ASOIS Project

Weather Android Application

Ciauș Gabriel Nicolae,
Service Engineering and Management, 2017

Table of contents

1. General presentation

The purpose of this project was to create an Android mobile application that allows user to view data provided by different weather sensors. You can choose between two different view types: one in real time and another that show evolution during a period of time. Presented that should be ones that are used by the meteorology: temperature, humidity, wind speed and atmospherically pressure. The data provided by the sensors are stored in a MySQL database. The application makes a periodically request to fetch data from database in the mobile application. The application should show temperature, humidity, speed wind and atmospherically pressure for different cities on a map.

What I was able to do as a part of this project was to create an application that presents different meteorology information for a city provided by the user in a text field. It can show information about:

- Temperature in Celsius degrees.
- Wind speed in meters per second
- Information about condition like: snow, cloud, sunny etc.
- Atmospherically pressure in hPA.
- Humidity
- Sunrise time
- Sunset time

For that I used a weather API from openweathermap.org to gain information about current weather data in different city. This is and simple and free API that uses JSON for data transmission.

The mobile application lets user to input of any city that is supported by API and can show weather information from that city in real time.

For Bucharest I also implemented a simple MySQL data base that store those information in a table. Every time when a user choose Bucharest as city, beside that application can provide him with weather information, it also insert this information in database.

When the city is Bucharest, on the main page of application user can see a link to another activity, where he or she can see all information stored in database.

2. Technology used in application

In this project I used different technology like:

- Android SDK
- Weather API
- JSON
- Volley
- MYSQL

Android SDK

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. It is purpose built for Android to accelerate your development and help you build the highest-quality apps for every Android device. Android provides a rich application framework that allows you to build innovative apps and games for mobile devices in a Java language environment.

Android apps are built as a combination of distinct components that can be invoked individually. Android apps are written in the Java programming language. The Android SDK tools compile your code along with any data and resource files into an APK, an Android package, which is an archive file with an .apk suffix. One APK file contains all the contents of an Android app and is the file that Android-powered devices use to install the app.

Android provides an adaptive app framework that allows you to provide unique resources for different device configurations. For example, you can create different XML layout files for different screen sizes and the system determines which layout to apply based on the current device's screen size.

App components are the essential building blocks of an Android app. Each component is an entry point through which the system or a user can enter your app. Some components depend on others.

There are four different types of app components:

- Activities - an activity is the entry point for interacting with the user. It represents a single screen with a user interface.
- Services - a service is a general-purpose entry point for keeping an app running in the background for all kinds of reasons. It is a component that runs in the background to perform long-running operations or to perform work for remote processes.
- Content providers
- Broadcast receivers - a broadcast receiver is a component that enables the system to deliver events to the app outside of a regular user flow, allowing the app to respond to system-wide broadcast announcements.

Weather API

OpenWeatherMAP API is a simple, clear and free API that provides support for application. It provides access to weather information for any location including 200.000 cities and data are updated by almost 40.000 weather stations. It can provide data in JSON, XML or HTML format.

Weather information include precipitation, clouds, pressure, temperature, wind and more. This API can also provide the user with historical data about weather condition from the last year and with a daily forecast for next days (up to two weeks).

It can be used to create triggers which will be working upon occurrence of specified weather conditions in a certain period of time.

JSON

JSON stand for JavaScript Object Notation and it represents a syntax for storing and exchanging data. Because in the moment of exchanging data between a browser and a server, the data can be only text, JSON is text and can be converted to any JavaScript object into JSON and send it to the server or vice versa by converting JSON received from server into JavaScript object.

JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

Volley

Volley is an HTTP library that makes networking for Android apps easier and most importantly, faster. Volley is available through the open AOSP repository. Volley excels at operations used to populate a UI, such as fetching a page of search results as structured data. It integrates easily with any protocol and comes out of the box with support for raw strings, images, and JSON.

Android Volley is a networking library was introduced to make networking calls much easier, faster without writing tons of code. By default all the volley network calls works asynchronously, so we don’t have to worry about using async task anymore.

Volley offers the following benefits:

- Automatic scheduling of network requests.
- Multiple concurrent network connections.
- Transparent disk and memory response caching with standard HTTP cache coherence.
- Support for request prioritization.
- Cancellation request API. You can cancel a single request, or you can set blocks or scopes of requests to cancel.
- Ease of customization, for example, for retry and backoff.
- Strong ordering that makes it easy to correctly populate your UI with data fetched asynchronously from the network.

MySQL

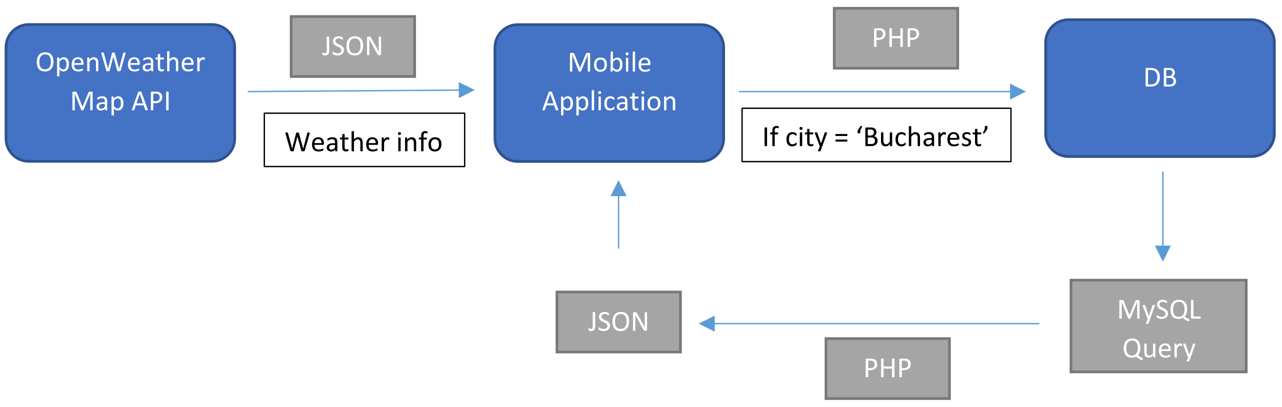
MySQL is an open-source relational database management system (RDBMS). MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

A Relational DataBase Management System (RDBMS) is a software that:

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

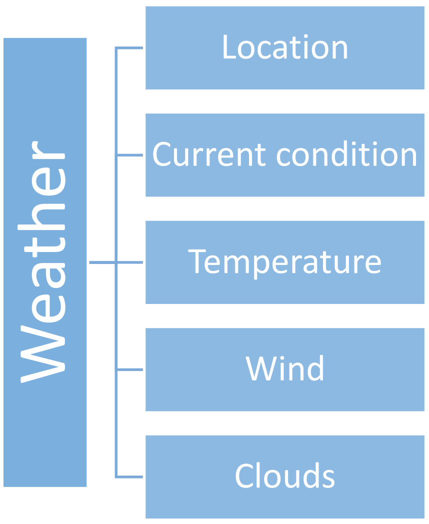
3. Application description

Application data flow:



The application main activity is used to create the homepage of weather application and to render the weather information. A task that runs in background is used to fetch information from OpenWeatherMap API and the image.

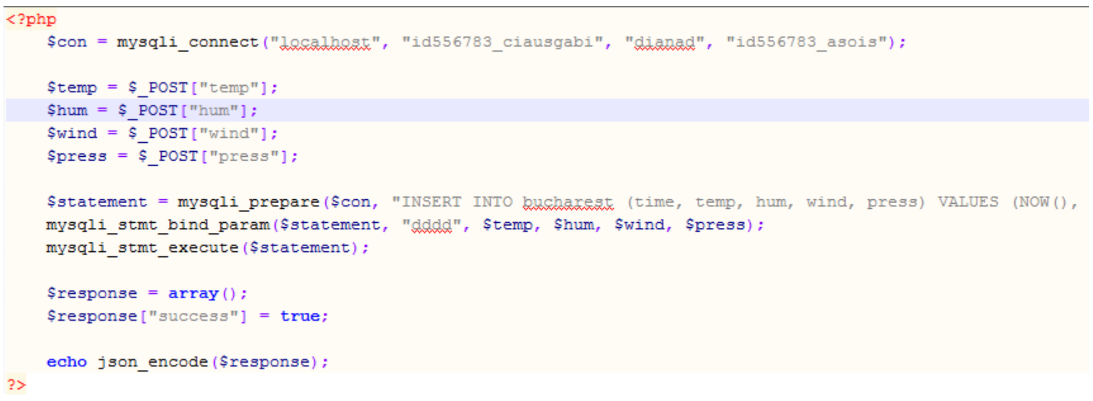
The data is transmitted in JSON format, and for that I created a JSON parser (JSONWeatherParser class) in order to get JSON object from API and set the members of weather Object. The weather object is modelled using different classes.



Default city is Bucharest, RO and it is set using CityPreferences class. This class is used also to store selected city.

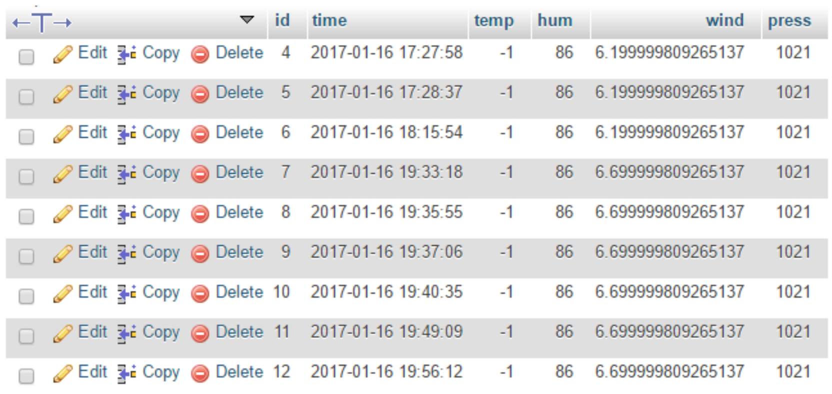
When this activity is executed, after data is fetch from the server, it will show data in the GUI. We can see information like: city name, current condition from that city, temperature, wind speed, humidity, atmospherically pressure, sunset time, sunrise time and the date of the last update time. If the selected city is Bucharest, the user can also see a link to another activity where he or she can see historical information for that city.

Also if selected city is Bucharest, main activity will make an insert request into database. For that I created InsertRequest class. Here I created a map with the fields that I want to store in data base and insert.php script. This script is used to create a database connection MySQL server on a webhosting platform and to execute a simple MySQL query to insert data into table. In database I store information about: time, temperature, humidity, wind speed and atmospherically pressure.



```
<?php
    $con = mysqli_connect("localhost", "id556783_ciausgabi", "dianad", "id556783_asois");

    $temp = $_POST["temp"];
    $hum = $_POST["hum"];
    $wind = $_POST["wind"];
    $press = $_POST["press"];

    $statement = mysqli_prepare($con, "INSERT INTO bucharest (time, temp, hum, wind, press) VALUES (NOW(),
    mysqli_stmt_bind_param($statement, "dddd", $temp, $hum, $wind, $press);
    mysqli_stmt_execute($statement);

    $response = array();
    $response["success"] = true;

    echo json_encode($response);
?>
```

Stored data in database:

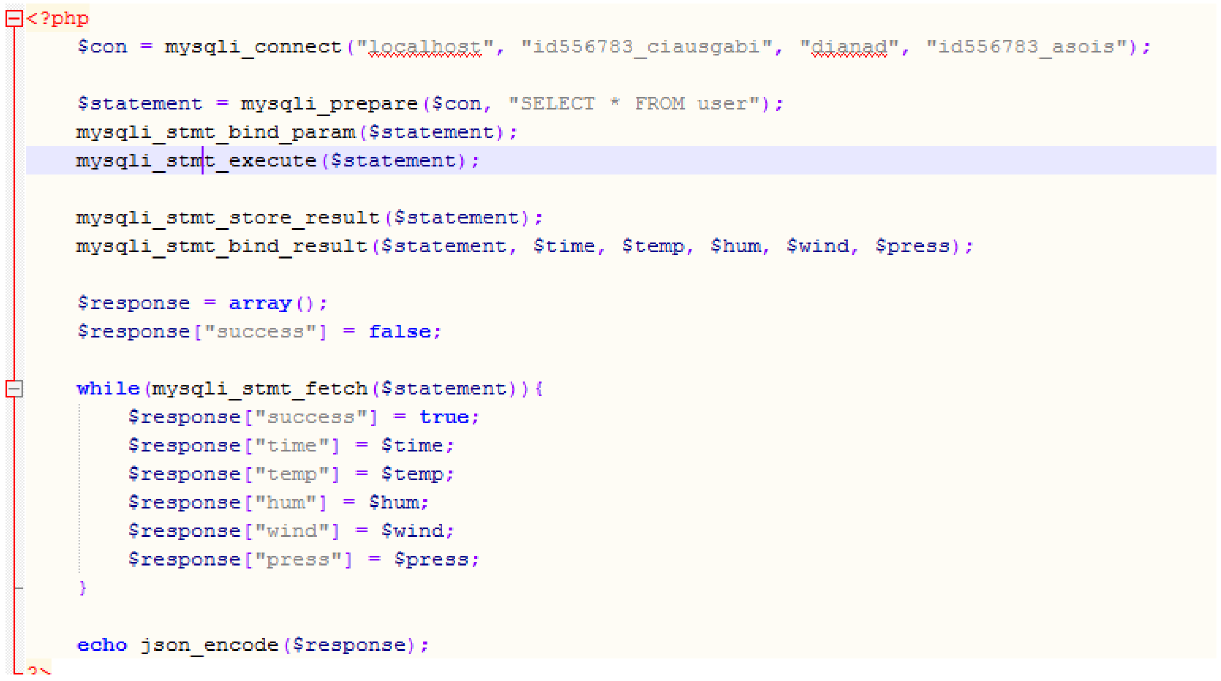


	id	time	temp	hum	wind	press
Edit Copy Delete	4	2017-01-16 17:27:58	-1	86	6.199999809265137	1021
Edit Copy Delete	5	2017-01-16 17:28:37	-1	86	6.199999809265137	1021
Edit Copy Delete	6	2017-01-16 18:15:54	-1	86	6.199999809265137	1021
Edit Copy Delete	7	2017-01-16 19:33:18	-1	86	6.699999809265137	1021
Edit Copy Delete	8	2017-01-16 19:35:55	-1	86	6.699999809265137	1021
Edit Copy Delete	9	2017-01-16 19:37:06	-1	86	6.699999809265137	1021
Edit Copy Delete	10	2017-01-16 19:40:35	-1	86	6.699999809265137	1021
Edit Copy Delete	11	2017-01-16 19:49:09	-1	86	6.699999809265137	1021
Edit Copy Delete	12	2017-01-16 19:56:12	-1	86	6.699999809265137	1021

As I said, if selected city is Bucharest, an additional link appear in main page, to allow user to go to a second activity that shows historical data.

For that we need to fetch data from database back into our application. For that I created GraphActivity where a request to database is sent. Also this is used to show listView used to print each row of table.

ParseJSON class is needed to parse JSON object provided by the PHP script that is used to select all rows in the table Bucharest.

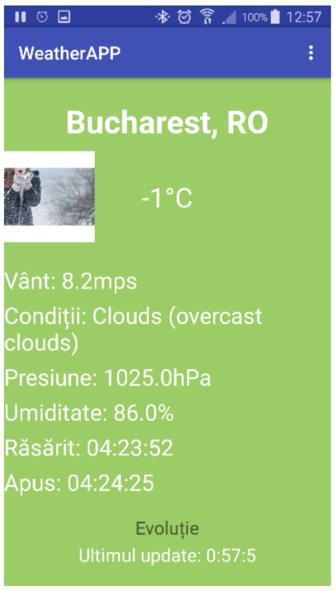


```php
<?php
    $con = mysqli_connect("localhost", "id556783_ciausgabi", "dianad", "id556783_asois");

    $statement = mysqli_prepare($con, "SELECT * FROM user");
    mysqli_stmt_bind_param($statement);
    mysqli_stmt_execute($statement);

    mysqli_stmt_store_result($statement);
    mysqli_stmt_bind_result($statement, $time, $temp, $hum, $wind, $press);

    $response = array();
    $response["success"] = false;

    while(mysqli_stmt_fetch($statement)){
        $response["success"] = true;
        $response["time"] = $time;
        $response["temp"] = $temp;
        $response["hum"] = $hum;
        $response["wind"] = $wind;
        $response["press"] = $press;
    }

    echo json_encode($response);
?>
```

Each JSON field is stored in string arrays by their key in CustomList class. This class is also used to set information for each item in ListView.

4. GUI

Application main page:



Bucharest, RO

-1°C

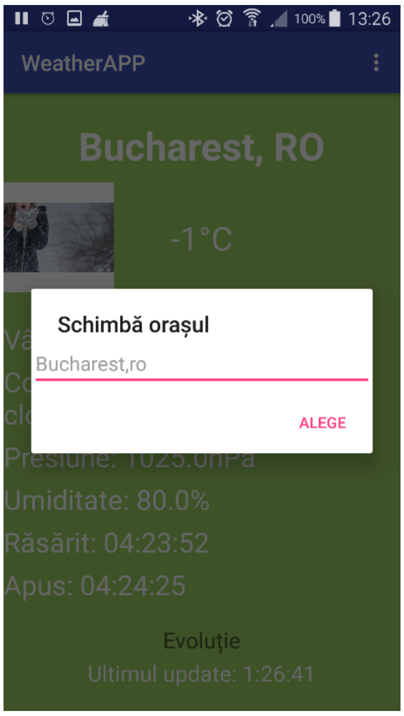
Vânt: 8.2mps
Condiții: Clouds (overcast clouds)
Presiune: 1025.0hPa
Umiditate: 86.0%
Răsărit: 04:23:52
Apus: 04:24:25

Evoluție
Ultimul update: 0:57:5

As you can see this page provides user information about city name, temperature, wind speed, meteorological information about current condition, atmospheric pressure, humidity sunset time, sunrise time and the time of the last update.

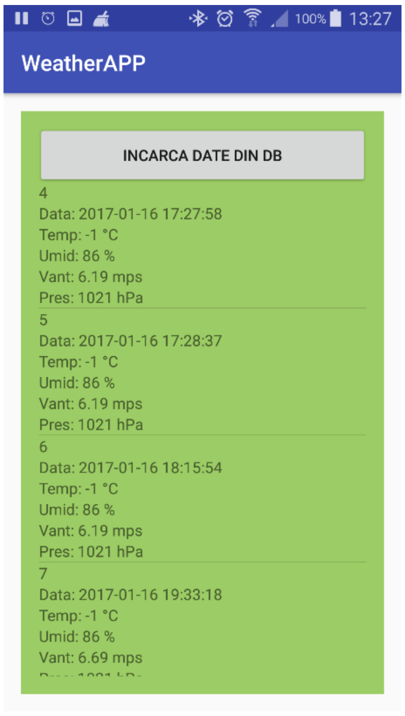
Because city is Bucharest, you can see the link for the historical evolution for Bucharest city. If we choose another city, this link will be invisible.

In the right corner we can change location.



Because the application uses and extern API, you can view weather information for about 200.000 cities.

If you press Evolutie button, another page will be opened (another activity will be executed). This page will allow user to see historical weather information for Bucharest city. These information are stored in a MySQL database and are fetched back in application a presented as a list view.



```
4
Data: 2017-01-16 17:27:58
Temp: -1 °C
Umid: 86 %
Vant: 6.19 mps
Pres: 1021 hPa
5
Data: 2017-01-16 17:28:37
Temp: -1 °C
Umid: 86 %
Vant: 6.19 mps
Pres: 1021 hPa
6
Data: 2017-01-16 18:15:54
Temp: -1 °C
Umid: 86 %
Vant: 6.19 mps
Pres: 1021 hPa
7
Data: 2017-01-16 19:33:18
Temp: -1 °C
Umid: 86 %
Vant: 6.69 mps
```

5. Bibliography

- https://code.tutsplus.com/tutorials/create-a-weather-app-on-android--cms-21587
- https://www.simplifiedcoding.net/android-volley-tutorial-to-get-json-from-server/
- https://developer.android.com/training/volley/index.html
- https://www.androidnames.com/development/tutorial-highlight-android-weather-app-tutorial-step-by-step-guide/
- Programarea Aplicatiilor cu Restrictii de Timp Real - curs, prof. dr. ing. Liliana Dobrica.