Міністерство освіти і науки України

Черкаський національний університет імені Богдана Хмельницького

Факультет обчислювальної техніки, інтелектуальних та управляючих систем Кафедра інформаційних технологій

КУРСОВА РОБОТА

з дисципліни «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

на тему: «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

Студента(ки) \_\_\_\_\_курсу \_\_\_\_\_ групи

спеціальності

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код Назва)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (прізвище та ініціали)

Керівник: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Оцінка за шкалою:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(національною, кількість балів, ECTS)

Члени комісії: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

Черкаси – 2020

Зміст

[ВСТУП 3](#_Toc40716856)

[1. ПОСТАНОВКА ЗАДАЧІ 4](#_Toc40716857)

[1.1. Функціональні вимоги 4](#_Toc40716858)

[1.2. Вимоги до технічних і програмних засобів 5](#_Toc40716859)

[1.2.1. Вимоги до технічних засобів 5](#_Toc40716860)

[1.2.2. Вимоги до програмних засобів 5](#_Toc40716861)

[2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ 6](#_Toc40716862)

[2.1. Інфологічне та датологічне проектування 6](#_Toc40716863)

[2.1.1. Інфологічне проектування 6](#_Toc40716864)

[2.1.2. Даталогічне проектування 7](#_Toc40716865)

[2.2. Проектування серверної частини 10](#_Toc40716866)

[2.2.1. Схема і об’єкти бази даних 10](#_Toc40716867)

[2.2.2. Виконання запитів 14](#_Toc40716868)

[2.3. Опис клієнтського додатку 18](#_Toc40716869)

[3. ВИСНОВКИ 22](#_Toc40716870)

[Список літератури 23](#_Toc40716871)

# ВСТУП

Темою даного програмного продукту є створення додатку для індивідуальних користувачів електроенергії. Програмний продукт повинен зберігати список користувачів та дані про історію використання електроенергії. Також система має виводити глобальну статистику користування електроенергією. Проект має бути реалізований як веб додаток. Рядовий користувач буде мати доступ до свого особистого кабінету з приватною інформацією.

Метою даної курсової роботи є аналіз та опрацювання предметної області, та, на основі отриманої інформації, про предметну область, виконання наступних завдань:

* створити базу даних;
* розробити запити обробки інформації;
* створити програмний додаток;
* в додатку організувати перегляд та додавання даних.

# 1. ПОСТАНОВКА ЗАДАЧІ

Завданням курсової роботи є проектування та реалізація бази даних на тему «Облік користувачів електроенергії». База даних повинна містити не менше п’яти таблиць на цю тему. На створену базу даних повинні бути написані запити. Також необхідно реалізувати клієнтський додаток для взаємодії з базою даних. Програма повинна бути простою, зрозумілою та комфортною в використанні.

## 1.1. Функціональні вимоги

Програмний продукт повинен надавати особисту інформацію про користувача, тарифи електроенергію та історію використання. Для адміністрації має генеруватися статистика користування по містам, областям і тип.

Також додаток має:

* Додавання нового користувача
* Давати доступ зареєстрованому користувачу
* Додавати нові рахунки на електроенергію
* Додавання даних в таблиці
* Видалення даних з таблиць
* Редагування даних в таблицях
* Автоматична обробка даних згідно бізнес логіки

## 1.2. Вимоги до технічних і програмних засобів

### 1.2.1. Вимоги до технічних засобів

Для швидкої і стабільної роботи додатку необхідно:

* частота процесора - 2000 МГц;
* оперативна пам’ять – 1 Гб;
* доступ до інтернету на швидкості не менше 256 кб/с.

### 1.2.2. Вимоги до програмних засобів

Для користування додатком через мережу інтернет користувачеві потрібно мати лише операційну систему, яка надає можливість встановлення браузера, що підтримує JS I HTML5 і доступ до інтернету.

Для ручного внесення змін в програму, або базу даних необхідно мати встановлений IDE Intellij IDEA, PostgreSQL server і клієнт до нього, доступ до терміналу або консолі.

# 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ

## 2.1. Інфологічне та датологічне проектування

### 2.1.1. Інфологічне проектування

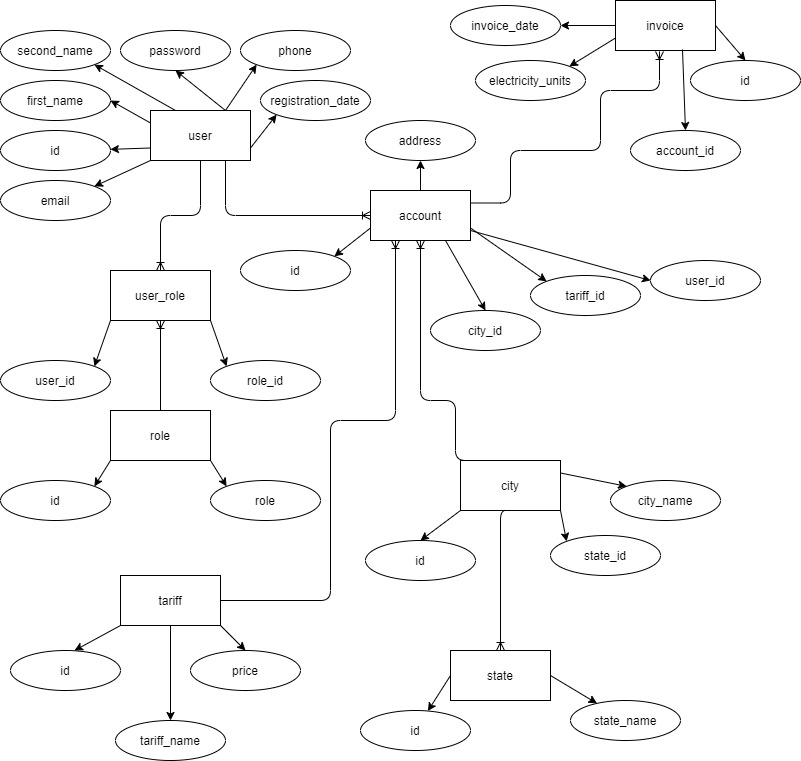


Рис. 2.1. – Інфологічна модель

У користувача може бути декілька ролей і ролі можуть мати більше одного користувача. У користувача може бути декілька профілів. Профіль має один тариф, одне місто кожне місто може мати одну область. Кожен виставлений рахунок має один профіль.

### 2.1.2. Даталогічне проектування

**Таблиця tariff(тарифи)**

* id тарифу, автоінкрементальне поле.
* tariff\_name – поле, в якому міститься назва тарифу, обмежене в 16 символів, оскільки майже гарантовано назва тарифу не буде більшою.
* price – поле типу числа с плаваючою комою яке відображає ціну за одну одиницю електроенергію згідно цього тарифу.

**Таблиця state(області)**

* id області, автоінкрементальне поле.
* state\_name – поле, в якому міститься назва області, обмежене в 32 символи, оскільки назва області не буде більшою.

**Таблиця city(міста)**

* id міста, автоінкрементальне поле.
* state\_id – поле, яке є зовнішнім ключем для поля id таблиці state і позначає область, в якій знаходиться місто.
* city\_name – поле, в якому міститься назва міста обмежене в 16 символів, оскільки назва міста не буде більшою.

**Таблиця role (ролі)**

* id ролі, автоінкрементальне поле.
* role – поле, в якому знаходиться назва ролі обмежене в 16 символів, оскільки назва ролі не буде більшою.

**Таблиця user\_role (користувач-роль)**

* user\_id – поле, яке є зовнішнім ключем для поля id таблиці user і позначає користувача, якому присвоєна дана роль.
* role\_id – поле, яке є зовнішнім ключем для поля id таблиці role і позначає яку саме роль було присвоєно користувачу.

**Таблиця user (користувач)**

* id користувача, автоінкрементальне поле.
* first\_name – поле, в якому знаходиться ім’я користувача, обмежене в 32 символа, оскільки ім’я користувача не буде більшим.
* second\_name – поле, в якому знаходиться прізвище користувача, обмежене в 32 символа, оскільки прізвище користувача не буде більшим.
* email – поле, в якому знаходиться емейл користувача, обмежене в 32 символа, оскільки емейл користувача не буде більшим.
* password – поле, в якому знаходиться зашифрований пароль користувача, обмежене в 128 символа, оскільки після того як пароль буде зашифрований він значно розшириться в розмірі.
* phone – поле, в якому знаходиться контактний номер телефону користувача.
* registration\_date – поле, в якому міститься дата реєстрації користувача.

**Таблиця account (профіль)**

* id профію, автоінкрементальне поле.
* city\_id – поле, яке є зовнішнім ключем для поля id таблиці city і позначає місто, до якого привязаний даний профіль.
* street – поле, яке зберігає на якій вулиці розташований клієнт.
* house – поле, яке зберігає номер будинку в якому знаходиться клієнт.
* appartament – поле, яке зберігає номер квартири в якій знаходиться клієнт. Поле є опіціональним тому, що клієнт може знаходитися в приватному будинку.
* tariff\_id – поле, яке є зовнішнім ключем для поля id таблиці tarrif і позначає тариф, за умовами якого обраховується розмір оплати електроенергії.
* user\_id - поле, яке є зовнішнім ключем для поля id таблиці city і позначає місто, до якого привязаний даний профіль.

**Таблиця invoice (виставлені рахунки)**

* id рахунку, автоінкрементальне поле.
* account\_id – поле, яке є зовнішнім ключем для поля id таблиці account і позначає профіль, якому був виставлений даний рахунок.
* electricity\_units – поле, в якому зберігається кількість одиниць електроенергії, які клієнт має сплатити.
* invoice\_date – дата нарахування даного рахунку.

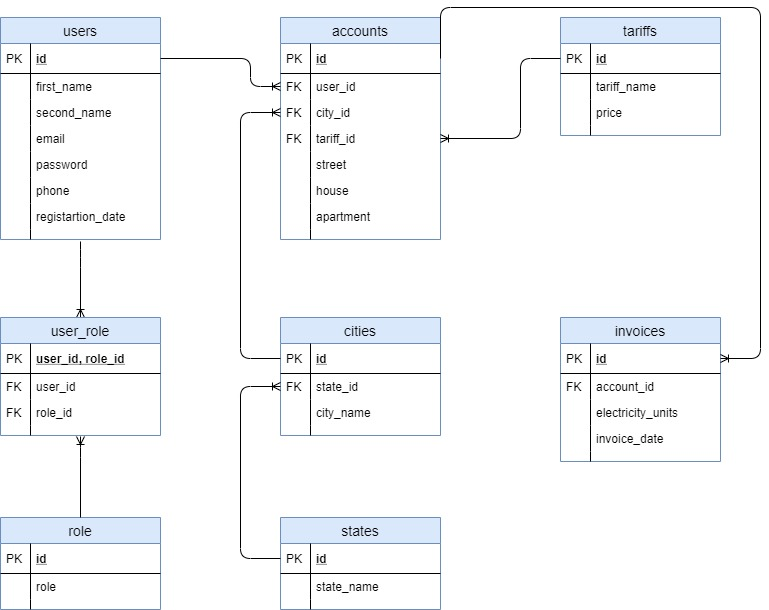


Рис 2.1.2. – Даталогічна модель

## 2.2. Проектування серверної частини

### 2.2.1. Схема і об’єкти бази даних

Схема бази даних:

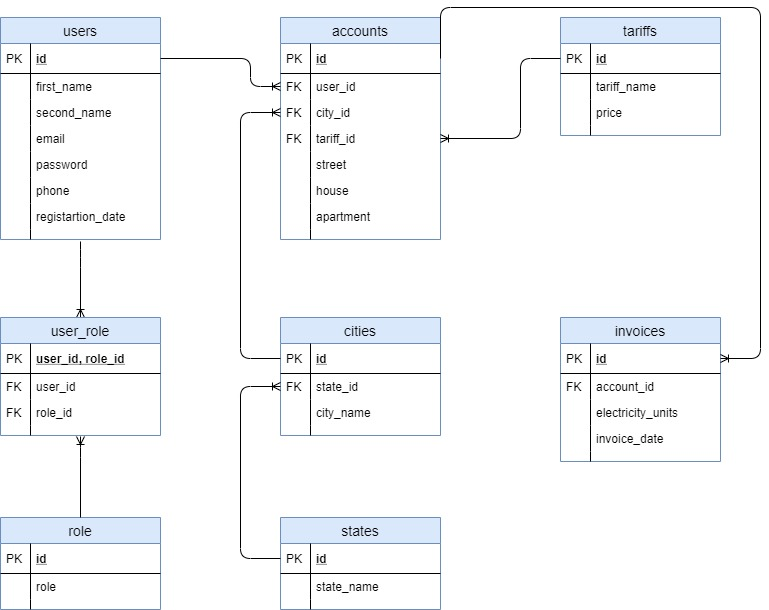


Рис. 2.2.1. – Структура бази даних

**Таблиці**

**USERS**

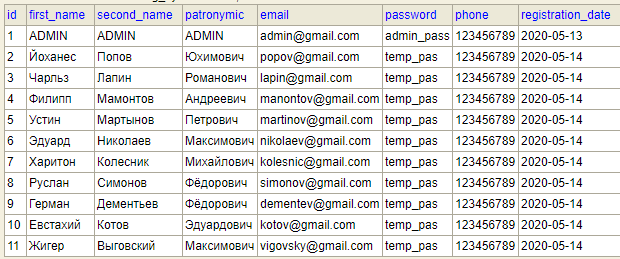
****

Рис. 2.2.2. – Таблиця користувачі

**ROLES**

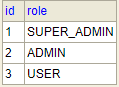
****

Рис. 2.2.3. – Таблиця ролі

**USER\_ROLE**

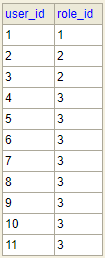
****

Рис. 2.2.4. – Таблиця користувач-роль

**STATES**

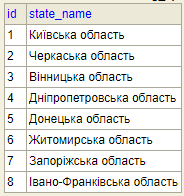
****

Рис. 2.2.5. – Таблиця області

**CITIES**

****

Рис. 2.2.6. – Таблиця міста

**TARIFFS**

****

Рис. 2.2.7. – Таблиця тарифи

**ACCOUNTS**

****

Рис. 2.2.8. – Таблиця профілі

**INVOICES**

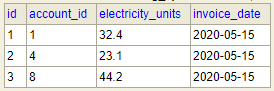
****

Рис. 2.2.9. – Таблиця рахунки до сплати

### 2.2.2. Виконання запитів

Запит INSERT\_ACCOUNTS, який додає нові записи до таблиці accounts:

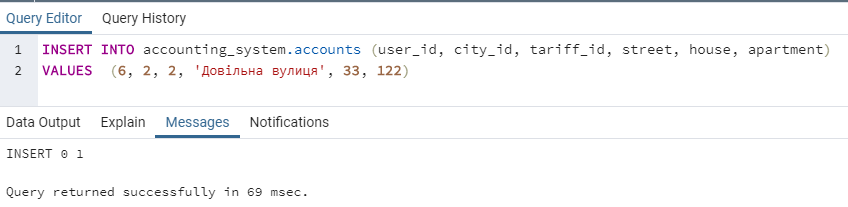
****

Рис. 2.2.2.1. – Запит INSERT\_ACCOUNTS

Запит UPDATE\_ACCOUNT, який змінює інуючий запис в таблиці accounts по id:

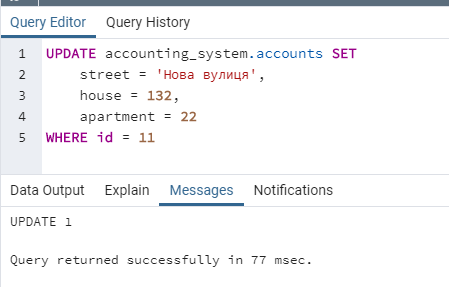


Рис. 2.2.2.2. – Запит UPDATE\_ACCOUNT

Запит GET\_INVOICES\_BY\_USER\_ID, який отримує всі виставлені рахунки по оплаті з усіх привязаних до профілю корисувача аккаунтів:

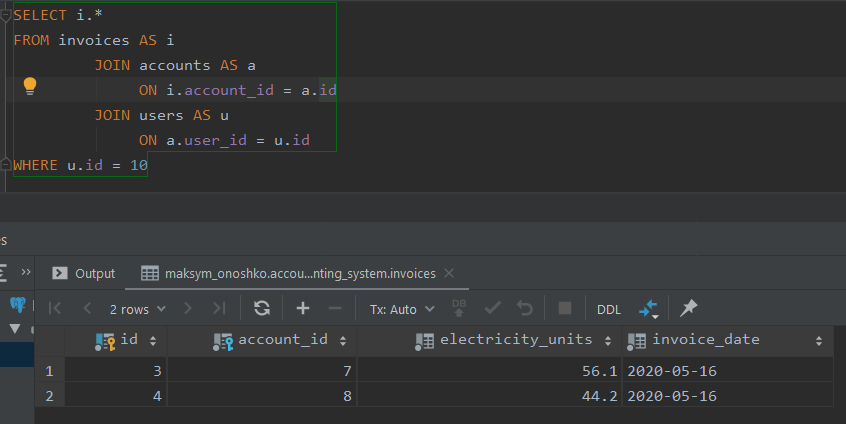


Рис. 2.2.2.3. – Запит UPDATE\_ACCOUNT

Запит GET\_USER\_ACCOUNTS, який отримує всі привязані до користувача аккаунти:

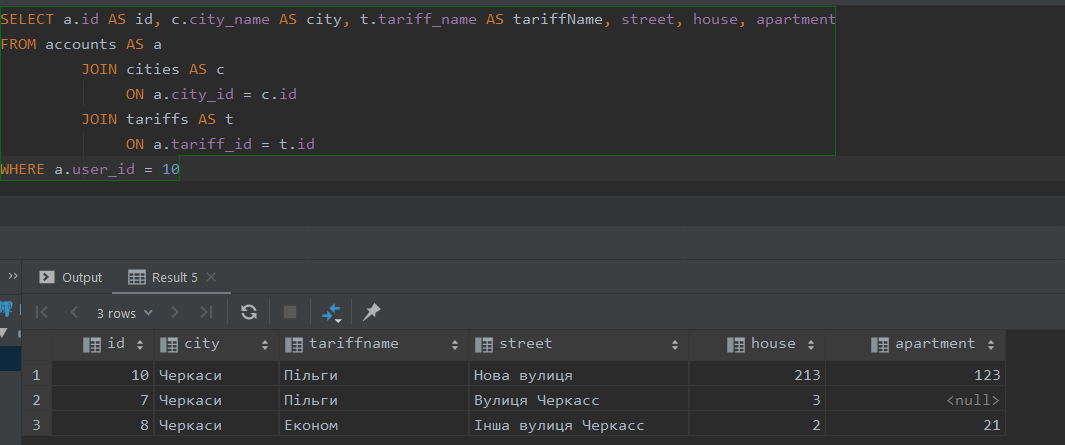


Рис. 2.2.2.4. – Запит GET\_USER\_ACCOUNTS

Запит GET\_USER\_DATA\_BY\_LOGIN, який отримує дані користувача по id:

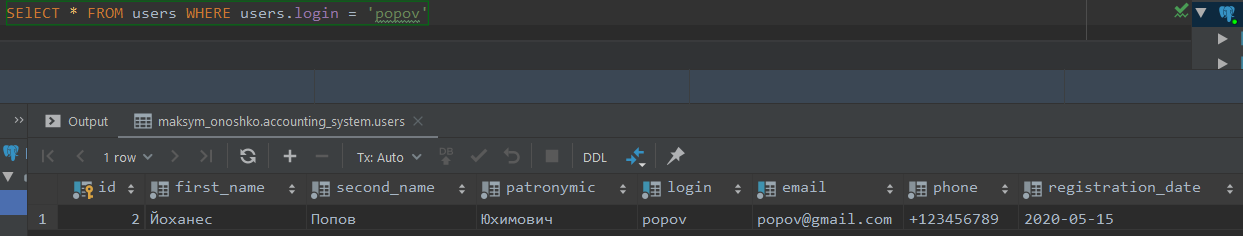


Рис. 2.2.2.5. – Запит GET\_USER\_DATA\_BY\_LOGIN

Запит GET\_INVOICES\_BY\_CITY\_ID, який отримує всі рахунки, які були виставлені користувачам в місті, яке вказується через id:

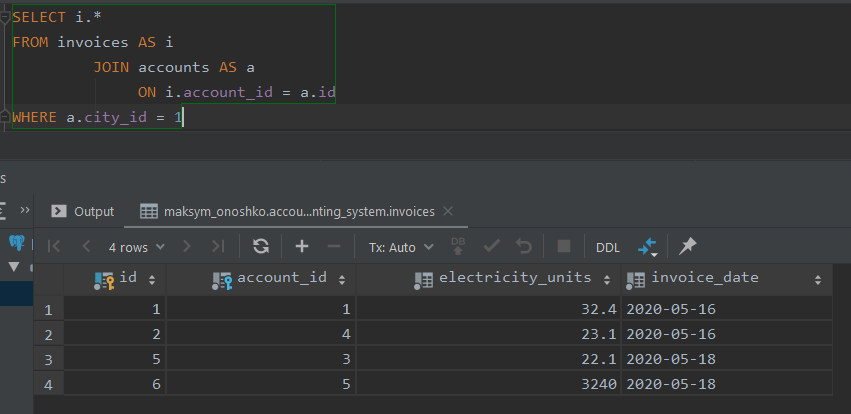


Рис. 2.2.2.6. – Запит GET\_INVOICES\_BY\_CITY\_ID

## 2.3. Опис клієнтського додатку

Програмний продукт орієнтований на два типа користувачів: індивідуальні користувачі електпоенергії та адміністрацію енерго компанії. Індивідуальний користувач електроенергії(клієнт) повинен мати змогу переглянути виставлені йому рахунки за електроенергію. Також користувач має можливість привязати до себе більше одної адреси, за яку він оплачує рахнки(аккаунти). Адміністрація енерго компанії може переглядати виставлені рахунки всіх користувачів, та сортувати їх по області, місту, вулиці міста і т.п. Адміністрація також має змогу виставляти рахунки користувачам. З ціллю безпеки клієнти не повинні мати доступ до панелі і функціоналу адміністрації, а лише до своїх даних.

Сторінка авторизації має вигляд:

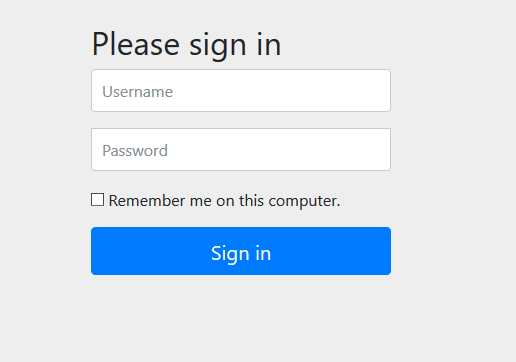


Рис. 2.3.1. – Сторінка авторизації

Після авторизації в залежності від того яку роль має користувач функціонал і інтерфейс користувачів будуть відрізнятися:

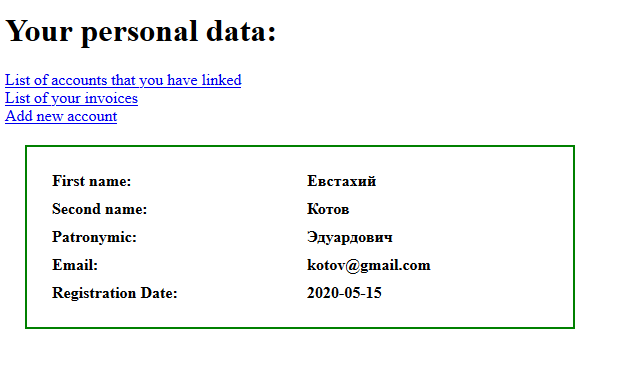


Рис. 2.3.2. – Домашня сторінка звичайного користувача

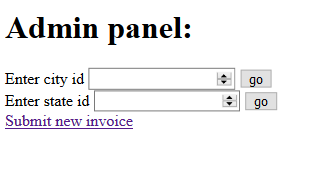


Рис. 2.3.3. – Панель адміністратора

Користувач може переглянути список своїх аккаунтів:



Рис. 2.3.4. – Сторінка аккаунтів користувача

Та подивитись виставлені цим аккаунтам рахунки по оплаті:

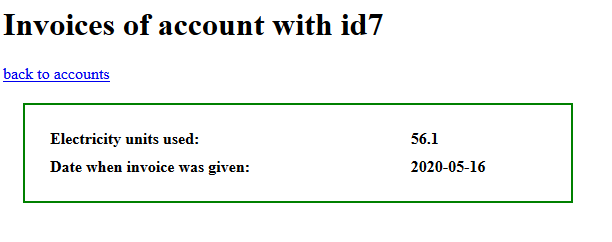


Рис. 2.3.5. – Сторінка рахунків виставлених аккаунту

Адміністрація може виставляти рахунки по id аккаунта:

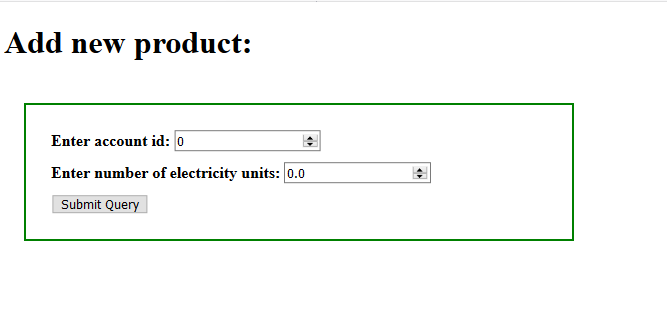


Рис. 2.3.6. – Сторінка виставлення нового рахунку по оплаті

Та дивитись всі рахунки по id міста або області:

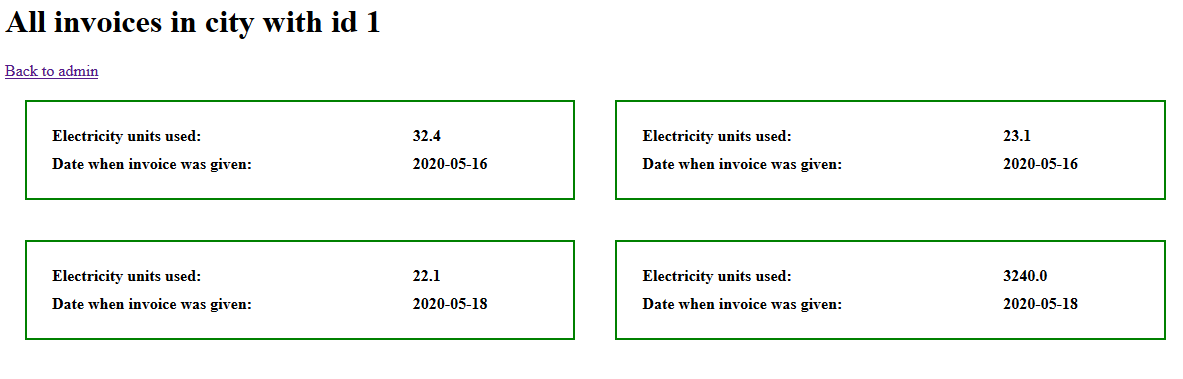


Рис. 2.3.7. – Сторінка з рахунками по місту з id 1

# 3. ВИСНОВКИ

З метою виконання курсової роботи та реалізації програмного продукту було досліджено предметну область за вказаною тематикою. Отриману інформацію було використано для проектування бази даних : створення таблиць, налаштування звязків між таблицями, їх нормалізації та коррекності полів цих таблиць.

Для написання бази даних було використано PostreSQL та H2 database. База даних складається з 9 таблиць. На отримуємі вхідні дані були створені обмеженя для того, щоб не порушувалась цілісніть бази даних.

Серверна частина додатку була реалізована за допомогою системи збирання проектів Gradle[1], мови програмування Java[2] та фреймворку Spring, а саме: Spring Boot[3], Spring Jdbc[4], Spring security[5].

Клієнська частина додатку була реалізована за допомогою html5, javascpript та thymeleaf[6].

В результаты було створено програмний продукт, який може допогти в веденні звітності індивідуальних користувачів електроенергії.

# Список літератури

1. Get started with Gradle // <https://docs.gradle.org/current/userguide/getting_started.html>

2. Java Documentation // <https://docs.oracle.com/en/java/>

3. Spring Boot Documentation //

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

4. Spring Data JDBC Documentation //

<https://docs.spring.io/spring-data/jdbc/docs/current/reference/html/#reference>

5. Spring Security Documentation //

<https://docs.spring.io/spring-security/site/docs/current/reference/html5/>

5. Thymeleaf documentation //

<https://www.thymeleaf.org/documentation.html>