

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ФРАНКА**

Факультет

прикладної математики та інформатики

(повне найменування назва факультету)

**Кафедра дискретного аналізу та інтелектуальних систем**

(повна назва кафедри)

**ДИПЛОМНА РОБОТА**

на тему:

**“Розробка WEB застосунку - Додаток для оплати комунальних платежів на базі CRM - системи Salesforce”**

Студента IV курсу, групи ПМі-45,  
спеціальності 122 – комп’ютерні  
науки

Казимира М.М.

(прізвище та ініціали)

Керівник: доцент Олійник Р.М.

(посада, прізвище та ініціали)

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

## ЗМІСТ

Анотація .....	3
Вступ .....	4
Розділ 1. Опис та вимоги до застосунку .....	5
1.1 Опис системи .....	5
1.2 User stories .....	6
1. User stories. Головна сторінка. ....	6
2. User stories: Роль адміністратора.....	6
3. User stories: Роль регіонального менеджера.....	6
4. User stories: Роль користувача. ....	7
Розділ 2. Технології використані в ході розробки. ....	8
2.1. Salesforce. Основні поняття. ....	8
2.2. Середовище розробки. Developer Organization. ....	8
2.3. Об'єкти, записи, поля, зв'язки між об'єктами. Data Modeling. ....	10
2.4. Стек технологій. ....	11
Visualforce:.....	12
Apex: .....	12
SOQL: .....	12
Salesforce Sites:.....	13
Batch Apex: .....	14
Apex Scheduler: .....	14
Telegram: .....	15
Telegram-Bot: .....	15
Telegram Bot API:.....	15
Webhook: .....	16
Flow. Flow Builder: .....	16
Apex REST Resource: .....	17
LWC (Lightning Web Components): .....	17
Розділ 3. Проектування та реалізація системи .....	18
3.1 Проектування WEB - застосунку.....	18
3.2 Проектування та імплементація бази даних .....	18
3.3 Головна сторінка.....	19
3.4 Реєстрація .....	20
3.5 Авторизація.....	21
3.6 Сторінка користувача.....	21
3.7 Сторінка регіонального менеджера .....	24
3.8 Оптимізація функціоналу. Виконання дій згідно графіку. ....	24
3.9 Інтеграція Salesforce з third-party сервісами. Інтеграція Salesforce &	

<b>Telegram. ....</b>	<b>25</b>
<b>3.10 Lightning Web Components.....</b>	<b>28</b>
<b>3.11 Автоматизація процесів.....</b>	<b>30</b>
<b>ВИСНОВКИ .....</b>	<b>33</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>34</b>

## **Анотація**

Дипломна робота присвячена розробці системи для внесення показників лічильників та оплати комунальних платежів. У сьогоденні певна частка людей продовжує сплачувати комунальні платежі у спеціально відведених для цього місцях, наприклад, у найближчому відділенні пошти, що не дуже зручно.

Мета даної дипломної роботи: розробка та створення WEB - застосунку для внесення показників лічильників та оплати комунальних платежів на основі CRM - системи Salesforce націленого на вирішення цієї загальної проблеми. Також застосунок буде розроблено для дослідження CRM - системи Salesforce як способу створити власний продукт у поєднанні з даною платформою.

Постановка задачі: кінцевим результатом даної роботи є створений WEB - застосунок для оновлення показників лічильників та оплати комунальних платежів у поєднанні з CRM - платформою Salesforce, що задовольняє всі потреби користувачів даного застосунку.

## Вступ

У теперішній час, коли світ та людство стрімко розвиваються, з'являються нові можливості для оптимізації людського життя. В сфері інформаційних технологій теж є чимало способів для вирішення тих чи інших проблем повсякденного життя людини.

Якщо Ви живете у населеному пункті, у будинку чи квартирі, Ви точно знаєте, що за надання таких послуг житлово-комунальних господарств, як газ, вода чи електроенергія, потрібно сплачувати гроші. Іншими словами, кожного місяця Ви або член Вашої родини сплачує комунальні послуги. Ще декілька років тому, для цього потрібно було вистояти велику чергу у найближчому відділенні пошти. Але хіба це актуально у двадцять першому столітті? Відповідь очевидна - ні. Саме тому були розроблені сервіси для оплати комунальних послуг онлайн. Тобто тепер можна оплатити всі комунальні платежі всього лиш у пару кліків на Вашому смартфоні чи комп'ютері. Це значно економить Ваш час та береже Ваші нервові клітини.

Отже, можна сказати, що інформаційні технології значно спрощують деякі з наших повсякденних дій. Однак, цього б не відбувалось, якби ці технології не розвивались з кожним днем. Однією з доволі швидко прогресуючих галузей інформаційних технологій стали CRM - системи.

Оскільки це порівняно нова практика, важливо дослідити, як за допомогою CRM - системи можна оптимізувати процес взяття та зберігання даних від користувачів у великих масштабах та чи справді CRM - системи більш зручні для такого типу задач.

## **Розділ 1. Опис та вимоги до застосунку**

### **1.1 Опис системи**

Даний WEB - застосунок користувач зможе використовувати як зручне середовище для того, щоб вчасно вносити показники лічильників та оплачувати суму, яка вираховується з цих показників.

1. Система включає в собі три ролі користувачів:
  - Системний адміністратор
  - Регіональний менеджер
  - Користувач застосунку
2. Кожен новий користувач, при реєстрації повинен обрати регіон, в якому він проживає.
3. При успішній реєстрації, користувачу прийде електронний лист електронну пошту, яку він вказав при реєстрації;
4. Кожен регіон повинен мати свої власні коефіцієнти для підрахунку комунальних платежів для кожного сервісу, а саме: води, газу та електроенергії.
5. Кожен регіон має закріпленого за ним менеджера.
6. Користувач має змогу сплатити комунальні платежі повністю або частково. Якщо до кінця місяця користувач не сплатить усю суму, при настанні нового місяця залишок, який він мав сплатити, автоматично перейде у розділ “Борг”.
7. З першого по п’ятий день кожного нового місяця, користувач повинен вносити нові показники лічильників. Якщо цього не буде зроблено, то на шостий день користувачу буде надісланий електронний лист з проханням внести нові показники лічильників. Якщо ж це прохання було проігноровано, то на десятий день система автоматично виставить нові показники лічильників.
8. Коли користувач сплачує комунальні платежі, йому буде надісланий електронний лист про успішну або не успішну оплату. Неуспішна оплата - коли в користувача на балансі недостатньо

коштів.

9. Якщо персональний борг користувача перевищує заданий ліміт, система автоматично надішле електронний лист регіональному менеджеру, до якого закріплений даний користувач.

## 1.2 User stories

Задля спрощення формувань вимог до системи, було використано написання user stories. У розробці програмного забезпечення та управлінні продуктами user stories – це неофіційний, природний опис однієї або декількох особливостей програмної системи. Історії користувачів часто пишуться з точки зору кінцевого користувача або користувача системи.

### 1. User stories. Головна сторінка.

Серед вимог до головної сторінки наступні:

- незареєстрований користувач має отримати запрошення зареєструватися або увійти у систему, коли намагається отримати доступ до можливостей, дозволених лише зареєстрованим користувачам;
- незареєстрований користувач повинен мати можливість зареєструватися, використовуючи адресу електронної пошти;
- зареєстрований користувач має мати можливість увійти до системи, використовуючи електронну пошту та пароль.

### 2. User stories: Роль адміністратора.

Серед вимог до ролі адміністратора наступні:

- адміністратор має повний доступ до організації;
- адміністратор може додати або видалити регіонального менеджера;

### 3. User stories: Роль регіонального менеджера.

Серед вимог до ролі регіонального менеджера наступні:

- менеджер може переглядати список користувачів, які проживають у його регіоні та їхню історію платежів;
- менеджер може змінювати регіональні коефіцієнти для таких

сервісів, як: газ, вода та електроенергія;

- менеджер може змінювати місячну оплату для кожного користувача;

#### **4. User stories: Роль користувача.**

Серед вимог до ролі користувача наступні:

- користувач може вводити нові показники лічильників;
- користувач може оплатити комунальні платежі, повністю або частково;
- користувач може погасити свій борг, якщо у нього достатньо грошей на балансі;
- користувач може змінювати регіон, в якому він проживає;
- користувач може переглядати історію своїх платежів;
- користувач може скontaktуватися з своїм безпосереднім менеджером через електронний лист;



## **Розділ 2. Технології використані в ході розробки.**

### **2.1. Salesforce. Основні поняття.**

Salesforce — це платформа управління взаємовідносинами з клієнтами (CRM) №1 у світі. Компанія Salesforce була заснована в 1999 році колишнім виконавчим директором Oracle Марком Беніоффом. Головна ідея створення — це побудова доступного програмного забезпечення і впровадження його повністю онлайн в якості сервісу.

Варто сказати, що Salesforce уже давно вийшла за рамки CRM - системи. Salesforce дозволяє створювати та розгортати індивідуальні рішення, автоматизувувати бізнес-процеси, інтегруватися із зовнішніми додатками. Більшість світових компаній є клієнтами компанії Salesforce і використовують цю платформу як рішення для своїх бізнес-потреб. Серед них: Adidas, AWS, Canon, Philips, Toyota та багато інших.

### **2.2. Середовище розробки. Developer Organization.**

Одразу хочу підкреслити простий початок роботи з даною системою. І це, на мою думку, одна з найбільших переваг Salesforce. Не потрібно встановлювати жодних програм, немає ніяких вимог до обладнання. Розміщення Salesforce платформи в хмарному середовищі надає доступ до інформації в режимі реального часу, тобто користуватися платформою можна будь-де і будь-коли.

Для роботи з Salesforce потрібно мати лише доступ до інтернету. Все, що потрібно, — це перейти в браузері на сайт, ввести свій логін і пароль. Тепер Ви можете створювати власні функціонали або користуватися стандартними, уже створеними Salesforce.

Кожен має можливість створити безкоштовну організацію, так звану **Salesforce Developer Edition**, для навчання, практикування чи тестування. Це фактично повнофункціональне середовище для розробки, яке можна також використовувати для створення власних продуктів. Як вона виглядає, можна побачити на рисунку 1.

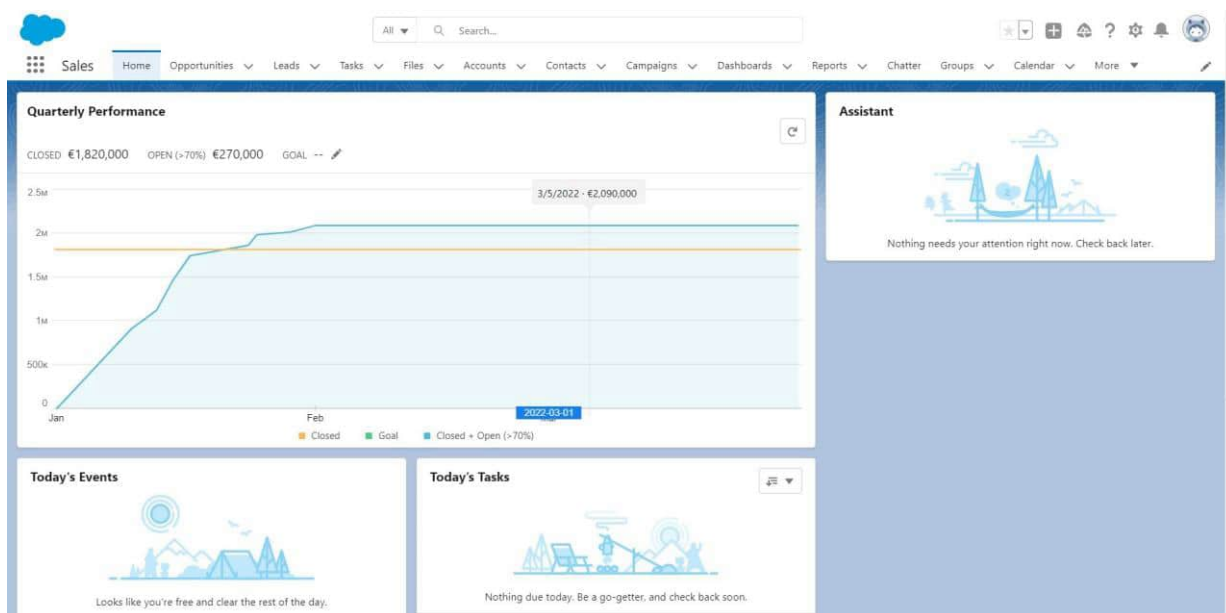


Рисунок 2.1. Вигляд домашньої сторінки в організації.

На рисунку 1 показана домашня сторінка організації. Тут можна переглянути записи об'єктів, а також графічне відображення потрібної інформації.

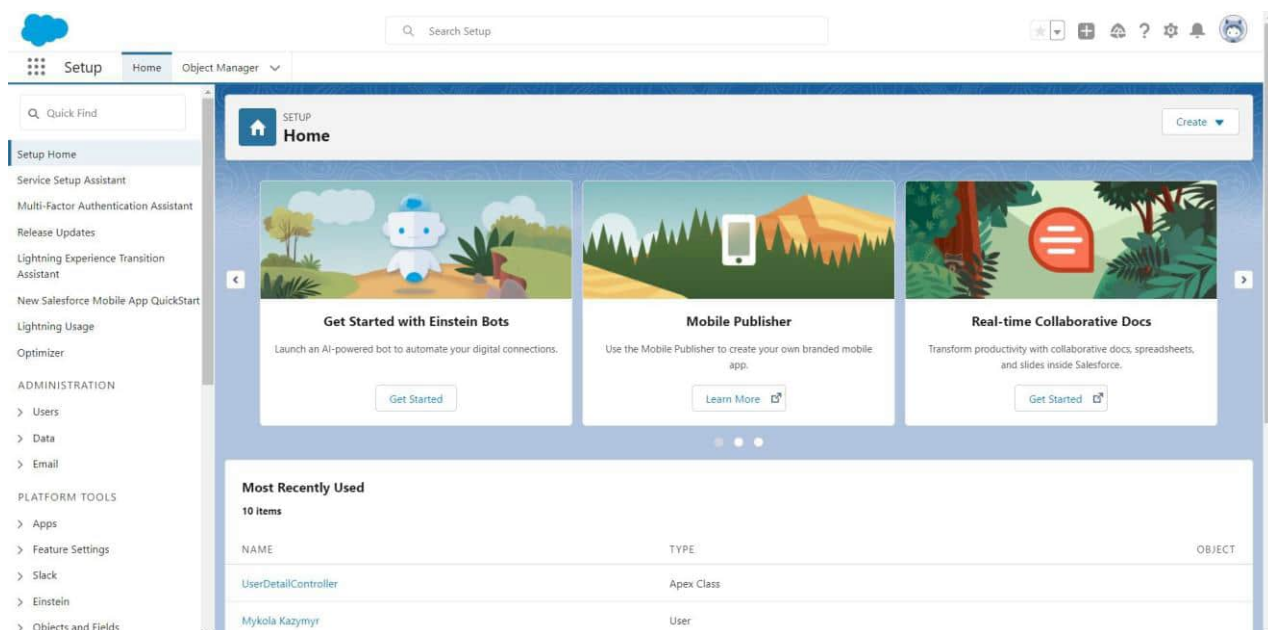


Рисунок 2.2. Вигляд сторінки налаштувань в організації.

На рисунку 2 зображена сторінка налаштувань в організації. Варто зауважити, що CRM - платформа Salesforce надає багато готових рішень, які можна використовувати, а також за допомогою декларативних засобів, не написавши жодного рядку коду, можна створити власний, справді хороший

продукт. Саме у розділі “Setup” можна створювати нові об’єкти, поля в цих об’єктах та налаштувати зв’язки між об’єктами.

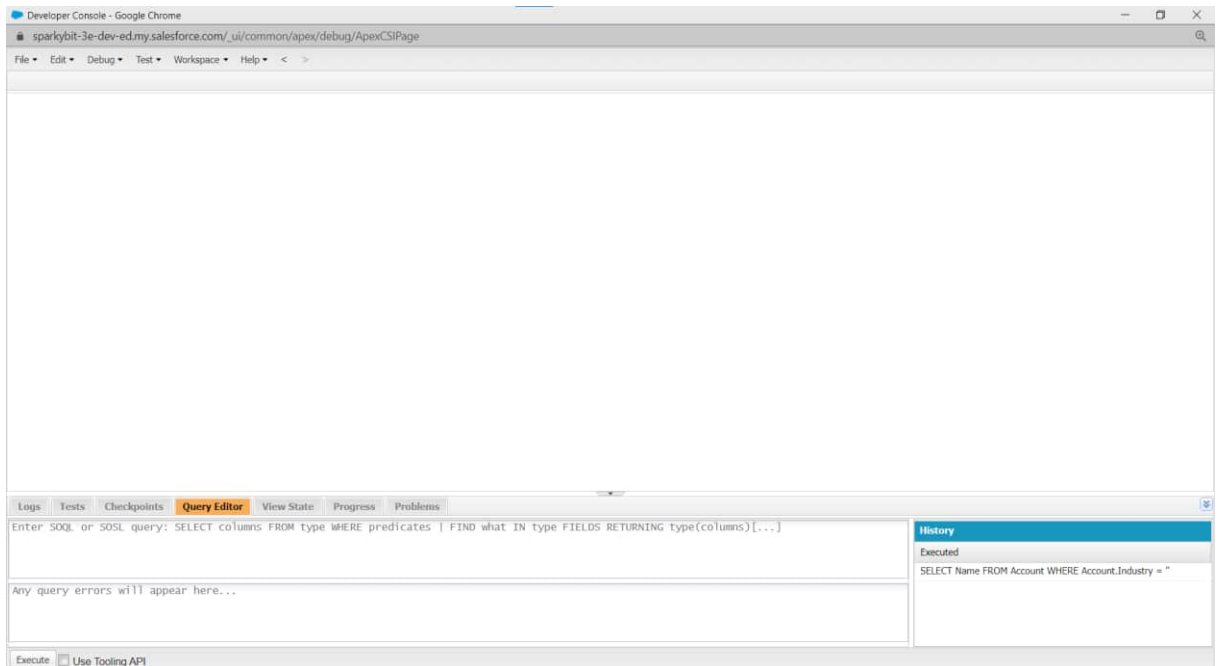


Рисунок 2.3. Вигляд консолі розробника

На рисунку 3 можна побачити, як виглядає консоль розробника ( Developer Console ). Консоль розробника — це інтегроване середовище розробки з набором інструментів, які можна використовувати для створення, налагодження та тестування програм у вашій організації Salesforce. Консоль розробника містить набір корисних інструментів, наприклад, з неї напряду можна написати запит до бази даних. Відповідно, консоль розробника застосовується в тих випадках, коли потрібно створити індивідуальне вирішення певної задачі, якого за допомогою декларативних засобів зробити неможливо або неефективно.

### **2.3. Об’єкти, записи, поля, зв’язки між об’єктами. Data Modeling.**

Salesforce зберігає всі дані користувачів в об’єкти та записи. Якщо провести аналогію до звичайної реляційної бази даних (для прикладу, MySQL), об’єктами називають таблиці в базі даних, а записом - один рядок даних в таблиці. Відповідно поля - це стовпці у таблиці.

Однак, об’єкти поділяються на стандартні та кастомні (тобто які були створені

власноруч адміністратором). Salesforce створює деякі стандартні об'єкти, які користувач може використовувати. Однак, якщо цього не достатньо, платформа дає можливість легко створити власні об'єкти.

Схожа ситуація і у полів. Кожен об'єкт, при створенні, одразу містить в собі чотири поля, які генерує сама платформа Salesforce, а саме: ідентифікатор запису (Id), назва запису (Name), та 3 службові поля: ким був останній раз змінений запис (LastModifiedById), дата створення запису (CreateDate) та дата останньої модифікації запису (LastModifiedDate). Ці поля є стандартними і їх неможливо видалити чи змінити. Однак, адміністратор середовища має змогу додати власні поля, які потрібні для користувача.

Щодо зв'язків між об'єктами, то у Salesforce існують два основних типи: Lookup Relationships та Master-Detail Relationships.

Зв'язок Lookup зазвичай використовують для того, щоб пов'язати два об'єкта між собою у відношенні зв'язків "один до одного" або "один до багатьох". Особливістю цього зв'язку є те, що обидва об'єкти залишаються по суті незалежними один від одного. Наприклад, якщо видалити один з двох об'єктів, то з іншим об'єктом нічого не відбудеться, він залишиться таким як є.

Зв'язок Master-Detail зазвичай використовують для того, щоб пов'язати два об'єкта між собою у відношенні зв'язків "багато до багатьох". Особливістю даного зв'язку є те, що батьківський об'єкт є головним, а дочірній - другорядним. Тобто, якщо видалити батьківський об'єкт, то всі його дочірні об'єкти теж будуть видалені. У зворотному процесі, якщо видалити дочірні об'єкти, батьківський об'єкт залишиться незмінним.

## **2.4. Стек технологій.**

Як відомо, будь-який WEB - застосунок складається з двох частин: клієнтської та серверної. У Salesforce уся серверна частина працює на платформі Lightning. Зрештою, вона поєднується з клієнтською частиною за допомогою викликів до інтерфейсу API платформи Lightning. Тому, для створення інтерфейсу використовують фреймворк Visualforce. Для написання бізнес - логіки використовуються мову програмування Apex.

## **Visualforce:**

**Visualforce** - це фреймворк веб-розробки, що дозволяє розробникам створювати складні інтерфейси, що настроюються та підтримуються платформою Lightning. В інфраструктурі Visualforce використовується мова розмітки на основі тегів, подібною до HTML.

## **Apex:**

**Apex** - суворо типізована об'єктно-орієнтована мова програмування, що дозволяє розробникам виконувати оператори керування транзакціями та потоками на сервері платформи Lightning. Завдяки синтаксису, що нагадує Java і діє подібно до збережених процедур бази даних, Apex дозволяє розробникам додавати бізнес-логіку в основні системні події, включаючи натискання кнопок, оновлення пов'язаних записів та сторінки Visualforce.

Деякі особливості мови Apex:

1. Apex зберігається, компілюється та виконується на сервері — платформі Lightning.
2. Мова Apex не чутлива до регістру.
3. Інтегрована з базою даних. Apex забезпечує прямий доступ до записів та їх полів.

## **SOQL:**

**SOQL** ( Salesforce Object Query Language ) - це мова, яка використовується для отримання даних із Salesforce. SOQL синтаксично схожий на SQL. Можна написати та виконати запит SOQL у коді Apex або в редакторі запитів консолі розробника. На рисунку 2.4 показано запит за допомогою SOQL, який повертає з організації дані про перших десятих користувачів. На рисунку 2.5 - отриманий результат.

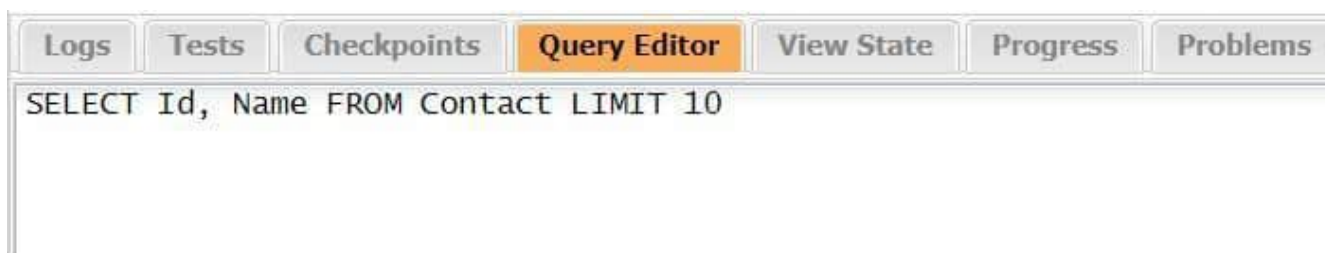


Рисунок 2.4. Запит за допомогою SOQL в редакторі запитів консолі розробника.

Query Results - Total Rows: 10	
Id	Name
0037R000036BPBDAQ4	Rose Gonzalez
0037R000036BPBEQA4	Sean Forbes
0037R000036BPBFQA4	Jack Rogers
0037R000036BPBGQA4	Pat Stumuller
0037R000036BPBHQ44	Andy Young
0037R000036BPBIQA4	Tim Barr
0037R000036BPBJQA4	John Bond
0037R000036BPBKQA4	Stella Pavlova
0037R000036BPBLQA4	Lauren Boyle
0037R000036BPBMQ44	Babara Levy

Рисунок 2.5. Результат попереднього запиту в редакторі запитів консолі розробника.

### Salesforce Sites:

Salesforce Sites дають змогу створювати загальнодоступні веб-сайти та програми, які безпосередньо інтегровані з вашою організацією Salesforce. Це дозволяє оприлюднити будь-яку інформацію, що зберігається у вашій організації, через фірмову URL-адресу, яку можна налаштувати особисто. Оскільки сайти розміщуються на серверах Lightning Platform, проблем з інтеграцією даних немає. А через те, що сайти створені на оригінальних сторінках Visualforce, перевірка даних зібраної інформації виконується автоматично. Таким чином, не потрібно користуватись послугами сторонніх сервісів (наприклад, Heroku), щоб сайт знаходився у хмарному середовищі і був доступний усім бажаючим. За нас це робить платформа Salesforce.

Для кожного з ваших сайтів треба визначити URL-адресу сайту, встановлюючи доменне ім'я сайту. Ось як буде виглядати приклад такого URL - адресу:

*http://MyDomainName.my.salesforce-sites.com.*

Варто додати, що налаштування Salesforce Sites відбувається у самій організації, у розділі "Налаштування".

## **Batch Apex:**

Batch Apex працює з невеликими партіями записів, охоплюючи весь набір записів і розбиваючи обробку на контрольовані фрагменти. Batch Apex представлений як інтерфейс **Database.Batchable**, який має бути реалізований розробником. **Database.Batchable** інтерфейс містить три методи, які потрібно імплементувати, а саме: `start`, `execute` та `finish`.

- **start** - зібрати записи або об'єкти для передачі в метод `execute`;
- **execute** - виконує усю необхідну функціональність;
- **finish** - повідомляє про успішне виконання методу `execute` та дозволяє робити операції постобробки, такі як надсилання електронних листів.

Зроблено це для того, аби мати змогу працювати з великою кількістю записів у нашій CRM системі та не боятись перевищити ліміт щодо запитів у контексті транзакції. Batch Apex ідеально підходить для тривалих завдань із великими обсягами даних, які потрібно виконувати пакетами, наприклад, робота з базою даних.

## **Apex Scheduler:**

Salesforce надає можливість запускати готовий функціонал у Apex класі у певний період часу. Для цього потрібно імплементувати готовий інтерфейс `Schedulable` для Apex класу. The `Schedulable` інтерфейс містить один метод, який необхідно реалізувати - це `execute`. Саме у ньому ми можемо реалізовувати весь функціонал, який ми бажаємо виконувати за графіком. Однак, найкращим рішенням буде створити додатковий клас, де буде реалізована вся необхідна функціональність, і потім викликати його у методі `execute`. Будувати розклад можна будь який, для прикладу запускати функціональність раз в місяць, або кожного дня чи в конкретну дату. Ще однією хорошою функцією є те, що інтерфейси `Schedulable` та `Database.Batchable` можна поєднувати, а саме викликати метод `executeBatch` інтерфейсу `Database.Batchable` у методі `execute` інтерфейсу `Schedulable`.

Контролювати процес виконання робіт, які створюють **Apex Scheduler** та **Batch**

Action	Submitted Date	Job Type	Job Sub Type	Status	Percent Complete	Errors	Submitted By	Completion Date
	10/05/2023, 00:15	Chatter Unified Digest		Completed	100%	<input type="checkbox"/>		10/05/2023, 00:26
	09/05/2023, 00:15	Chatter Unified Digest		Completed	100%	<input type="checkbox"/>		09/05/2023, 00:24
	08/05/2023, 00:15	Chatter Unified Digest		Completed	100%	<input type="checkbox"/>		08/05/2023, 00:30
	07/05/2023, 00:15	Chatter Unified Digest		Completed	100%	<input type="checkbox"/>		07/05/2023, 00:23

Рисунок 2.6. Список виконання робіт

У цьому розділі можна переглянути усі необхідну інформацію, такі як статус чи дата та час завершення роботи.

### Telegram:

Telegram - це месенджер, який дозволяє користувачам обмінюватися повідомленнями, зображеннями, відео та іншими файлами. Один із способів розширення можливостей Telegram - використання Telegram-ботів.

### Telegram-Bot:

Telegram-бот - це програмний інтерфейс, який дозволяє автоматизувати взаємодію з Telegram. Він може використовувати різноманітні функції, такі як надсилання повідомлень, відповіді на запити користувачів, інтеграція з іншими сервісами тощо. Створити власний Telegram-бот можна за допомогою головного бота телеграму під назвою “**BotFather**”. Для роботи з Telegram-ботом необхідно використовувати Telegram Bot API.

### Telegram Bot API:

Telegram Bot API - програмний інтерфейс, який дозволяє взаємодіяти з ботом, а саме:

- Відправлення повідомлень користувачам
- Обробка команд, які надсилають користувачам
- Створення кнопок та інтерактивних елементів інтерфейсу

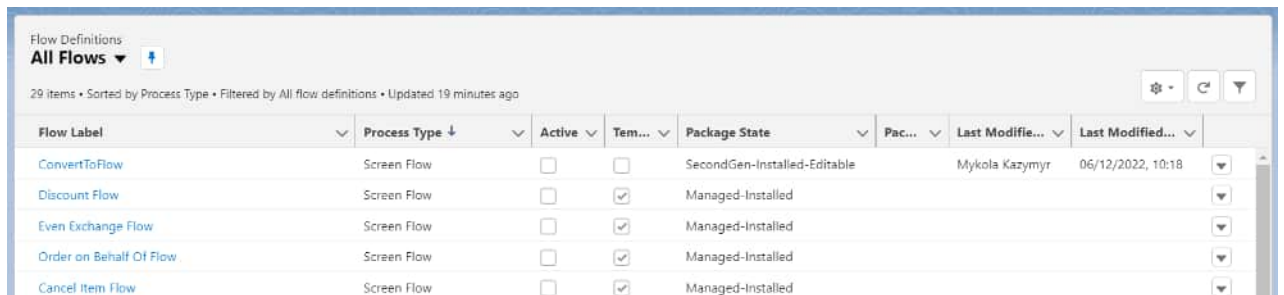
### Webhook:



Webhook - це механізм веб-розробки, який дозволяє веб-додатку отримувати в реальному часі сповіщення про події, які відбуваються на іншому веб-сайті або веб-додатку. За допомогою webhook, веб-додаток може отримувати повідомлення від інших додатків або веб-сайтів у момент їх виникнення, що дозволяє створювати більш ефективні та інтегровані програми. Webhook передає дані у вигляді HTTP-запиту в заданій URL-адресі, що дозволяє прийняти та обробляти ці дані. При цьому веб-додаток, який отримує webhook, може здійснювати різноманітні дії з отриманими даними, наприклад відправляти повідомлення користувачам.

### Flow. Flow Builder:

Flow - це механізм автоматизації у Salesforce, який дозволяє створювати складні потоки роботи з точними інструкціями та правилами для виконання різних дій. Flow можна використовувати як і для внутрішніх процесів у середовищі Salesforce, так і для інтеграції з іншими системами. Переглянути список усіх процесів Flow можна у розділі **All Flows**, який зображено на рисунку 2.7.



The screenshot shows the 'All Flows' section in Salesforce. It displays a table with the following columns: Flow Label, Process Type, Active, Tem..., Package State, Pac..., Last Modifie..., and Last Modified... The table contains five rows of flow definitions.

Flow Label	Process Type	Active	Tem...	Package State	Pac...	Last Modifie...	Last Modified...
ConvertToFlow	Screen Flow	<input type="checkbox"/>	<input type="checkbox"/>	SecondGen-Installed-Editable		Mykola Kazymyr	06/12/2022, 10:18
Discount Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Even Exchange Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Order on Behalf Of Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Cancel Item Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			

Рисунок 2.7. Розділ All Flows

Flow Builder - це графічний інтерфейс, який дозволяє створювати Flow без необхідності в програмуванні. За допомогою Flow Builder можна створювати Flow різного типу, в залежності від бізнес потреб. На рисунку 2.7 зображено вигляд даного графічного інтерфейсу.

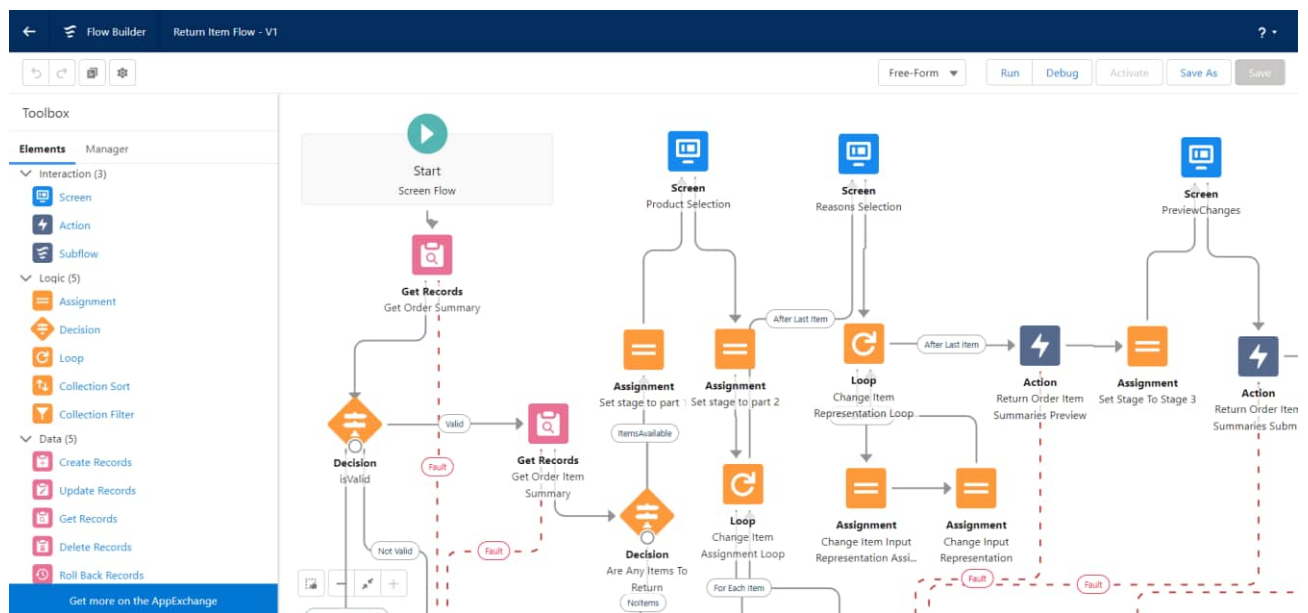


Рисунок 2.8. Flow Builder

### Apex REST Resource:

Apex REST Resource - це фреймворк в Salesforce, який дозволяє створювати RESTful веб-сервіси на базі Apex-класів. За допомогою Apex REST Resource можна створювати кастомні API, які дозволяють забезпечити інтеграцію між Salesforce та іншими системами, в тому числі зовнішніми додатками, веб-сервісами та мобільними додатками. Apex REST Resource дозволяє передавати дані в різноманітних форматах, таких як JSON та XML.

### LWC (Lightning Web Components):

LWC (Lightning Web Components) - це фреймворк для створення веб-інтерфейсів на платформі Salesforce. LWC базується на стандартах веб-розробки, таких як HTML, CSS і JavaScript, і дозволяє розробникам створювати компоненти, які можна використовувати в різних частинах додатків Salesforce.

## Розділ 3. Проектування та реалізація системи

### 3.1 Проектування WEB - застосунку

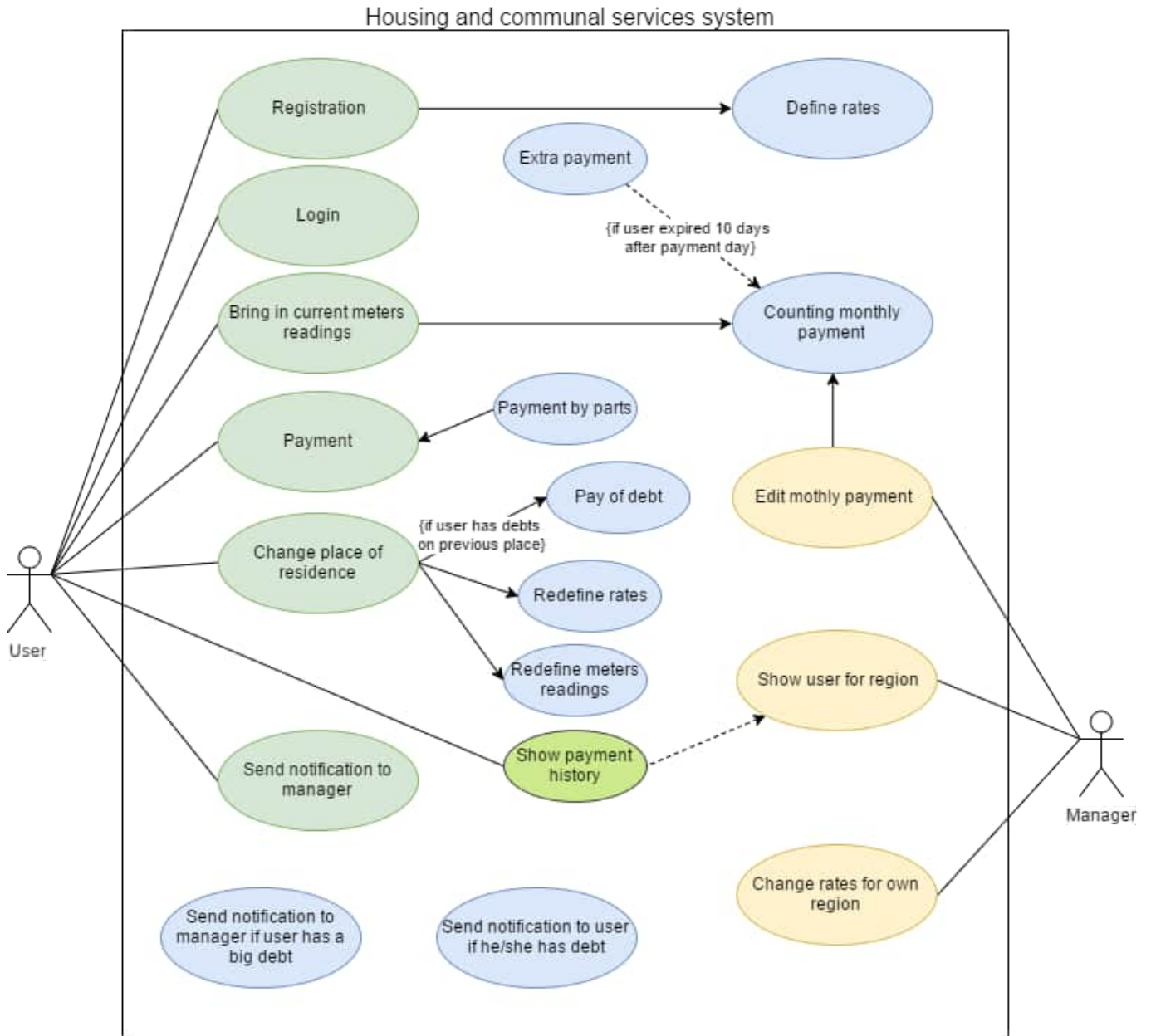


Рисунок 3.1. Use - case діаграма

На рисунку 1 показана use - case діаграма данного WEB - застосунку. Варто сказати, оскільки роль адміністратора нам надає сама організація, відповідно, на даній діаграмі можна помітити лише дві ролі: менеджера (manager) та користувача (user). Зеленим кольором позначені дії, які може виконувати користувач. Жовтим - які виконує менеджер. Синім кольором позначено процеси, які надає організація.

### 3.2 Проектування та імплементация бази даних

Відповідно до вимог даного WEB - застосунку, мною було створено 4 об'єкти (таблиці), а саме:

1. Custom User. Цей об'єкт буде зберігати дані користувачів. Відповідно, було створено поле isManager типу Boolean, щоб можна було записувати два типи користувачів.
2. Region. Даний об'єкт зберігає дані про певний регіон, а саме тарифи на такі сервіси, як газ, вода та електроенергія.
3. Payment. Цей об'єкт зберігає дані про історію комунальних платежів кожного користувача.
4. Token. Даний об'єкт був створений для авторизації та аутентифікації користувачів. Там зберігаються токени для кожного користувача, який хоче увійти в систему.

За допомогою такого засобу, як Schema Builder у організації Salesforce, можна з легкістю переглядати потрібні об'єкти та зв'язки між ними.

Це показано на рисунку 2.

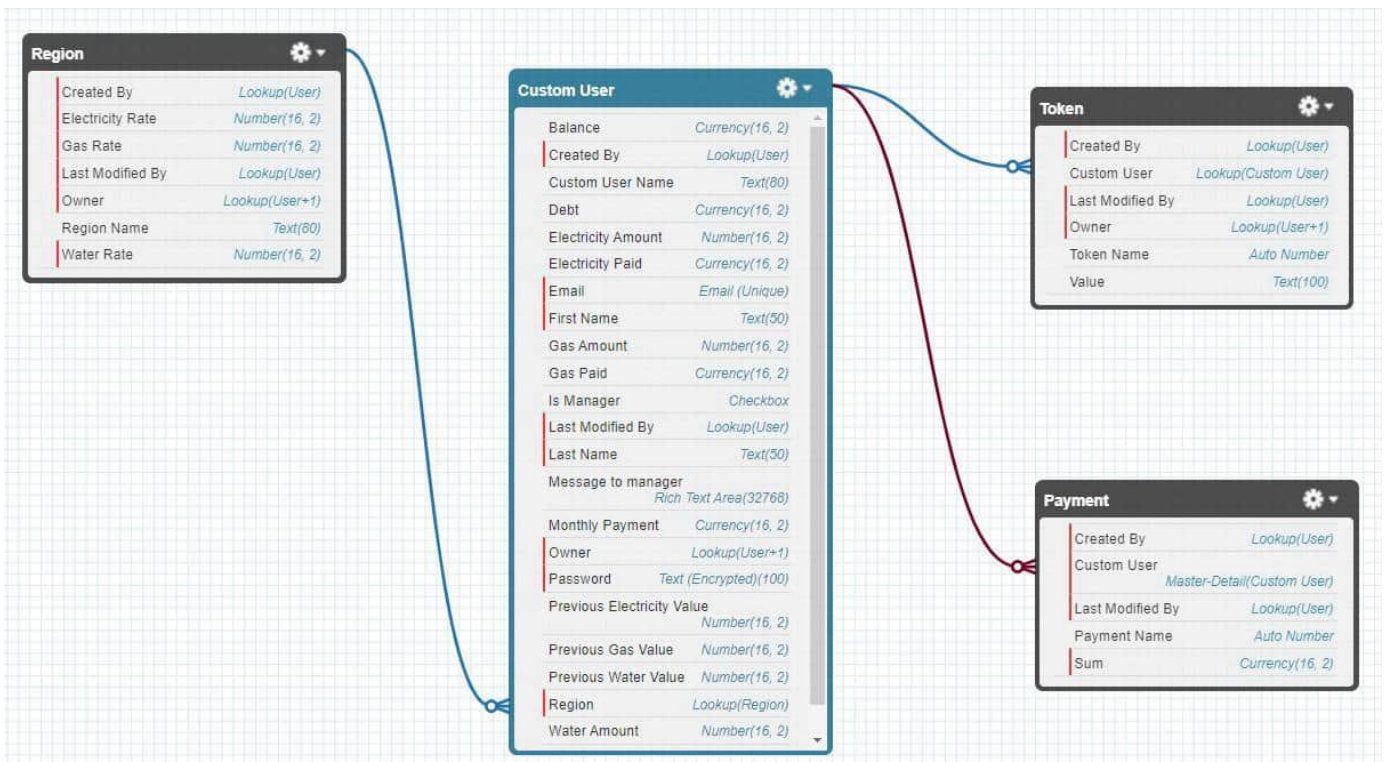


Рисунок 3.2. Створенні об'єкти та зв'язки між ними.

### 3.3 Головна сторінка.

Коли користувач вперше бачить даний застосунок, його зустрічає сторінка з інформацією, що це за сервіс і для чого він призначений. Також користувач може увійти в систему, якщо він попередньо зареєструвався, або, якщо він вперше на цій платформі - створити новий обліковий запис.

## Комунальні платежі

За допомогою сервісу "Communal Service" Ви можете забути про черги, щоб оплатити Ваші комунальні платежі. Тепер Ви можете це зробити не виходячи зі свого дому!

Почати

Рисунок 3.3. Вигляд головної сторінки.

### 3.4 Реєстрація

Для початку роботи з сервісом, користувач повинен зареєструватись. Для реєстрації необхідно ввести ім'я, прізвище, електронну адресу, пароль та обрати регіон, в якому Ви проживаєте. На рисунку 1 показано, як виглядає форма реєстрації.

The screenshot shows the registration form titled "Реєстрація нового користувача" (New user registration) on the "Communal Service" website. The form contains five input fields: "Введіть Ваше Ім'я:" (Enter your name), "Введіть Ваше Прізвище:" (Enter your surname), "Введіть електронну пошту:" (Enter your email), "Введіть пароль:" (Enter password), and "Оберіть Ваш регіон проживання:" (Select your region of residence). The region dropdown menu is currently set to "Lvivska". A "Зареєструватись" (Register) button is located at the bottom of the form.


Рисунок 3.4. Форма реєстрації користувача

Якщо користувач хоче ввести електронну адресу, яка вже зареєстрована в системі, йому буде показане відповідне повідомлення.

The screenshot shows the same registration form as in Figure 3.4, but with an error message displayed at the top in a red banner: "Error: This email is already taken!". The input fields are filled with the following data: "Введіть Ваше Ім'я:" (Mykola), "Введіть Ваше Прізвище:" (Kazymyr), "Введіть електронну пошту:" (mykola.kazymyr@gmail.com), "Введіть пароль:" (12), and "Оберіть Ваш регіон проживання:" (Lvivska). The "Зареєструватись" (Register) button is still visible at the bottom.

Рисунок 3.5. Валідація електронної пошти

Однак, якщо реєстрація пройшла успішно, користувач отримає електронний лист з привітанням:

Welcome!  Вхідні ✕



**ComunalPayment Larkin** з домену 5vbbthylhthp0oma.tnm.7r-3obt1uae.eu45.bnc.salesforce.com  
кому мені ▾

 англійська ▾ > українська ▾ [Перекласти повідомлення](#)

Thanks for your registration, Mykola Kazymyr

 Відповісти

 Переслати

Рисунок 3.6. Електронний лист з вітанням щодо успішної реєстрації

### 3.5 Авторизація

Зареєстрований користувач має можливість увійти у систему ввівши електронну адресу і пароль. Форма входу в особистий кабінет показана на рисунку 3.

Communal Service

**Вхід до особистого кабінету**

Введіть електронну адресу:

Введіть пароль:

Рисунок 3.7. Форма входу в особистий кабінет

Для менеджера та звичайного користувача форма входу є абсолютно однаковою. Лише перевіривши вхідні дані, тобто електронну адресу та пароль, система зможе зрозуміти, хто саме входить у обліковий запис.

### 3.6 Сторінка користувача

Після успішного входу в систему або успішної реєстрації, користувач потрапляє на сторінку для всіх необхідних операцій.

Input Water Value:	<input type="text" value="10.00"/>
Input Gas Value:	<input type="text" value="10.00"/>
Input Electricity Value:	<input type="text" value="10.00"/>
Choose, what to want to choose:	<input type="text" value="Payment"/>
Region:	<input type="text" value="Lvivska"/>
Input Amount:	<input type="text" value="0"/>
Your Debt:	0.00
Your Monthly Payment:	0.00
Payment for Gas:	0.00
Payment for Water:	0.00
Payment for electricity:	0.00
Enter message to manager:	<input type="text"/>

**Історія платежів**

PAYMENT NUMBER	SUM	DATE AND TIME
P-0109	€90.00	15/05/2022, 15:57
P-0110	€90.00	15/05/2022, 16:03
P-0111	€90.00	15/05/2022, 16:05

Рисунок 3.8. Сторінка користувача

На даній сторінці користувач може:

- внести показники лічильників для трьох послуг: газ, вода та електроенергія;
- вибрати спосіб оплати: оплатити всю суму чи лише за певну послугу;
- оплатити борг;
- надіслати лист регіональному менеджеру;
- переглянути історію платежів з додатковою інформацією (ідентифікаційний номер квитанції, сума та час сплати);
- переглянути баланс коштів, а також суму боргу;
- переглянути оплату за комунальні послуги, яку нарахувала система за цей місяць, а також за кожен сервіс окремо.

Відповідно до функціоналу системи, користувачу, після успішної оплати приходить електронна квитанція на електронну пошту.

Payment Information	
Payment Number	P-0036
Who Pay	Mykola Kazymyr
Sum	10.00
Date	2022-02-15 21:54:12

Рисунок 3.9. Електронна квитанція про оплату комунальних послуг.

Однак, якщо у користувача недостатньо коштів, він отримає оповіщення про це:

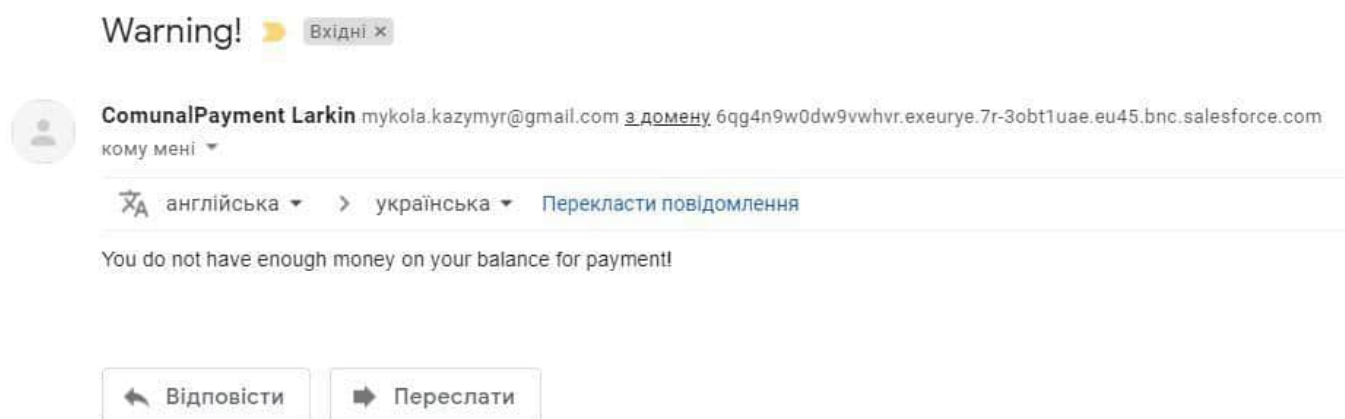


Рисунок 3.10. Електронний лист з попередженням про недостатній баланс коштів.

Також, користувач повинен отримувати нагадування на електронну пошту про те, що він не вніс нові показники лічильників першого по п'яте число кожного нового місяця:

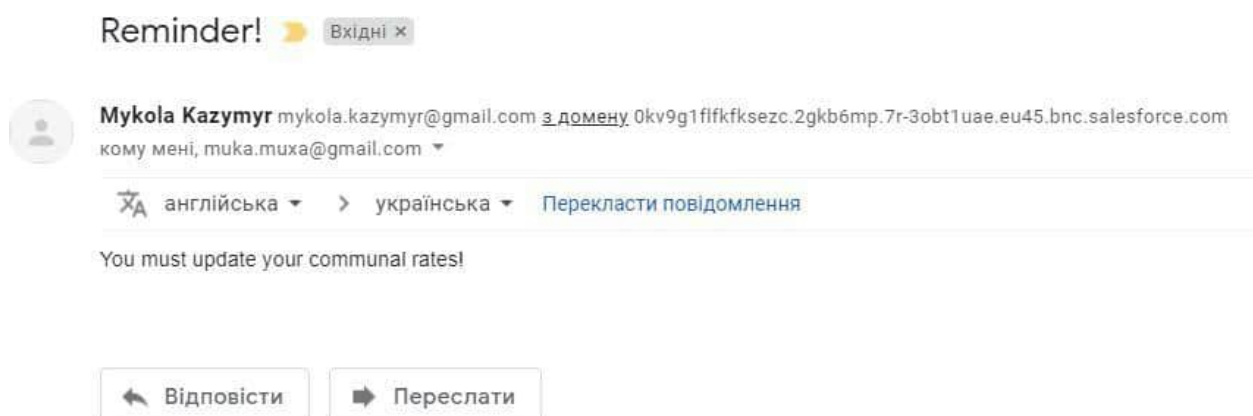




Рисунок 3.11. Нагадування про не внесені показники лічильників.

### 3.7 Сторінка регіонального менеджера

У даному застосунку, адміністратор може створити обліковий запис регіонального менеджера. Далі, менеджер повинен увійти до нього за допомогою електронної пошти та паролю. Після чого, для менеджера відкривається сторінка з функціональністю для його операцій.

The screenshot displays the 'Communal Service' manager interface. At the top right, it says 'Yurii Manager Lvivska'. The interface is divided into several sections:

- Rate Input Section:** Contains three input fields for 'Input New Water Rate', 'Input New Gas Rate', and 'Input New Electricity Rate', each with a value of '3.00'. Below these are 'Save Rates' and 'Show Users' buttons.
- User Management Section:** Contains two input fields for 'Input Users Email' (with 'mykola.kazymyr@gmail.com') and 'Input new Monthly Payment Value' (with '0'). Below are 'Change Payment' and 'Show User History' buttons.
- User List Table:** A table with columns 'USER NAME', 'MONTHLY PAYMENT', and 'EMAIL'. It lists one user: 'Mykola Kazymyr' with a monthly payment of '€0.00' and email 'mykola.kazymyr@gmail.com'.
- Payment History Table:** A table with columns 'PAYMENT NUMBER', 'SUM', and 'DATE AND TIME'. It lists three payments: 'P-0109' (€90.00, 15/05/2022, 16:57), 'P-0110' (€90.00, 15/05/2022, 17:03), and 'P-0111' (€90.00, 15/05/2022, 17:05).

Рисунок 3.12. Сторінка менеджера На даній сторінці менеджер має змогу:

- встановлювати тарифи для свого регіону для таких сервісів, як: газ, вода та електроенергія;
- переглядати список користувачів зі свого регіону;
- змінювати місячну оплату для певного користувача зі свого регіону;
- переглядати історію платежів певного користувача

### 3.8 Оптимізація функціоналу. Виконання дій згідно графіку.

Оптимізація функціональності програми дозволяє забезпечити її ефективну роботу, зменшити витрати ресурсів та забезпечити більш якісну роботу програми. В результаті, користувачі отримують більш швидкий, надійний та ефективний продукт.

Оптимізувати роботу нашого застосунку допоможе **Asynchronous Apex**. Це готове рішення від Salesforce, яке допоможе запускати код асинхронно. У ньому є декілька функцій, з яких було обрано Apex Scheduler та Batch Apex.

Однією з функцій даного застосунку є те, щоб система автоматично нагадувала користувачеві про необхідність оновити показники лічильників. Користувач повинен

внести нові показники кожного місяця у перші 5 днів. У разі не внесення актуальних даних, було б доцільно нагадати про це на 6 та 10 день кожного нового місяця. На шостий день користувачу прийде сповіщення на пошту про те, що йому необхідно оновити дані. Відповідне повідомлення зображено на рисунку 3.13.

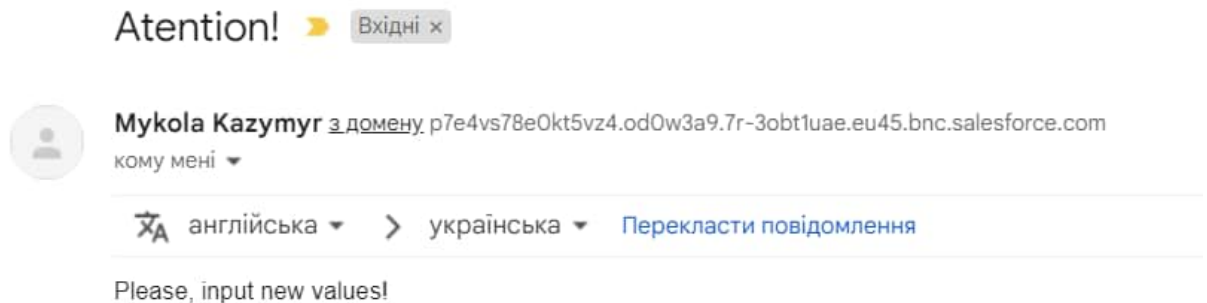


Рисунок 3.13. Повідомлення з проханням оновити показники лічильників

Якщо користувач цього не зробить, то на десятий день йому нараховується сума, яка є встановлена менеджером для кожного регіону. Таким чином, це полегшить життя користувачеві, адже система нагадає йому, щоб він оновив показники лічильників. І у разі невиконання, менеджер буде впевненим, що система внесла суму боргу за ігнорування користувачем оновити дані.

### 3.9 Інтеграція Salesforce з third-party сервісами. Інтеграція Salesforce & Telegram.

Інтеграція сторонніх додатків з Salesforce - це процес, який дозволяє підключати різноманітні зовнішні сервіси та рішення до CRM-системи Salesforce. Інтеграція є важливою, оскільки дозволяє компаніям розширити функціональність своєї CRM-системи та підвищити ефективність своїх бізнес-процесів.

Важливим функціоналом для застосунку для оплати комунальних платежів є те, як наша система може сповіщати користувача. Це може відбуватись за допомогою електронної пошти, однак зараз великої популярності набирають онлайн месенджери, такі як Telegram. Це зручний спосіб сповістити користувача, адже популярні онлайн месенджери доступні на всіх пристроях, наприклад на смартфоні чи навіть смарт-годиннику. Тому інтегрувати WEB-застосунок для оплати комунальних платежів з додатком Telegram буде хорошим рішенням для зручності користувачів даної системи.

Використовуючи BotFather було створено телеграм-бот під назвою Salesforce Notification Bot. Також BotFather згенерував для цього бота токен, за допомогою якого можливо налаштувати webhook. Для налаштування webhook необхідно вказати токен бота, а також endpoint, куди буде надходити інформація про взаємодію з ботом. Ним буде URL-адреса сайту комунальних платежів, а саме доменна та кастомна API частина, яка була створена за допомогою Rest Resource та присвоєна конкретному Арех класу. Пересвідчитись, що webhook було встановлено допоможе відповідне повідомлення, яке зображене на рисунку 3.14.

```
{"ok":true,"result":true,"description":"Webhook is already set"}
```

Рисунок 3.14. Результат успішного встановлення webhook

Тепер, коли користувач почне роботу з Salesforce Notification Bot, він зможе побачити наступні повідомлення:

1. Бот надішле повідомлення з проханням вказати електронну адресу, яку користувач ввів при реєстрації на сайті.



Рисунок 3.15. Повідомлення з проханням ввести електронну адресу користувача

2. Якщо користувач введе електронну адресу, яка є у базі даних Salesforce, бот привітає користувача та повідомить, що він тепер успішно може надсилати сповіщення користувачу.

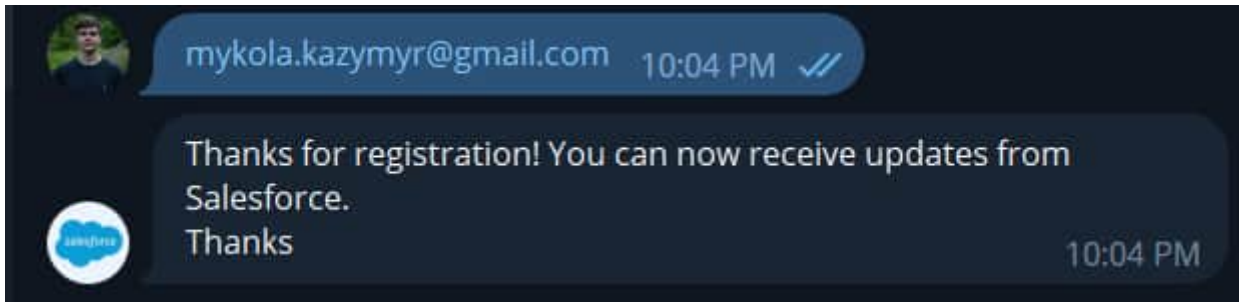


Рисунок 3.16. Повідомлення про те, що користувач успішно зареєструвався та може отримувати сповіщення

3. У випадку, якщо користувач не зареєстрований на сайті, то відповідно у базі даних Salesforce немає його електронної адреси. Тоді бот відправить відповідне повідомлення.

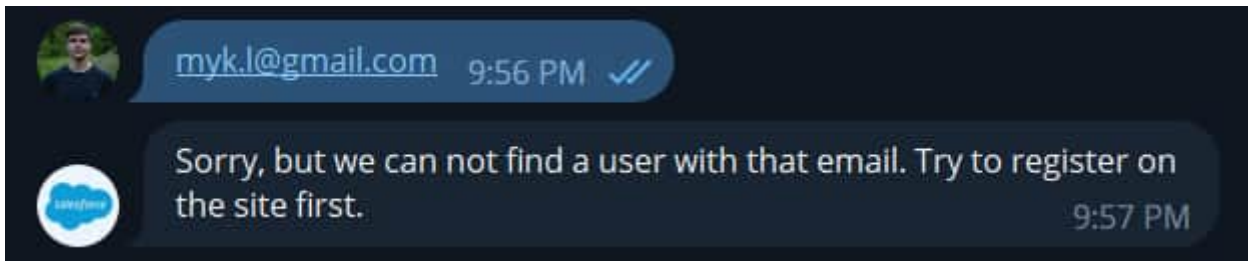


Рисунок 3.17. Повідомлення з проханням спершу зареєструватись на сайті, адже даного електронного адресу немає у базі даних Salesforce

4. Існує також відповідь бота на випадок, якщо користувач спробує повторно зареєструватись, використовуючи той самий електронний адрес. У такому випадку бот відповідним повідомленням нагадає користувачу, що він вже зареєстрований у системі.

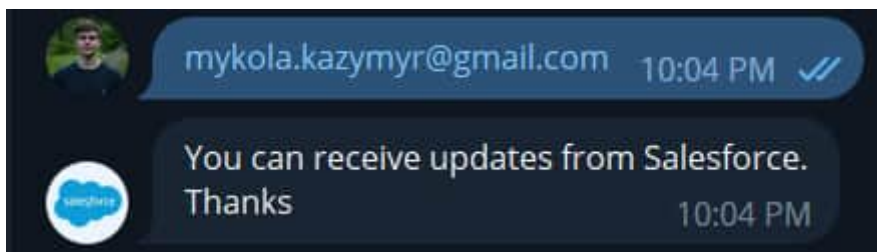


Рисунок 3.18. Нагадування про те, що користувач вже зареєстрований та може отримувати сповіщення

Основний процес інтеграції Salesforce з Telegram успішно виконаний, однак для комфортного використання месенджера Telegram, а саме надсилання повідомлень зі

сторони Salesforce необхідно створити зручний графічний інтерфейс. Зробити це допоможе фреймворк Lightning Web Components.

### 3.10 Lightning Web Components.

Для того щоб менеджер або адміністратор застосунку міг надіслати повідомлення користувачу, необхідно створити власний компонент. На ньому буде відображатись текстове поле для вводу тексту та кнопка відправлення повідомлення. Реалізацію даного LWC компоненту зображено на рисунку 3.19.

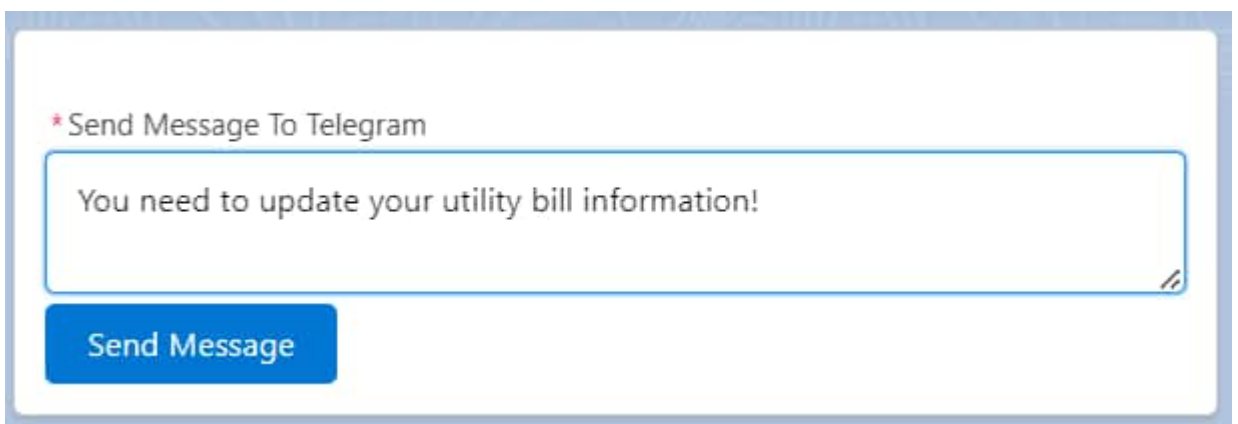


Рисунок 3.19. LWC компонент для надсилання повідомлень

За допомогою функціоналу, який надає Salesforce можна зручно розмістити компонент на сторінці користувача у самій Salesforce організації.

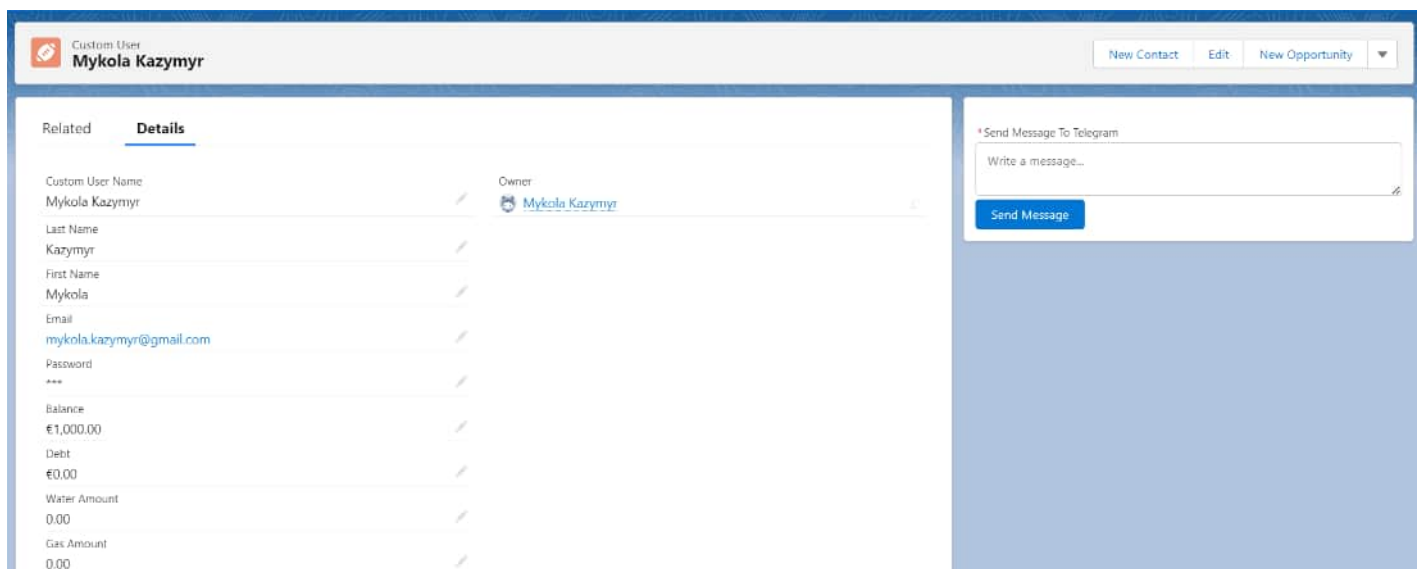


Рисунок 3.20. Загальний вигляд сторінки користувача у Salesforce організації з

компонентою для надсилання повідомлень користувачу у Telegram

Тепер адміністратор або менеджер, який взаємодіє з CRM-системою Salesforce зможе написати і надіслати повідомлення. Користувачеві прийде сповіщення в застосунок Telegram з текстом повідомлення.

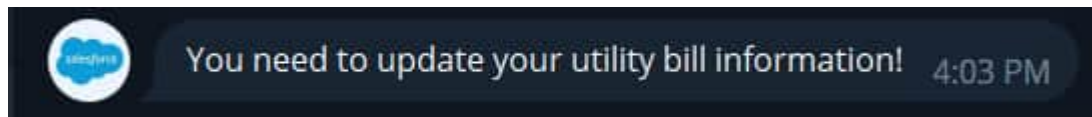


Рисунок 3.21. Повідомлення про необхідність оновити показники лічильників

Для LWC компоненти передбачена обробка повідомлень в ситуаціях, коли виникла проблема з надсиланням повідомлення, або навпаки, коли повідомлення було успішно надіслане. Нижче описані випадки, як саме система реагує на поведінку компоненти.

1. Користувач хоче надіслати повідомлення без тексту. В такому випадку система попередить його, що спершу необхідно ввести текст.

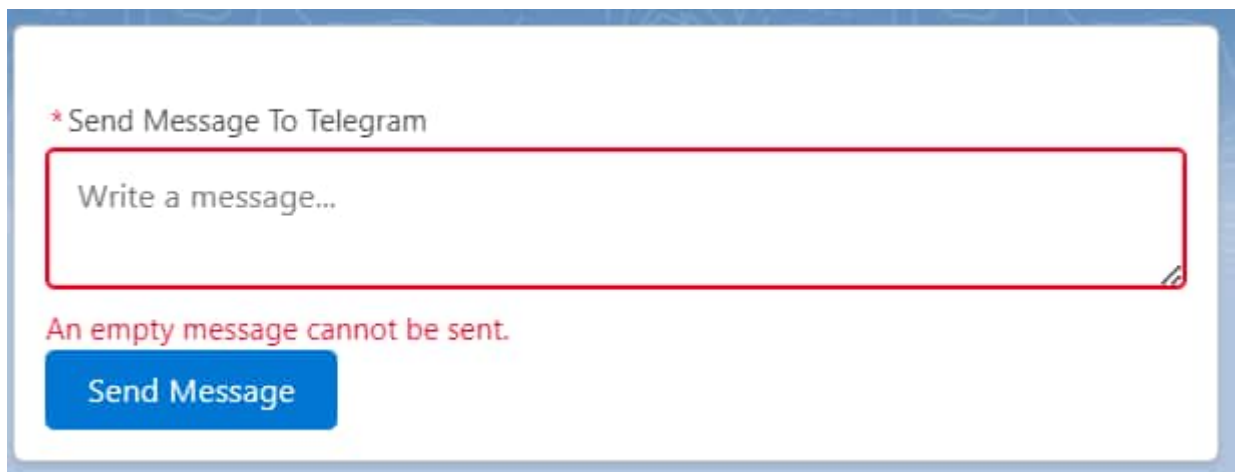


Рисунок 3.22. Системне повідомлення про необхідність заповнити текст для повідомлення

2. Користувач натискає кнопку "Send Message" та повідомлення успішно було надіслане.

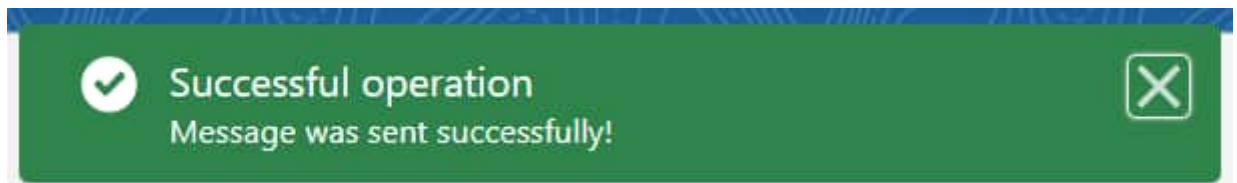


Рисунок 3.23. Системне сповіщення про успішну відправку повідомлення

3. Користувач натискає кнопку “Send Message” та повідомлення не надсилається через системний збій.

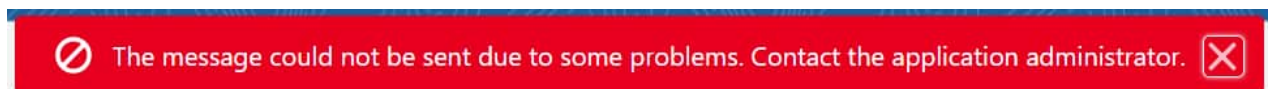


Рисунок 3.24. Системне сповіщення про те, що повідомлення не було надіслане

### **3.11 Автоматизація процесів.**

В сучасному світі швидкість і ефективність мають вирішальне значення. Тому автоматизація процесів стає ключовим фактором успіху для будь-якого бізнесу. У рамках цього контексту, CRM-система Salesforce пропонує потужні інструменти для автоматизації бізнес-процесів. Серед них найкращим і найефективнішим є Flow. Для створення Flow існує візуальний редактор, який надає інтуїтивно зрозумілий інтерфейс для створення Flow без необхідності у написанні коду.

Оптимізація для застосунку по оплаті комунальних платежів є просто необхідною, адже системою будуть користуватись багато користувачів. Одним із випадків, де може знадобитись автоматизація за допомогою Flow у додатку це зміна місячного платежу користувача. Регіональний менеджер має можливість змінювати щомісячний платіж для кожного користувача у своєму регіоні. Це важлива інформація для самого платника, тому варто надсилати її у зручний для нього спосіб. Це може бути як лист з інформацією, надісланий на електронну пошту, так і повідомлення в онлайн месенджер Telegram.

Для цього необхідно створити процес Flow, який буде автоматично реагувати на зміну місячного платежу для певного користувача і сповіщати його. На рисунку 3.25 зображено структуру даного процесу.



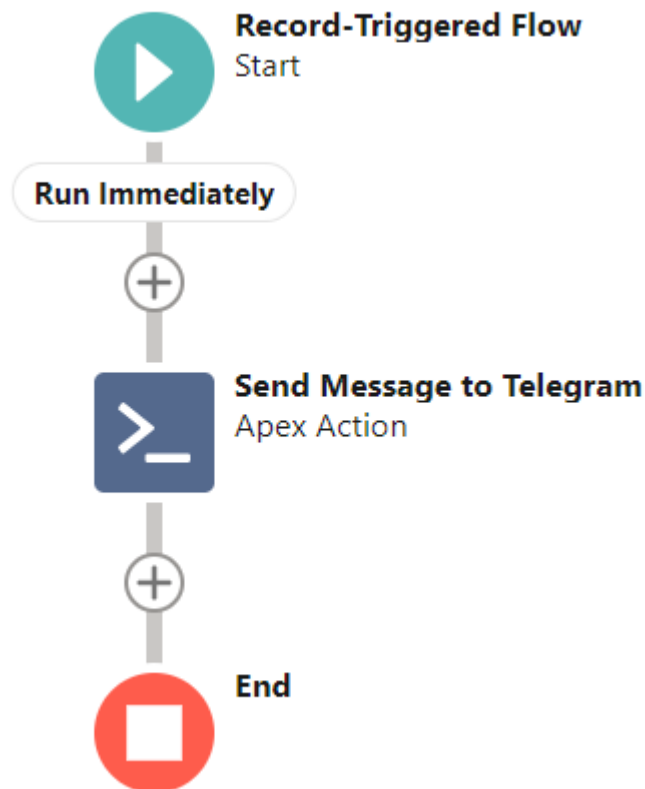


Рисунок 3.25. Вигляд автоматизованого процесу у Flow Builder

Даний процес відбувається за умови, коли запис користувача оновлюється у базі даних. Важливим уточненням є те, що процес спрацьовує лише тоді, коли оновлюється значення у полі місячного платежу, а не весь запис. Якщо користувач використовує Telegram-бот для сповіщень, Flow автоматично візьме унікальний ідентифікатор чату, який записаний у базі даних Salesforce, та надішле повідомлення про зміну оплати для конкретного користувача.



Рисунок 3.26. Повідомлення про зміну місячного платежу для користувача у Telegram

У даному повідомленні вказана сума, яка тепер є актуальною для сплати та хто саме її змінив. Таким чином цей процес полегшує роботу регіонального менеджера, адже йому не потрібно самостійно відправляти повідомлення кожному користувачу, за нього це зробить Flow. Також менеджер може бути впевненим, що користувач

застосунку був вчасно проінформований про зміну місячного платежу.

## **ВИСНОВКИ**

У ході виконання дипломної роботи було досліджено, що CRM - система Salesforce чудово підходить для реалізації WEB - застосунку, метою якого є оплата комунальних платежів. За допомогою цієї платформи можна легко налаштувати середовище для зручної роботи та швидко почати процес створення продукту.

Система, яка була розроблена в ході цієї роботи дає змогу користувачам легко та зручно оновлювати показники лічильників, а також оплатити потрібну суму за комунальні послуги. Було розглянуто додаткові функції Salesforce, такі як автоматизація процесів і можливість створювати зручні та сучасні WEB - компоненти для кращої взаємодії з користувачем. Також Salesforce надає можливість інтегрувати систему із сторонніми додатками, що дозволяє розширити функціонал WEB - додатку. Усе ці фактори свідчать про це, що CRM - система Salesforce чудово адаптується для зручного проектування даного WEB - застосунку, а також дозволяє успішно та швидко комунікувати з користувачами.

Отже проведені дослідження можна назвати успішним, адже зрозуміло, що для розробки справді цікавих та індивідуальних рішень, Salesforce є доволі перспективною технологією, враховуючи те, яку популярність здобувають CRM - системи останнім часом.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Visualforce. [Електронний ресурс]. - Режим доступу:  
[https://help.salesforce.com/s/articleView?id=sf.pages\\_about.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.pages_about.htm&type=5)
2. Apex Code Overview. [Електронний ресурс]. - Режим доступу:  
[https://help.salesforce.com/s/articleView?id=sf.code\\_about.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.code_about.htm&type=5)
3. Salesforce Sites. [Електронний ресурс]. - Режим доступу:  
[https://help.salesforce.com/s/articleView?id=sf.sites\\_overview.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.sites_overview.htm&type=5)
4. What is Salesforce? [Електронний ресурс]. - Режим доступу:  
<https://www.salesforce.com/products/what-is-salesforce/>
5. Salesforce Data Modeling. [Електронний ресурс]. - Режим доступу:  
[https://trailhead.salesforce.com/en/content/learn/modules/data\\_modeling](https://trailhead.salesforce.com/en/content/learn/modules/data_modeling)
6. Salesforce Object Query Language. [Електронний ресурс]. - Режим доступу:  
[https://developer.salesforce.com/docs/atlas.en-us.soql\\_sosl.meta/soql\\_sosl/sforce\\_api\\_calls\\_soql.htm](https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql.htm)
7. Developer Console. [Електронний ресурс]. - Режим доступу:  
[https://help.salesforce.com/s/articleView?id=sf.code\\_dev\\_console.htm&type=5](https://help.salesforce.com/s/articleView?id=sf.code_dev_console.htm&type=5)
8. Asynchronous Apex. [Електронний ресурс]. - Режим доступу:  
[https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_async\\_overview.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_async_overview.htm)
9. Telegram Bot API. [Електронний ресурс]. - Режим доступу:  
<https://core.telegram.org/bots/api>
10. Webhooks in Salesforce. [Електронний ресурс]. - Режим доступу:  
<https://www.apexhours.com/salesforce-integration-using-webhooks/>
11. Flow. Flow Builder. [Електронний ресурс]. - Режим доступу:  
<https://help.salesforce.com/s/articleView?id=sf.flow.htm&type=5>
12. Apex REST Resource. [Електронний ресурс]. - Режим доступу:  
[https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_classes\\_annotation\\_rest\\_resource.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_classes_annotation_rest_resource.htm)
13. Lightning Web Components. [Електронний ресурс]. - Режим доступу:  
<https://developer.salesforce.com/docs/component-library/documentation/en/lwc>
14. Казимир М.М. Integrate Salesforce with Telegram. [Електронний ресурс]. -

Режим доступа: <https://salesforce.in.ua/t/integrate-salesforce-with-telegram/1596>