CS 570 Programming Project

Weather Analytic System

By

Name: Ahmed Al Obaidi

CIN: 302605529

Name: Jordan Ton-That

CIN: 302586237

Name: Bhagyeshree Gaikwad

CIN: 304360204

**Requirements**

1. We should have “WAMPServer” to have our database in it. Then we have to create a user for the database called “ahmed” and the password is “1234”. We create a database using the file “sql.txt” in project file “Data Aggregator”.
2. We need to use some libraries to execute our system, so we build our project paths by adding them. The first is “mysql-connector-java-5.1.33-bin.jar”; we add it to “Data Aggregator and Analytics Server”. The second is “json-simple-1.1.1.jar”; added to “Analytics Server, WeatherTrafficSocket, WeatherWebsiteRPC, and WheatherWbsite”. The third is “javax.servlet-api-3.0.1.jar” and the fourth is “jstl-1.2.jar”; we add them to “WeatherWebsiteRPC and WheatherWbsite”; we should make sure to put them under the “lib” files.
3. We should have “eclipse” to run the system.

**Programming Languages**

1. Java.
2. JavaScript.
3. HTML.
4. CSS.
5. MySQL.

**Steps to Run the Distributed System**

1. Run “WAMPServer” to make the database available.
2. Run the socket in “WeatherTrafficSocket” project by executing the “SocketServer” class.
3. Start the RMI registry, but first we have to go to the “bin” file directory in “Data Aggregator” project using command line. Then we type “start rmiregistry”.
4. Run the RMI server in “Data Aggregator” project by executing the “MainController\_RMIserver” class.
5. Run the RMI client in “WheatherWbsite” project by executing “WeatherDisplayInfo” servlet.
6. Run the RPC server in “Analytics Server” by executing “WebServiceServerRPC\_Publisher” class.
7. Run the RPC client in “WeatherWebsiteRPC” by executing “WeatherAnalyticsDisplay” servlet.

**Required Modules**

1. Weather Info Website (WeatherTrafficSocket)

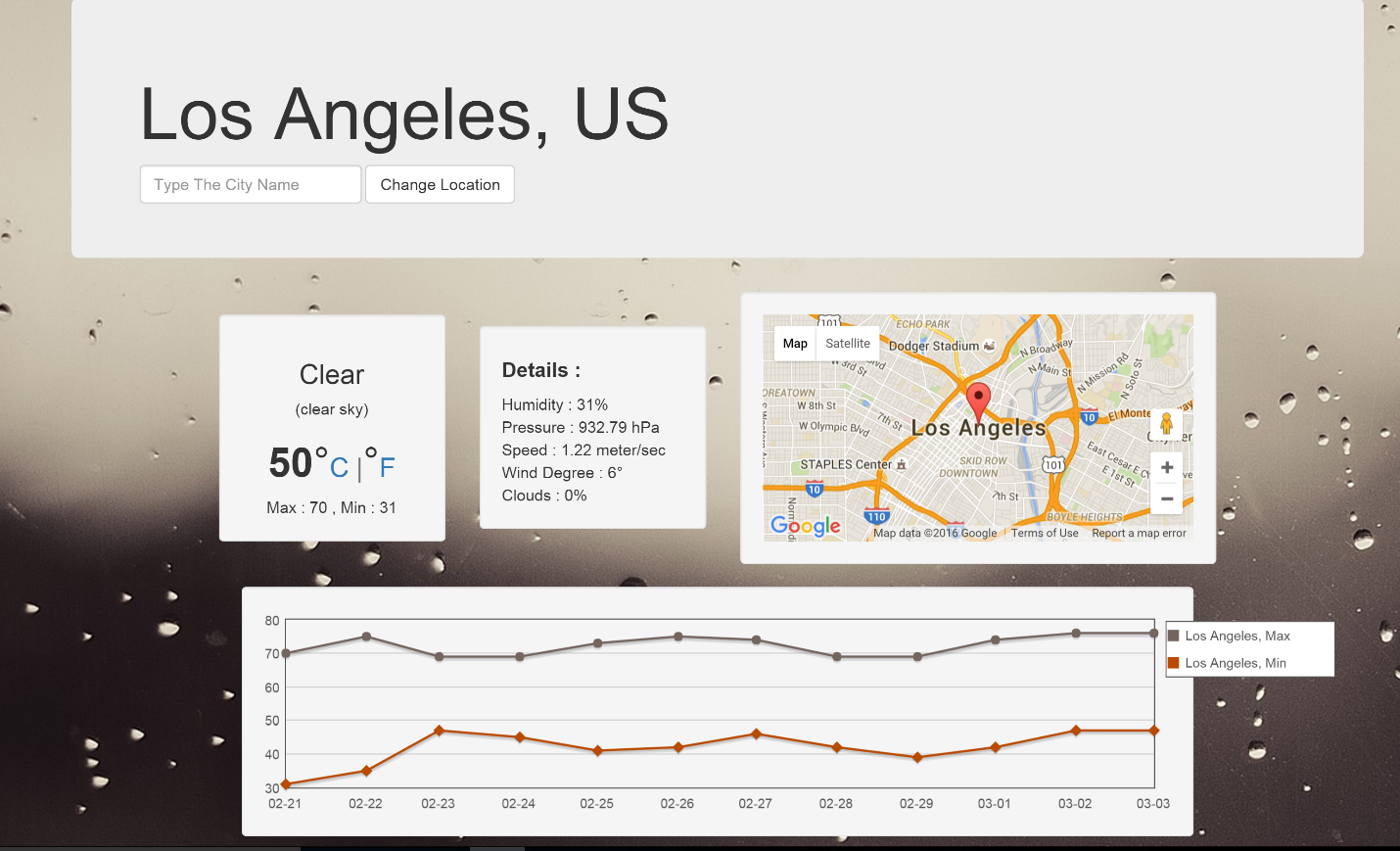
* We created two model classes called “CityWeather” and “Temperature”, and *we put it in all the other steps*. We implement the Serializable interface to pass the object through the socket.
* We create two controllers called “SocketServer” and “GetData”. The “SocketServer” is waiting for any request to get new data from “GetData”, and then send back to the client the weather info. The “GetData” gets the weather info from the weather API.

1. RMI Weather Server (Data Aggregator)

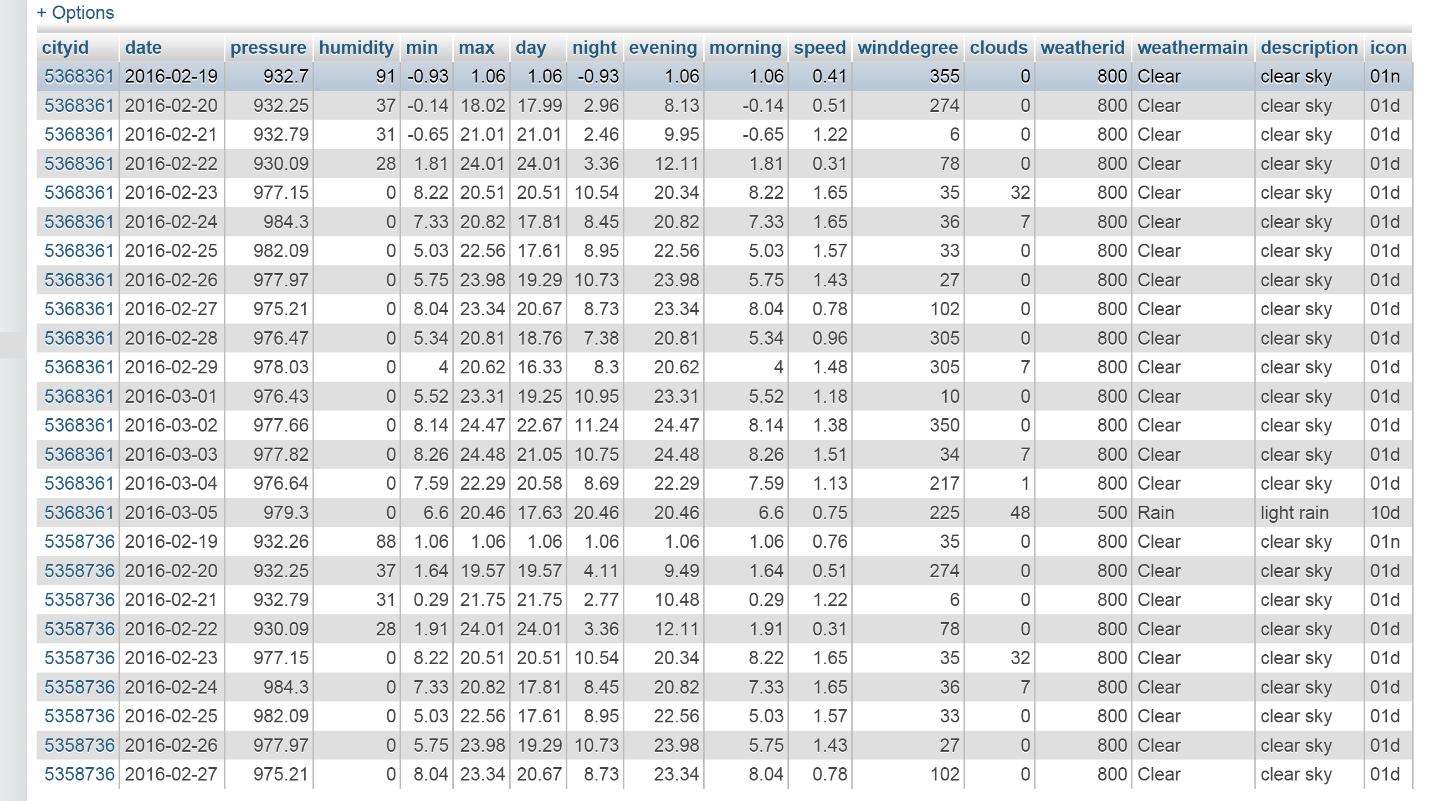
* We create a RMI interface
* We create two controllers called “DB\_Controller” and “MainController\_RMIserver”. We need the “DB\_Controller” to access, add, update, and get the weather info for the clients. We need the “MainController\_RMIserver” to send the RMI clients the required weather info.

1. RMI Weather Client (WheatherWbsite)

* We create a RMI interface
* We create two controllers called “CurrentLocation” and “WeatherDisplayInfo”. The “CurrentLocation” gives us the current location and the time zone id. *We use the time zone id to specify the current weather based on the time zone of that particular city.* The “WeatherDisplayInfo” displays all the important weather info in details along with a graph that has the maximum and minimum temperature for 12 days.



1. Database

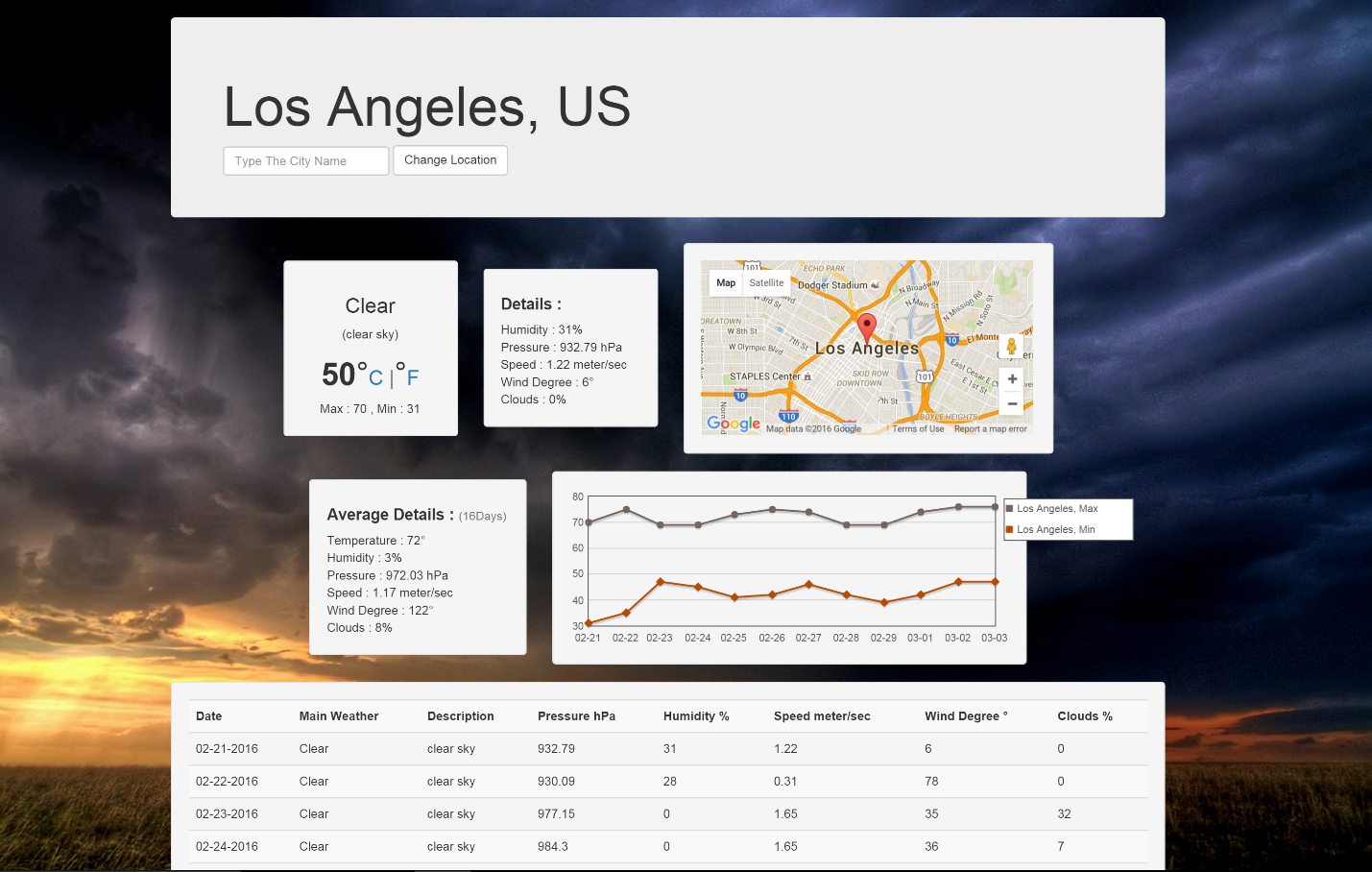
* We create a database using SQL script. The database contains two tables called “City” and “Temperature”.
* The City table has cityID, cityName, country, cityLocationLatitude, and cityLocationLongitude. We use cityID as primary key.
* The Temperature table has cityID, date, pressure, humidity, min, max, day, night, evening, morning, speed, windDegree, clouds, weatherID, weatherMain, description, and icon. We use cityID as a foreign key.

1. Weather Analytic Server (Analytics Server)

* We create an RPC interface
* We create an RPC server to send the clients the weather info when requested
* We create one controllers called “DB\_Controller”. We need the “DB\_Controller” to access, add, update, and get the weather info for the clients.

1. Weather Service Client (WeatherWebsiteRPC)

* We create an RPC interface
* We create two controllers called “CurrentLocation” and “WeatherAnalyticsDisplay”. The “CurrentLocation” gives us the current location and the time zone id. *We use the time zone id to specify the current weather based on the time zone of that particular city.* The “WeatherAnalyticsDisplay” displays all the important weather info in details along with a graph that has the maximum and minimum temperature for 12 days. In addition, it has the average temperature, humidity, speed, windDegree, pressure, and clouds. It also shows a table of all the important weather info for 16 days.



**Contributor Links**

<http://ip-api.com/json>

<https://maps.googleapis.com/maps/api/timezone/json>

<http://getbootstrap.com/>

<https://maps.googleapis.com/maps/api/js>

**Additional Features**

* We included a dynamic map of the current location and nearby cities.
* We included the Celsius to Fahrenheit and Fahrenheit to Celsius conversions.
* We included the current temperature based on current location and time zone for that particular city.
* We included a detailed graph of minimum and maximum temperature for the entire 12 days.
* We included interactive user interface.