

**JOWeather Project**

**By**

**Ahid Alsharea**

**Ahedalsharoo@gmail.com**

**ACKNOWLEDGEMENT**

The major goal was to present meteorological data for Jordanian cities.

The following technical requirements were applied to this task:

I utilized the MVC design pattern, Flutter for both Android and iOS, and SQLite as a local database to save weather data.

Every time the weather is needed, the API is used to find it. The weather conditions for 12 governorates are then recorded in the database, and the information is then presented on the application screen.

The code was flawlessly written and adaptable.

The most recent versions of the Flutter libraries were utilized.

**Keyword**: MVC , Flutter , SQLite, API.

## **1.1 Introduction:**

Weather has a variety of effects on our life. The style of clothing we wear, for example, is heavily influenced by the weather, and in many circumstances, such as leisure excursions, we make decisions based on the weather. Our mood is also influenced by the weather.

For thousands of years, people have attempted to predict the weather. For this, scientists now use sophisticated technologies such as radar, satellites, and computers.

Radio and television stations broadcast weather forecasts, and newspapers publish them. Weather forecasts are more accurate than ever before thanks to advanced scientific monitoring devices.

People nowadays, thanks to the rapid advancement of technology, prefer to check the weather on their smartphones, because it is the most convenient and often used way.

## **1.2 Background**

I developed my app using Flutter, a new programming language that can be used on both Android and iOS and has the most up-to-date libraries.

Because this system is self-contained, I've also used the MVC design pattern.

The weather data was then saved in SQLite, a local database, and the weather was eventually presented on the app screen.

**1.3 Problem data**

For all categories utilized in the application, update the weather condition on a regular basis and in an appropriate manner.

**1.4 Objectives**

The major purpose on which I worked was to display the weather in Jordan's 12 governorates at all times of the day in a simple, comfortable, and safe manner for the user.

**1.5 Motives**

The main goal of my project was to show the weather in Jordan's 12 governorates at all times of the day in an easy, pleasant, and secure way for the user.

**1.6 Scope of the Project**

Because flutter is one of the most recent programming languages in use, I used it for this project.

My goals for creating the program were to get accurate and timely information about the weather in each of the 12 governorates.

**1.7 Project Scope**

When I requested the weather condition for one of the governorates after completing the program's application process, I received true and accurate results.

For the 12 governorates, an examination was undertaken, and all of the results were true and correct..

**BACKGROUND AND RELATED WORKS**

**2.1 Introduction**

High or low temperatures, as well as strong rainfall, are becoming more common all throughout the world.

Knowing the weather has become a top priority for many people. And, as we all know, man seeks the most convenient means of facilitating all parts of his life, knowing the weather is an important topic that must be adopted and used.

**2.2 Theoretical background**

Even if he has a smart phone, a person who wants to know the weather turns to the television screen, a newspaper, or any other traditional means, even if he has a smart phone.

Despite its apparent simplicity, this complicates the matter.

As a result, I designed an application to make this process as simple as possible.

**2.3 Related Applications**

2.3.1 AccuWeather

is a fantastic weather app for Android that forecasts the weather with great accuracy. It comes in 33 languages and allows you to modify the weather for your region in a variety of ways. The app can display the temperature in "Fahrenheit or Celsius," as well as humidity, wind, and the sunrise and sunset times. Minute Cast is a function in the app that can predict rain on a minute-by-minute basis. GPS Radar maps, live weather reports, and weather weather are among the other features.

2.3.2Weather Underground uses

It is a unique network that collects weather data and generates a credible weather report by connecting over 180,000 weather stations. A 10-day weather prediction with text summary, severe weather alerts, National Weather Service radio, customisable widgets, and snow and winter reports are also included in the app. An interactive map of all weather stations, as well as changing radar, satellite, and other data, is included in the app.

**SOFTWARE DESIGN AND IMPLEMENTATION**

**I based my project on:**

1- API from [https://openweathermap.org/api .](https://openweathermap.org/api%C2%A0.)

2- Flutter.

3- Local Database (SQLITE).

4- MVC.

5- Unit tests.

**1 API**

**1.1** API call: (GET)

api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}

**1.2** Parameters: city name

**1.3** Response: (JSON)

Example of API response

{

"coord": {

"lon": -122.08,

"lat": 37.39

},

"weather": [

{

"id": 800,

"main": "Clear",

"description": "clear sky",

"icon": "01d"

}

],

"base": "stations",

"main": {

"temp": 282.55,

"feels\_like": 281.86,

"temp\_min": 280.37,

"temp\_max": 284.26,

"pressure": 1023,

"humidity": 100

},

"visibility": 16093,

"wind": {

"speed": 1.5,

"deg": 350

},

"clouds": {

"all": 1

},

"dt": 1560350645,

"sys": {

"type": 1,

"id": 5122,

"message": 0.0139,

"country": "US",

"sunrise": 1560343627,

"sunset": 1560396563

},

"timezone": -25200,

"id": 420006353,

"name": "Mountain View",

"cod": 200

}

**1.4 Accessibility**

**The use of the API is valid on the free offer, which is available for use within 5 days.**

**2 Flutter**

It is a Google-developed open source software development kit for creating user interfaces for Android, iOS, Windows, and Web apps.

Dart is a programming language for creating web, Android, and iOS applications. It is aimed at web and mobile app developers. One of the language's goals is to function with all modern online browsers, mobile devices, and even web servers. The ability to develop a program and publish it on Android and iPhone devices without having to rewrite it in another language is what sets the language apart.

Where the code has been written in a more organized and readable manner, it can be developed and added to in the future, and the most recent libraries were used.

**3 Local Database (SQLITE)**

SQLite is an in-process library that creates a transactional SQL database engine that is self-contained, serverless, and requires no configuration. SQLite's code is in the public domain, which means it can be used for any purpose, commercial or personal. SQLite is the most extensively used database on the planet, with an uncountable number of applications, including some high-profile projects.

SQLite is a SQL database engine for embedded systems. SQLite does not have a separate server process like most other SQL databases. SQLite reads and writes to regular disk files directly. A single disk file contains a whole SQL database, including many tables, indices, triggers, and views. You can freely copy a database across 32-bit and 64-bit platforms, or between big-endian and little-endian architectures, because the database file format is cross-platform. SQLite is a popular Application File Format because of these qualities. The US Library of Congress recommends SQLite database files as a storage format. Consider SQLite as a replacement for open () -source databases rather than Oracle.

SQLite is a small and lightweight library. Depending on the target platform and compiler optimization options, the library size with all features enabled can be less than 600KiB. (A 64-bit code is larger than a 32-bit code.) Additionally, some compiler optimizations, like as aggressive function inlining and loop unrolling, might result in significantly larger object code.) Memory use and speed are mutually exclusive.

SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be faster than direct filesystem I/O.

- The data is saved in the Database in the form of a table:

**JoWeather**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **cityname** | **temperature** | **description** | **Icon** |
| 1 | Irbid | 80.76 | scattered clouds | 03d |
| 2 | Mafraq | 90.91 | clear sky | 01d |
| 3 | Jerash | 84.13 | few clouds | 02d |
| 4 | Ajloun | 82.36 | broken clouds | 04d |
| 5 | Balqa | 84.04 | few clouds | 02d |
| 6 | Amman | 82.49 | scattered clouds | 03d |
| 7 | Zarqa | 84.11 | scattered clouds | 03d |
| 8 | Tafilah | 98.55 | broken clouds | 04d |
| 9 | Karak | 77.72 | clear sky | 01d |
| 10 | Madaba | 80.76 | clear sky | 01d |
| 11 | Maan | 81.77 | overcast clouds | 04d |
| 12 | Aqaba | 91.74 | clear sky | 01d |

Readings were taken on Friday 9/10/2021 am 11:42

- To view the list of icons by number, visit the website:

<https://openweathermap.org/weather-conditions>

**4 MVC**

Model–view–controller (MVC) is a software design pattern that divides related program functionality into three interrelated pieces and is often used for building user interfaces. This is done to distinguish internal information representations from how information is presented to and accepted by users.

This pattern has become popular for creating online apps after being traditionally utilized for desktop graphical user interfaces (GUIs). MVC frameworks for popular programming languages make the pattern easier to implement.

## **Components**

**-Model**

The pattern's most important element. It's the app's dynamic data structure, which isn't affected by the user interface. It is in charge of the application's data, logic, and rules.

**-View**

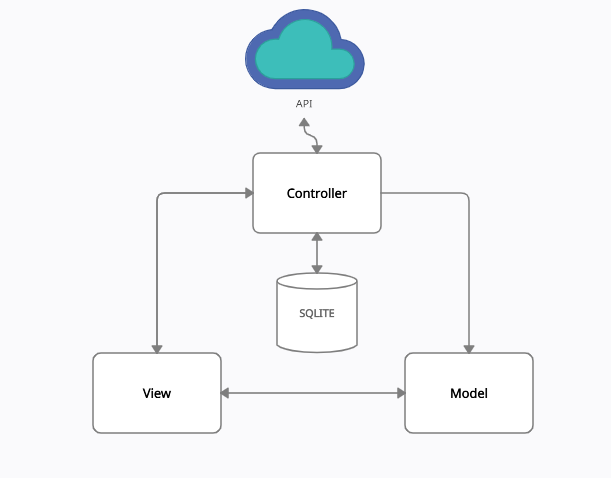
A chart, diagram, or table is a visual representation of data. It is feasible to have many views of the same data, such as a bar chart for management and a tabular view for accountants.

**-Controller**

Accepts input and turns it to model or view commands.

The model–view–controller design not only divides the program into these components, but it also defines the interactions between them.

* The model is responsible for managing the data of the application. It receives user input from the controller.
* The view renders presentation of the model in a particular format.
* The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.
* As with other software patterns, MVC expresses the "core of the solution" to a problem while allowing it to be adapted for each system.Particular MVC designs can vary significantly from the traditional description here.



- Every time, it must load the data from database and get the updated version of it from API.

-Explanation: The controller uses the API to find the requested city, and the response matches the response form and is saved in the JoWeather database. The controller then receives the result from the database and shows it to the user.

**5 Unit tests**

-Unit testing is a software testing method in which tests are written for individual components of a software source, along with any instructions or data that that component requires. The smallest portion that can be tested in the source code is called a unit test.

-A test was conducted between google weather and JoWeather app at the same times and in the same cities, and it gave these results:

JoWeather app google weather

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **City name** | **Temperature** | **Description** | **vs** | **Temperature** | **Description** |
| Irbid | 80.76 | scattered clouds |  | 80.76 | scattered clouds |
| Mafraq | 95.61 | clear sky |  | 95.61 | clear sky |
| Jerash | 87.73 | clear sky |  | 87.73 | clear sky |
| Ajloun | 87.73 | clear sky |  | 87.73 | clear sky |
| Balqa | 87.42 | clear sky |  | 87.42 | clear sky |
| Amman | 87.89 | scattered clouds |  | 87.89 | scattered clouds |
| Zarqa | 89.51 | scattered clouds |  | 89.51 | scattered clouds |
| Tafilah | 99.23 | overcast clouds |  | 99.23 | overcast clouds |
| Karak | 83.12 | clear sky |  | 83.12 | clear sky |
| Madaba | 85.73 | clear sky |  | 85.73 | clear sky |
| Maan | 80.06 | broken clouds |  | 80.06 | broken clouds |
| Aqaba | 93.06 | clear sky |  | 93.06 | clear sky |

Readings were taken on Friday 9/10/2021 pm 06:00