FK | ServiceId | integer | NOT NULL | CASCADE | CASCADE | Якщо послуга видаляється/оновлюється, то і рахунок для неї видаляється/оновлюється. |
|----------|----|-----------|---------|----------|---------|---------|---|

Counters (для екземплярів типу сутності “Лічильник”)

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|----------|------|-------------------|---------|----------------------|-----------|-----------|--|
| 1 | PK | CounterId | integer | NOT NULL | - | - | - |
| 2 | FK | PersonalAccountId | integer | NOT NULL | CASCADE | CASCADE | Якщо особовий рахунок видаляється/оновлюється, то лічильник видаляється/оновлюється. |

Indicators (для екземплярів типу сутності “Показник”)

| № | Ключ | Ім'я атрибуту | Тип | Can be NULL/NOT NULL | ON UPDATE | ON DELETE | Пояснення |
|----------|------|---------------|---------|----------------------|-----------|-----------|--|
| 1 | PK | IndicatorId | integer | NOT NULL | - | - | - |
| 2 | - | Amount | integer | NOT NULL | - | - | - |
| 3 | FK | CounterId | integer | NOT NULL | CASCADE | CASCADE | Якщо лічильник видаляється/оновлюється, то і показник видаляється/оновлюється. |

У ході розробки було додані деякі атрибути до певних сутностей, наприклад, дату створення, тощо.

4.2 Створення програмних інтерфейсів бота

4.2.1 Обраний інструментарій

Для реалізації програмних інтерфейсів бота було обрано мову Java та створено проект. Для збірки проекту було використано фреймворк Apache Maven. Для взаємодії з користувачем було використано Telegram Bot API. Для під'єднання до бази даних було обрано MySQL Connector/J driver.

4.2.2 Логіка навігації

Логіку навігації можна уявити у вигляді дерева. Навігацію було реалізовано спираючись на Рисунок 7, де зображено “Дерево навігації”. Було створено клас `NavigationNode`, об’єкти якого містили посилання на відповідні елементи та є кнопками у меню. Також `NavigationNode` містить текст, який потрібно показати на кнопці та допоміжний текст, який варто показати після її натискання. Насправді, якщо називати точно у термінах Telegram, то кнопка – це клавіша, а меню – то клавіатура.

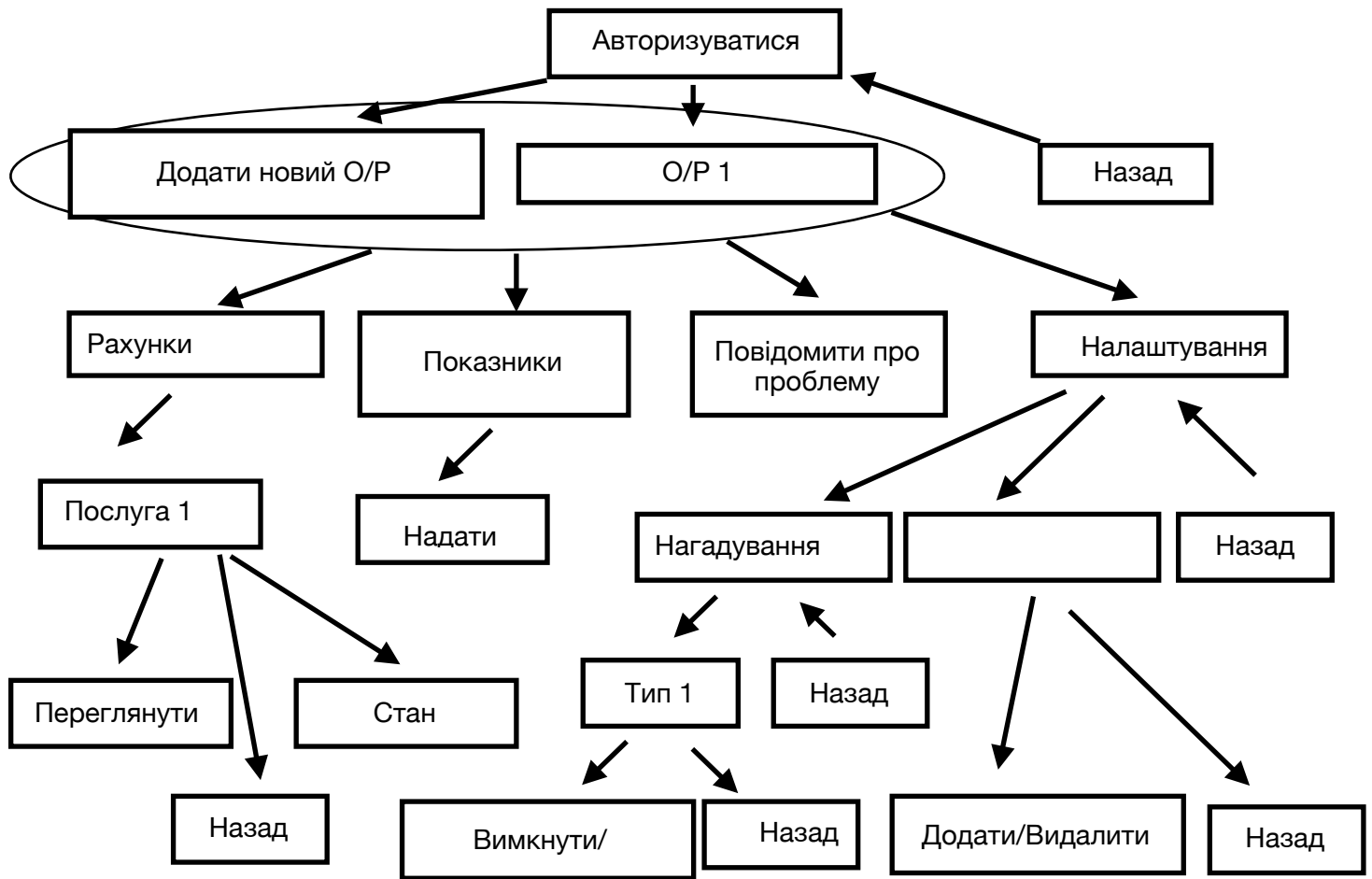


Рисунок 7 – “Дерево навігації”

4.2.3 Реалізовані функції

4.2.3.1 Авторизація

Авторизація складається з декількох етапів. На першому етапі потрібно створити або дістати з бази даних обліковий запис користувача – це було зроблено за допомогою унікального ідентифікатора чату, який можна дістати з будь-якого повідомлення користувача. Такий ідентифікатор дістається з першого повідомлення для бота і він також є унікальним в межах бази даних. Якщо в базі даних не міститься обліковго запису з такими ідентифікатором, то він створюється, а якщо є – то дістається. На другому етапі потрібно, щоб користувач обрав особовий рахунок, з яким бажає працювати. Якщо в користувача вже є такі особові рахунки, то вони виводяться, якщо ні – то він бачить лише кнопку “Додати новий особовий рахунок”. У межах даної роботи не була реалізована функція “Додати новий особовий рахунок”. Вигляд бота під час авторизації зображено на Рисунку 8 та Рисунку 9.

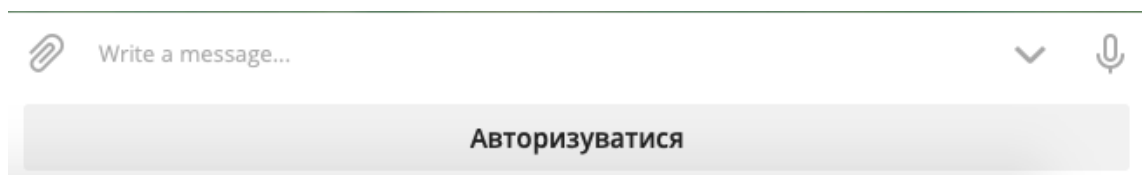


Рисунок 8 – Авторизація, етап 1

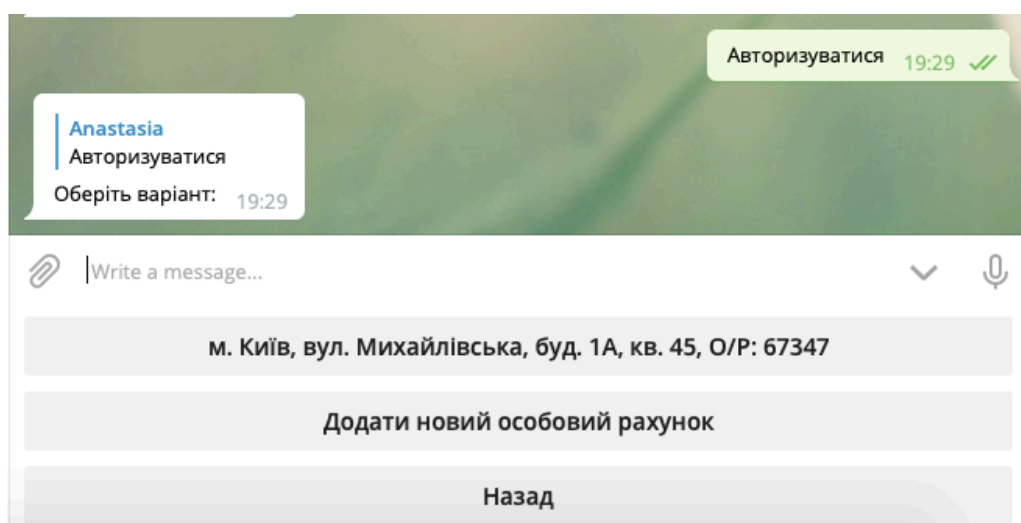


Рисунок 9 – Авторизація, етап 2

4.2.3.2 Можливість додати або видалити лічильник

Була реалізована функція додати або видалити лічильник. Для цього потрібно обрати пункт меню “Лічильники” з пункту “Налаштування”, зображений на Рисунку 10. Додати лічильник для певної послуги можна, якщо його немає, а видалити – якщо він є. Для реалізації такої функції здійснюються запити до бази даних. На Рисунку 11 зображено послуги, для яких можна додати лічильник, а на Рисунку 12 – зображено пункти меню, після кліку на відповідну послугу.

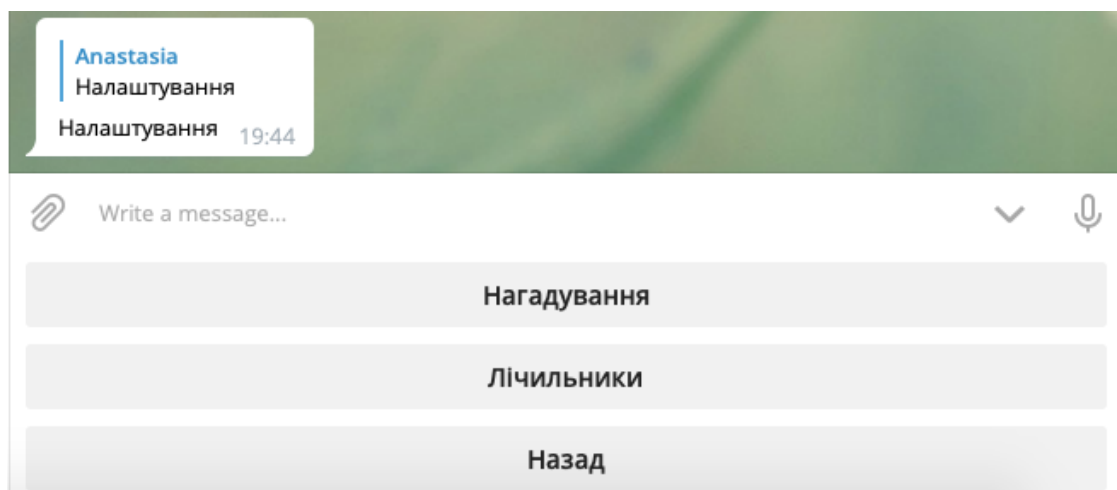


Рисунок 10 – Налаштування для бота

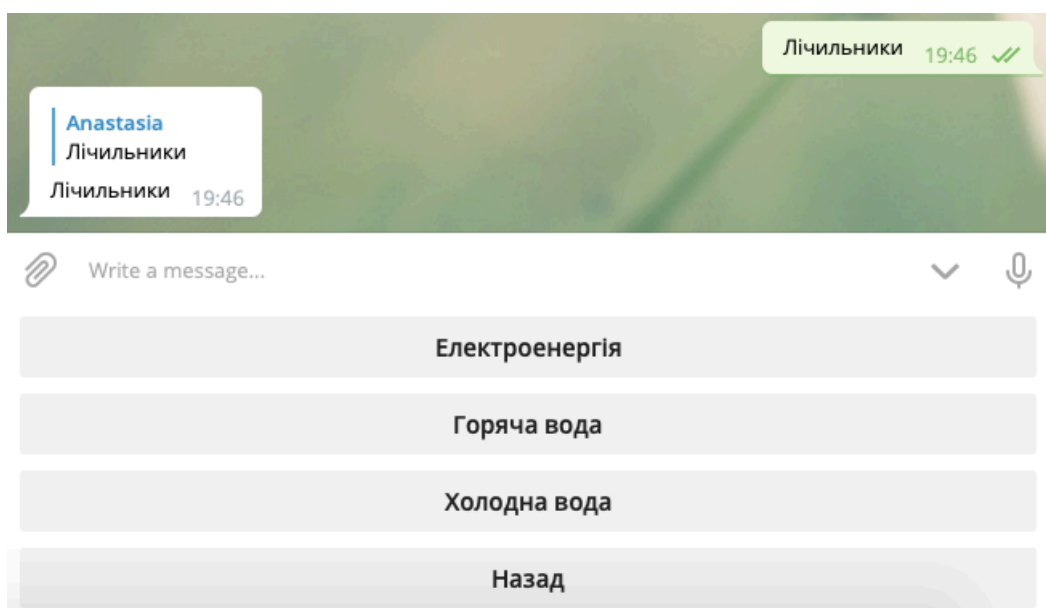


Рисунок 11 – Послуги, для яких можна додати лічильник

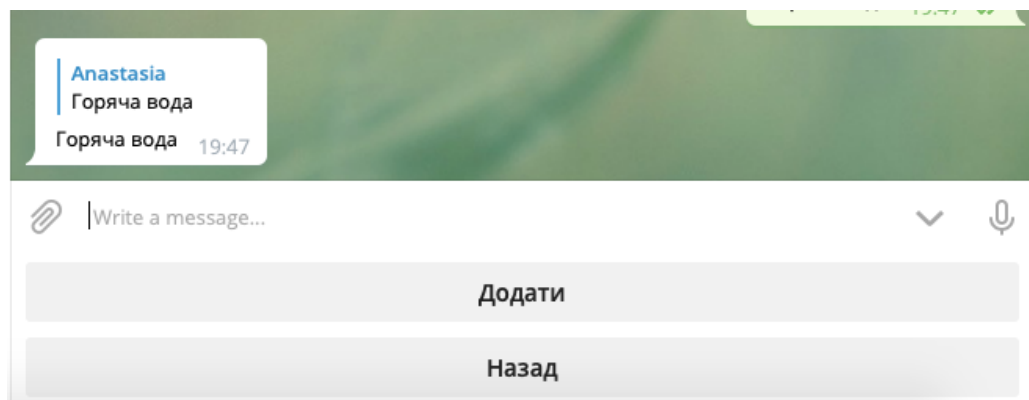


Рисунок 12 –Можливість додати лічильник після вибору відповідної псолуги

4.2.3.3 Можливість надати показники лічильника

Була реалізована функція для надання показників користувача. Для цього потрібно обрати кнопку “Показники лічильника” після входу під особовим рахунком. Якщо не було додано жодного лічильника до особового рахунку, за допомогою якого користувач авторизувався до системи, то кнопка доступна, але після її натискання бот викидує повідомлення з пропозицією додати лічильники в налаштуваннях (Рисунок 13). Якщо у користувача додано хоча б один лічильник для деякої послуги, то вона виводиться після натискання кнопки “Показники лічильника”, так само виводяться інші послуги, для яких додано лічильник. Послуга, для якої є лічильник, виводить відповідну кнопку, за допомогою якої можна надати показники (Рисунок 14).

Показники для лічильника можна надавати багато разів, але база даних буде зберігати один показник для одної дати. Взагалі можна зробити так, щоб можна було зберігати один показник для одного місяця та року, тобто база даних не має містити декілька показників за один місяць.

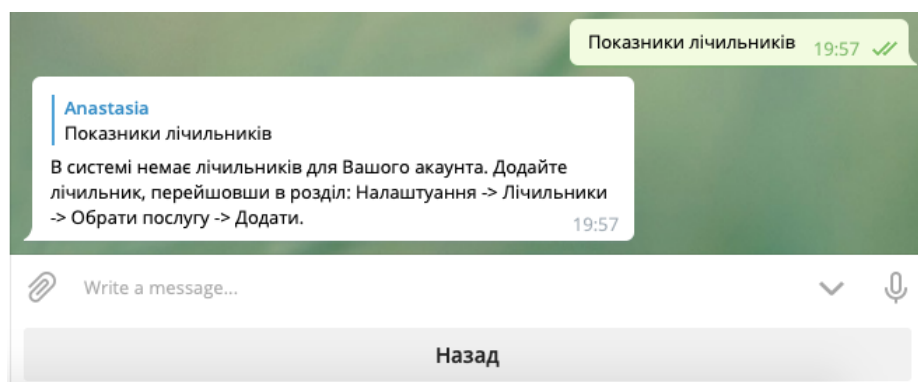


Рисунок 13 – Повідомлення від бота, якщо користувач не має жодного лічильника

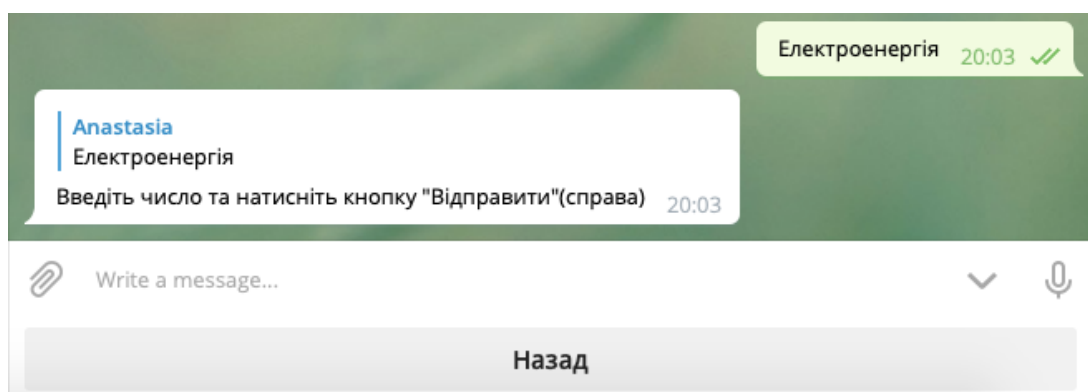


Рисунок 14 – Можливість надати показники для лічильника

4.2.3.4 Можливість сплатити рахунок

Була реалізована можливість сплатити рахунок. Для цього користувачу потрібно обрати пункт меню “Рахунки”. Після цього користувач зможе обрати послугу, рахунок якої хоче сплатити. Рахунки повинні надходити в базу даних зі сторнонних ресурсів у певні проміжки часу, тому вони не завжди доступні. Якщо рахунок не доступний для сплати, то виводиться кнопка “Немає”, натиснувши на яку користувач отримує відповідне повідомлення. Також рахунок може мати статус – сплачено або ні, для цього замість кнопки “Немає” виводяться відповідні кнопки. Щоб сплатити рахунок, потрібно натиснути відповідну кнопку і зробити це можна, якщо рахунок не сплачено. Після того, як рахунок буде сплачено – він має замінятися на новий, який буде приходити зі сторнонних ресурсів. Вигляд бота після вибору послуги, рахунок для якої потрібно сплатити, зображено на Рисунку 15.

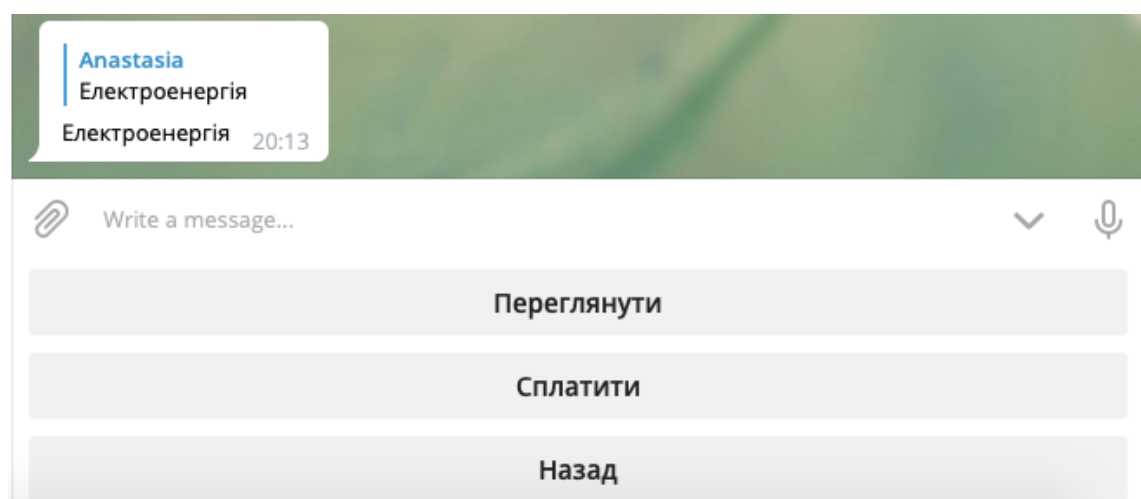


Рисунок 15 – Можливість сплатити рахунок (вигляд бота)

4.2.3.5 Можливість отримати квитанцію

Користувач може отримати квитанцію в чат, натиснувши на кнопку “Переглянути” для потрібного рахунку (Рисунок 16).

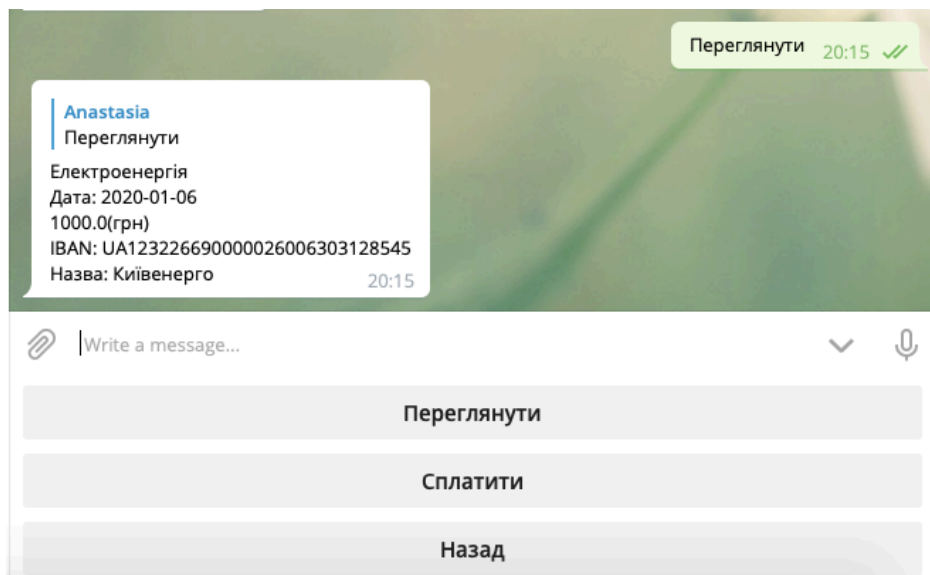


Рисунок 16 – Квитанція

4.2.3.6 Можливість налаштувати сповіщення

Було реалізовано функцію для налаштування користувача. У користувача в налаштуваннях є можливість вимкнути або увімкнути певний тип сповіщень (Рисунок 17).

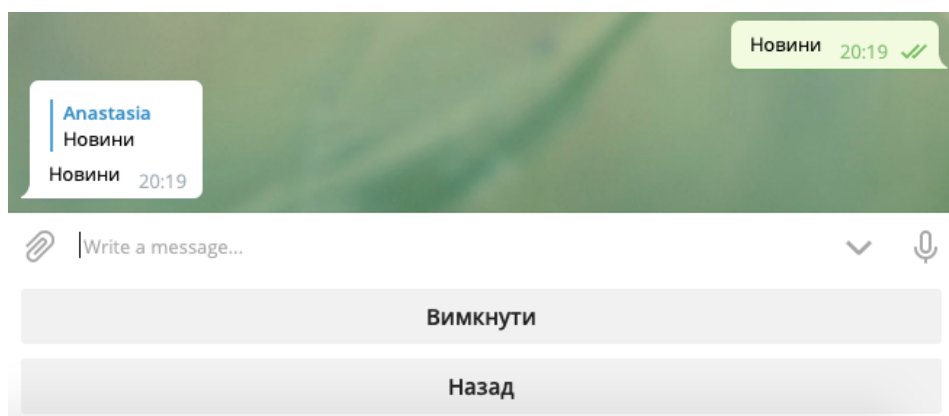


Рисунок 17 – Можливість увімкнути/вимкнути сповіщення

ВИСНОВКИ

У даній роботі було досліджено деякі чат-боти для житлово-комунального господарства, а також створено власний чат-бот із вдосконаленою системою навігації. Також було спроектовано базу даних для чат-бота.

З проробленої роботи можна зробити висновок, що Telegram надає досить прості та зручні інструменти для створення власних ботів, але створення якісного чат-боту для житлово-комунального господарства виявилось досить нетривіальною задачею, бо потрібно було продумати зручну навігацію для користувача та ретельно продумати ER-модель для бази даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://cutt.ly/кypycOZ>
2. <http://eprints.kname.edu.ua/38765/1/13-15.pdf>
3. https://protocol.ua/ua/gitlovo_budivelni_kooperativi_buti_chi_ne_buti/
4. <https://jkg.com.ua/>
5. <https://cutt.ly/Opyx0m>
6. <https://citybot.pro/citybot-nazar-mariupol>
7. <https://github.com/rubenlagus/TelegramBots>
8. <https://core.telegram.org/bots>