Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 3

**З дисципліни « Інженерія програмного забезпечення »**

**на тему: “**Розробка серверної частини. Розробка

комунікації за протоколом TCP. Підключення

серверного модуля до БД.”

Варіант № 17.

Виконала: ст. гр. КІ-35

Обнявка С.Р.

Прийняв:

Цигилик Л.О.

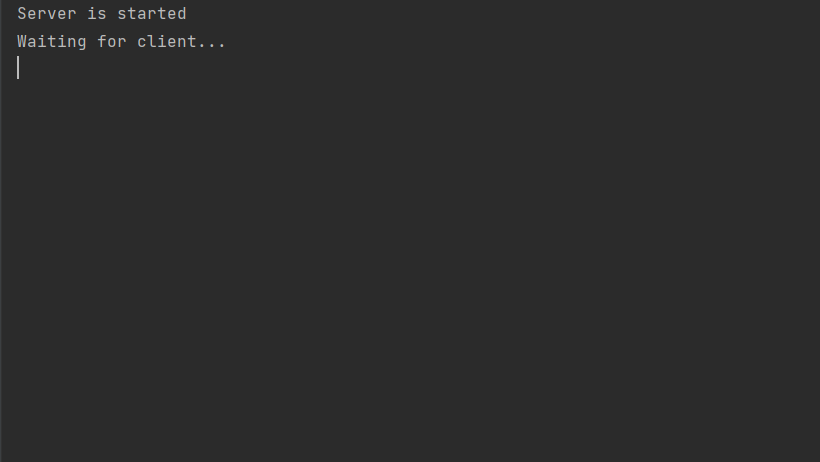
Львів – 2021

**Мета:** Розробити консольну аплікацію що буде підтримувати зв’язок по протоколу TCP/IP, отримувати дані та записувати у БД. Також, згідно деякої команди, вичитувати з БД необхідну інформацію та передавати по TCP протоколу на клієнтську частину.

**Завдання:** 17.Система моніторингу та ведення історії погоди.

Виконання:

Вигляд запущеної програми сервера:



*Рис. 1. Програма серверної частини*

Функція **public static void main(String[] a)** запускає сервер та очікує клієнта в безкінечному циклі. При появі клієнта виводиться повідомлення, що клієнт підключений. Під’єднання клієнта відбувається за допомогою функції **accept().**

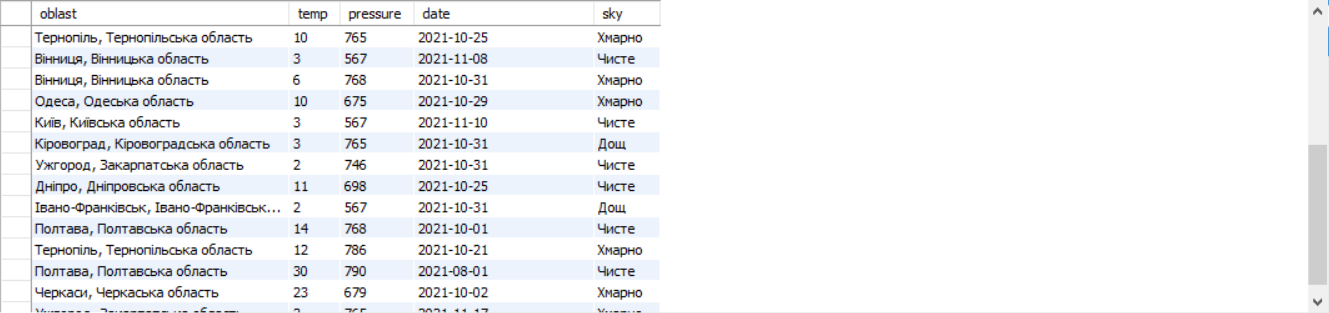
Функція **readLine()** читає дані, які вводить клієнт. Спочатку ця функція зчитує дані про те, у якому вікні знаходиться клієнт. В залежності від вікна будуть виконуватися наступні функції:

* **findForecast().** Дана функція приймає як параметр строку (назву міста) та шукає прогноз погоди на тепер та на наступних 7 днів. Клієнт вводить назву міста функція обробляє запит, надсилаючи додатковий запит на міжнародний сервер прогнозу погоди **Openweathermap**, та отримує дані. У випадку некоректності назви міста сервер надсилає помилку клієнту, що такого міста немає
* **findInfo()**. Дана функція приймає як параметри місто та дату та шукає погоду на заданий час та задане місто у базі даних. Якщо дані на обрані критерії не вдалося знайти, клієнту повертається “null” значення, що в свою чергу викликає віконне повідомлення на клієнтській частині.
* **authorize()**. Дана функція приймає два параметри (дві строки) – логін та пароль, за якими адміністратор може увійти, щоб додати історію погоди. Робиться пошук у базі даних, чи наявний такий адміністратор з таким логіном та паролем. Якщо такий адміністратор існує, тоді клієнту надсилається строка “admin”, що дозволяє перейти на вікно вводу у клієнтській частині. В іншому випадку надсилається строка “notAdmin” та виводиться вікно з повідомленням у клієнтській частині.
* **addWeather()**. Дана функція приймає 5 параметрів(5 строк) – область, температура, тиск, дата, небо. Додає дані у базу даних.

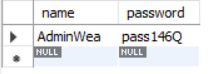
Функція **out.println()** повертає дані клієнту.

Функція **getUrlContent()** приймає як параметр адресу та здійснює запити до сайтів/серверів.

**Вигляд бази даних:**



*Рис. 2. База даних історії погоди.*



*Рис. 3. База даних адміністраторів.*

**Зчитування з БД:**

1. **Пошук погоди.**

public ArrayList<String> findInfo(String text, String dateInput) throws SQLException, ClassNotFoundException {

ArrayList<String> list = new ArrayList<>();

Statement statement = getConnection().createStatement();

ResultSet search = statement.executeQuery("SELECT temp, pressure, sky FROM weather.weather\_history WHERE oblast = '" + text + "' AND date = '" + dateInput + "'");

if(search == null){

return null;

}

while (search.next()) {

list.add(search.getString("temp"));

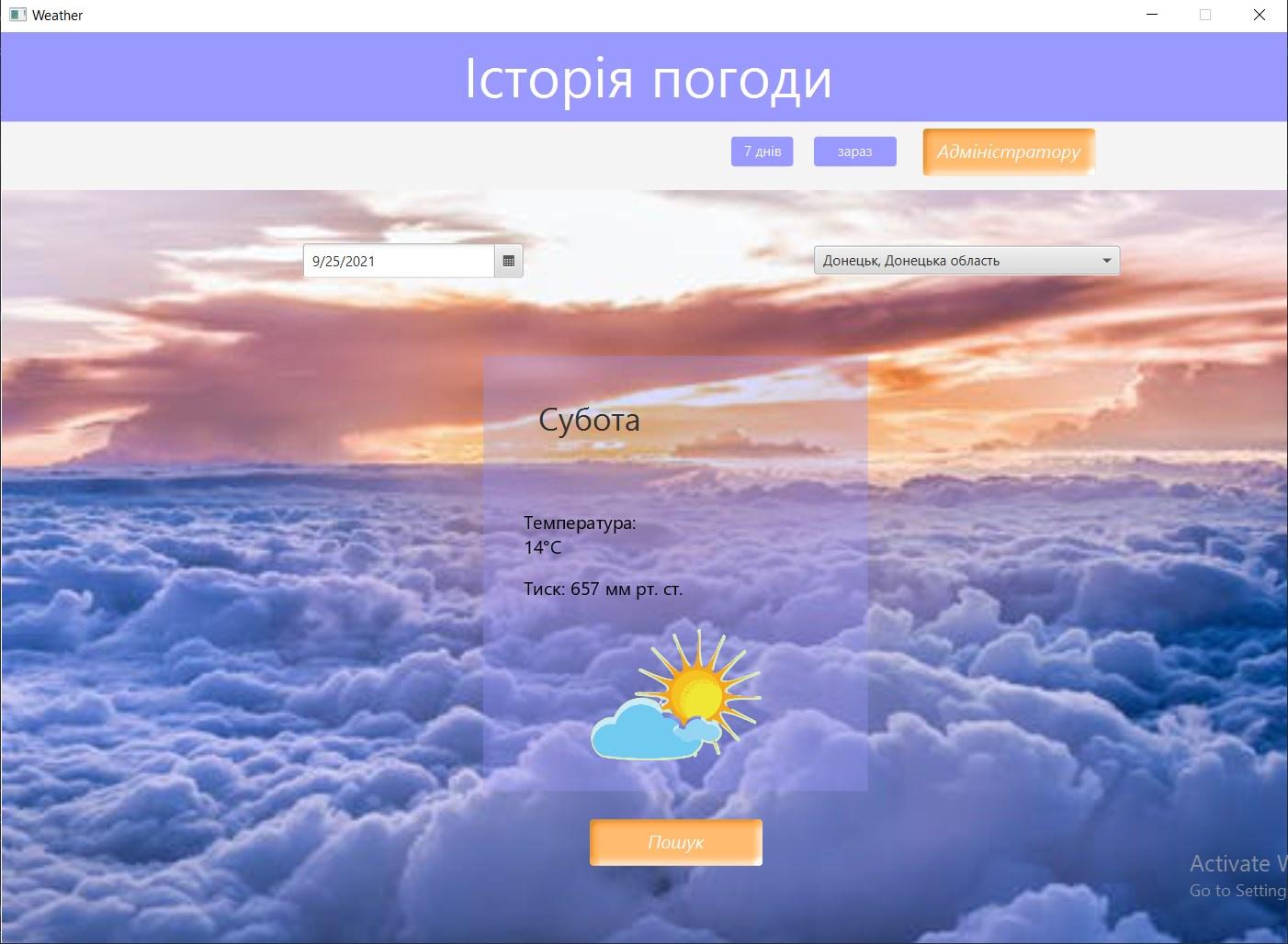
list.add(search.getString("pressure"));

list.add(search.getString("sky"));

}

return list;

}



*Рис. 4. Вікно програми клієнта. Пошук історії погоди.*



*Рис. 5. Вікно серверної частини. Пошук історії погоди.*

1. **Авторизація адміністратора**

public String authorize(String name, String password) throws SQLException, ClassNotFoundException {

Statement statement = getConnection().createStatement();

ResultSet admin = statement.executeQuery("SELECT name, password FROM weather.moderator");

while (admin.next()) {

String dataAdmin;

String dataPass;

dataAdmin = admin.getString("name");

dataPass = admin.getString("password");

if (dataAdmin.equals(name) && dataPass.equals(password)) {

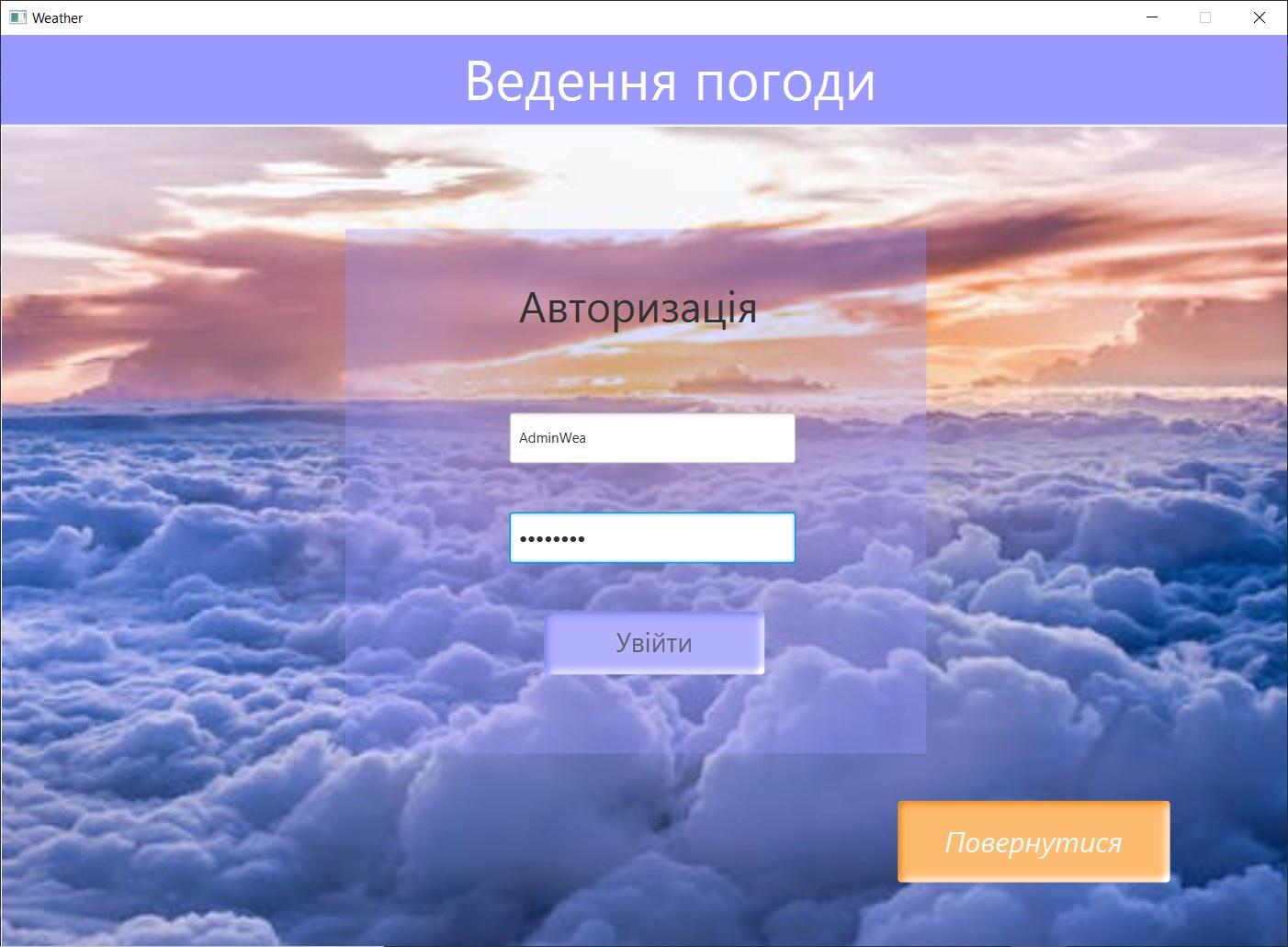
return "admin";

}

}

return "notAdmin";

}



*Рис. 6. Вікно програми клієнта. Авторизація адміністратора.*

https://i.imgur.com/nISU6Lx.png

*Рис. 7. Вікно серверної частини. Авторизація адміністратора.*

**Запис в БД:**

public void addWeather(String text,String temp,String pressure, String date, String sky) throws SQLException, ClassNotFoundException {

String insert = "INSERT INTO weather.weather\_history(oblast, temp, pressure, date, sky)"

+ "VALUES(?,?,?,?,?)";

PreparedStatement prST = getConnection().prepareStatement(insert);

prST.setString(1, text);

prST.setInt(2, Integer.parseInt(temp));

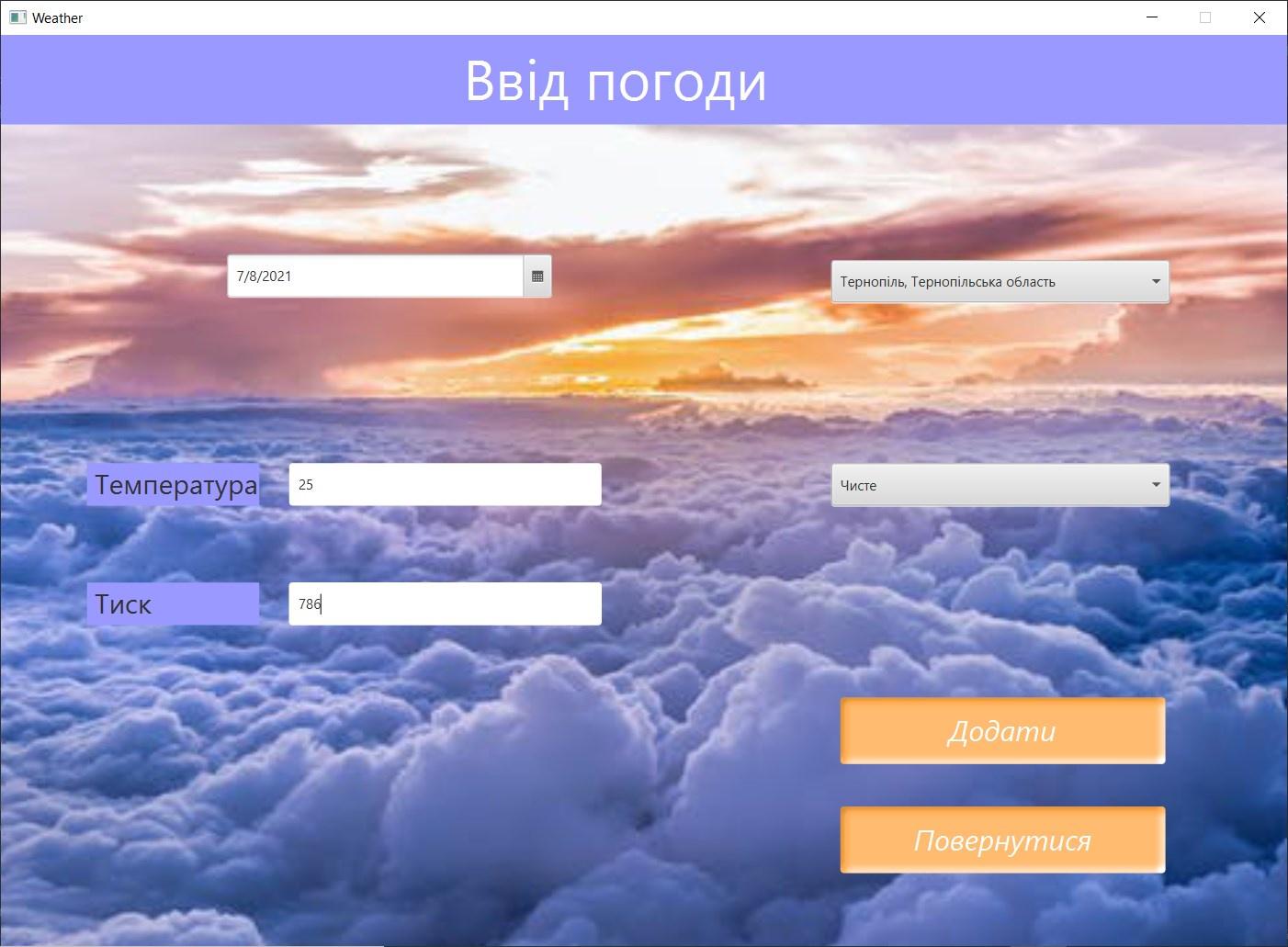
prST.setString(3,pressure);

prST.setString(4,date);

prST.setString(5,sky);

prST.executeUpdate();

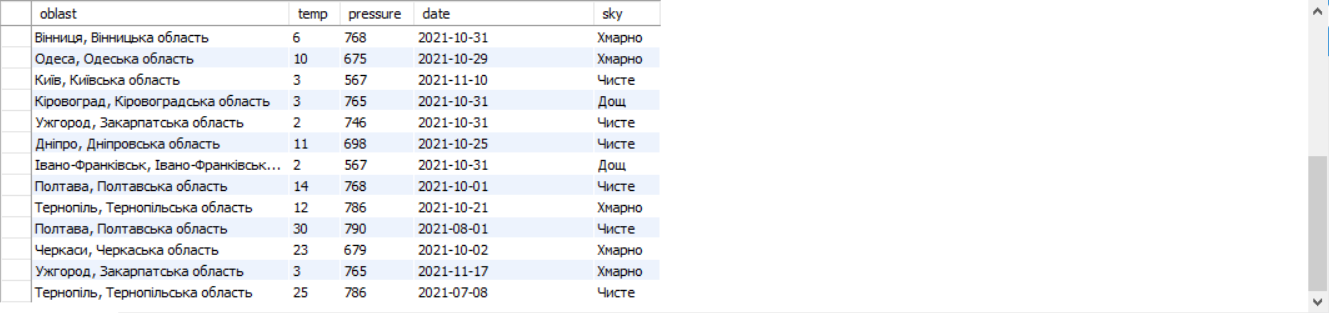
}



*Рис. 8. Вікно програми клієнта. Ввід погоди.*



*Рис. 9. Вікно серверної частини. Ввід погоди.*



*Рис. 10. Результат додавання даних до БД.*

Пошук прогнозу погоди на Openweathermap:

public static ArrayList<String> findForecast(String city) {

ArrayList<String> forecast = new ArrayList<>();

String outputToCheck = URLContent.getUrlContent("http://api.openweathermap.org/data/2.5/weather?q=" + city + "&appid=2cbefecfc08bdc704d243280d9a5599b");

String output = URLContent.getUrlContent("http://api.openweathermap.org/data/2.5/forecast?q=" + city + "&appid=2cbefecfc08bdc704d243280d9a5599b");

if(Objects.equals(outputToCheck, "null")){

return null;

}

JSONObject obj = new JSONObject(output);

JSONArray weatherObject = obj.getJSONArray("list");

for (int i = 0; i < 40; i+=6) {

forecast.add(String.valueOf(Integer.parseInt(weatherObject.getJSONObject(i).toString().substring(weatherObject.getJSONObject(i).toString().indexOf("temp") + 4,weatherObject.getJSONObject(i).toString().indexOf("feels")).replaceAll("\\D+", "").substring(0,3))-273));

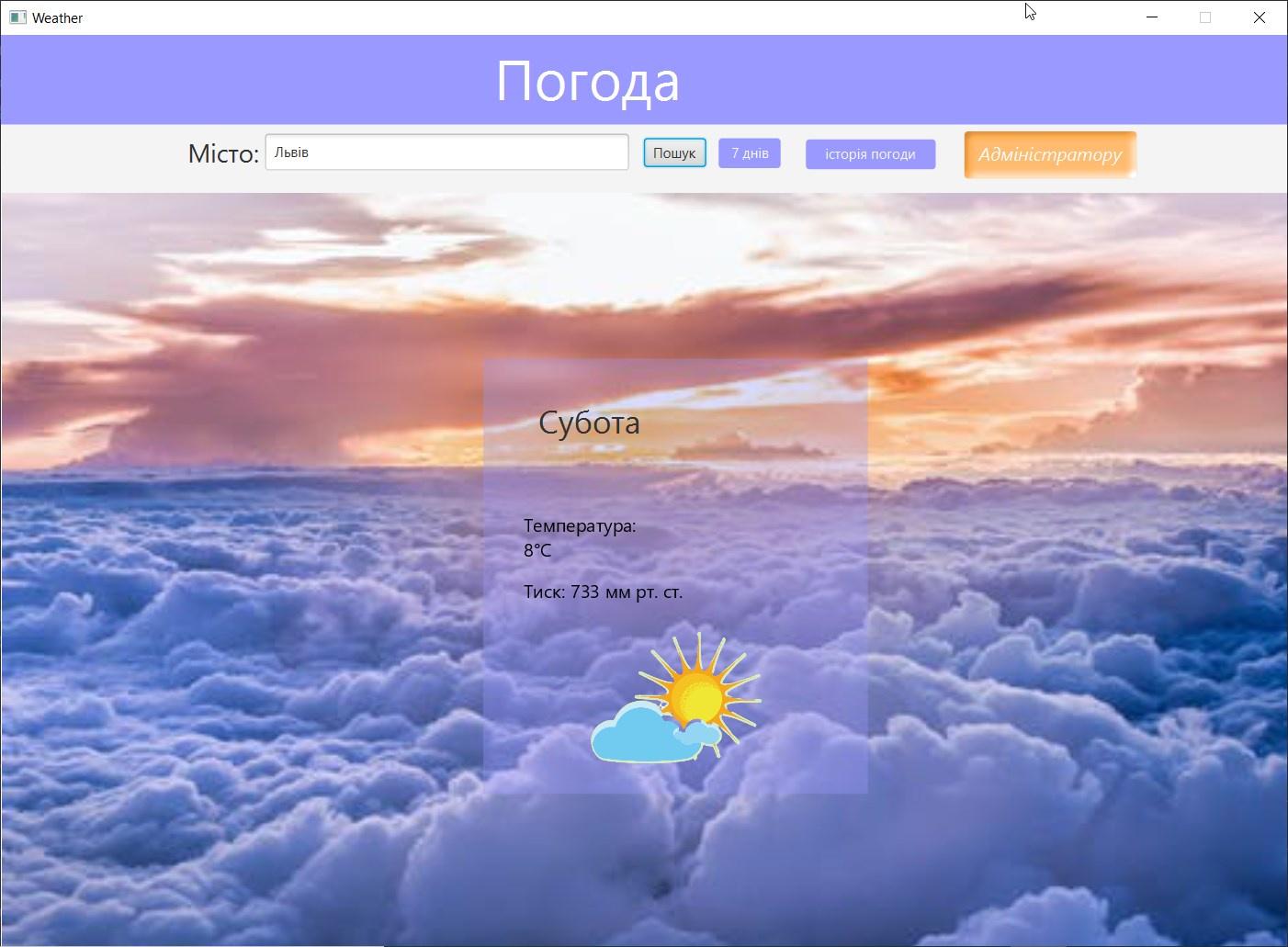
forecast.add(String.valueOf(Integer.parseInt(weatherObject.getJSONObject(i).toString().substring(weatherObject.getJSONObject(i).toString().indexOf("pressure") + 4,weatherObject.getJSONObject(i).toString().indexOf("sea\_level")).replaceAll("\\D+", ""))-282));

forecast.add(weatherObject.getJSONObject(i).toString().substring(weatherObject.getJSONObject(i).toString().indexOf("main") + 6,weatherObject.getJSONObject(i).toString().indexOf("id")).replace(",", "").replace("\"", ""));

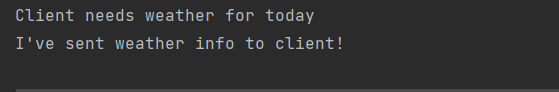
}

return forecast;

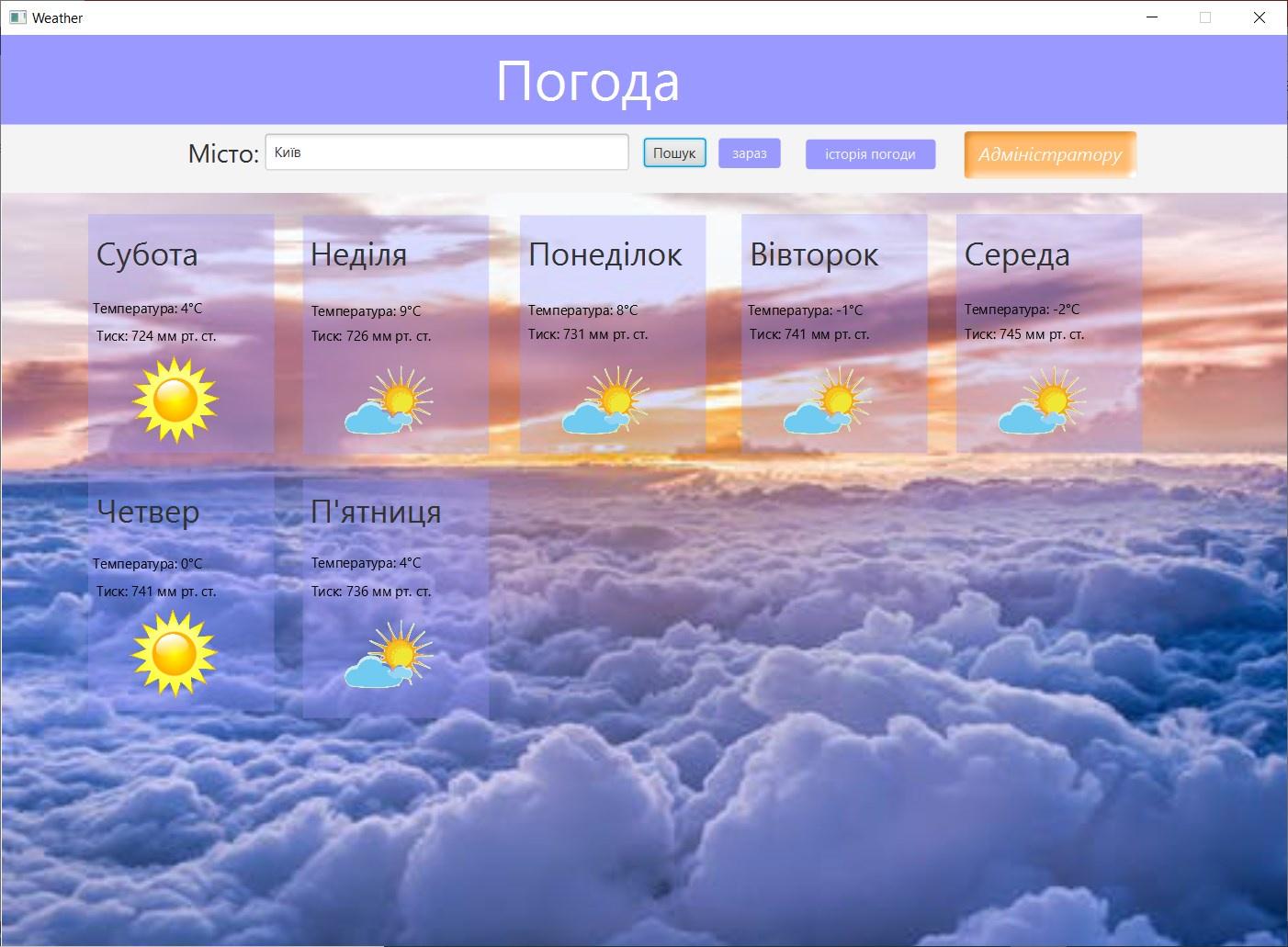
}



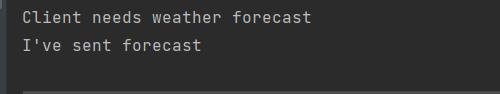
*Рис. 10. Вікно програми клієнта. Пошук погоди на сьогодні.*



*Рис. 11. Вікно серверної частини. Пошук погоди на сьогодні.*



*Рис. 12. Вікно програми клієнта. Пошук прогнозу погоди на сьогодні.*



*Рис. 13. Вікно серверної частини. Пошук прогнозу погоди на сьогодні.*

**Висновок:** на даній лабораторній роботі я розробила консольну аплікацію що буде підтримувати зв’язок по протоколу TCP/IP, отримувати дані та записувати у БД. Також, згідно деякої команди, вичитувати з БД необхідну інформацію та передавати по TCP протоколу на клієнтську частину.