



# Piscine Mobile - 2

## API and Data Management

*Summary: This document contains the subject for Module 02 of the Piscine Mobile.*

*Version: 2.2*

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Specific Instructions</b>	<b>3</b>
<b>III</b>	<b>Introduction</b>	<b>4</b>
<b>IV</b>	<b>Exercise 00: Where are we?</b>	<b>5</b>
<b>V</b>	<b>Exercise 01: Searcher</b>	<b>7</b>
<b>VI</b>	<b>Exercise 02: Fill the Views</b>	<b>9</b>
<b>VII</b>	<b>Exercise 03: What's wrong with you?</b>	<b>11</b>
<b>VIII</b>	<b>Submission and peer-evaluation</b>	<b>13</b>

# Chapter I

## Instructions

- Only this page will serve as reference. Do not trust rumors.
- Read attentively the whole document before beginning.
- Your exercises will be corrected by your piscine colleagues.
- The document can be relied upon, do not blindly trust the demos or pictures example which can contain not required additions.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- By Odin, by Thor ! Use your brain !!!



Intra indicates the date and the hour of closing for your repositories. This date and hour also corresponds to the beginning of the peer-evaluation period for the corresponding piscine day. This peer-evaluation period lasts exactly 24h. After 24h passed, your missing peer grades will be completed with a 0.

# Chapter II

## Specific Instructions


This module is a continuation of the previous one. For clarity, you will copy the project from the previous module into a new repository to continue working on it.

# **Chapter III**

## **Introduction**

# Chapter IV

## Exercise 00: Where are we?

	Exercise :
Where are we?	
Turn-in directory : <code>mobileModule02</code>	
Files to turn in : <i>medium_weather_app</i> and all necessary files	
Forbidden functions : <b>None</b>	

In the previous module, you created a weather app that currently displays a simple text.

Now it's time to add real data to your app.  
To achieve this, you will need:

- A weather API.
- Geolocation using the device's GPS.
- A Geocoding API to retrieve the city name from the coordinates.

Start by implementing geolocation.

When the application starts or when the user clicks a geolocation button, you need to determine the device's location to fetch the weather.

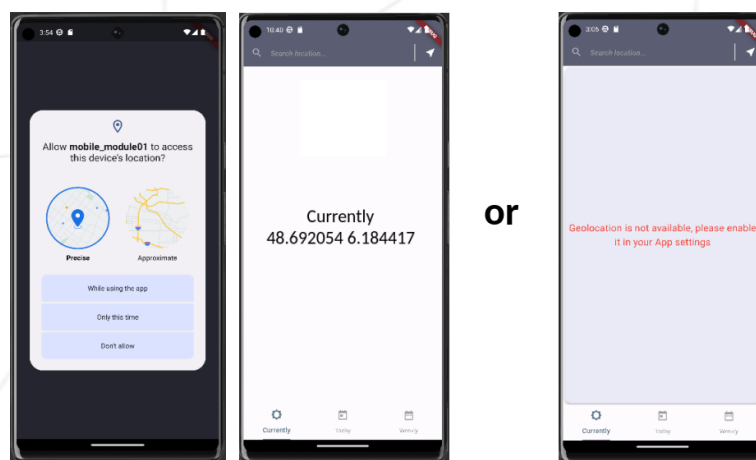
Use the device's GPS to get the coordinates.

This requires obtaining the user's permission to retrieve and use their location.

You must handle both cases: when the user grants permission and when they deny it.

- If the user grants permission, you will retrieve the coordinates and use them to fetch the weather.  
For now, just display the coordinates as text.
- If the user denies permission, your app should still function.  
The user should be able to enter a city name in the search field to get the weather, but they must be informed that you don't have access to their location.


At the start of your application, you should have something like this:



You must not use an external API for geolocation; you are required to use the device's GPS to obtain the coordinates.

# Chapter V

## Exercise 01: Searcher

	Exercise :
	Searcher
	Turn-in directory : <code>mobileModule02</code>
	Files to turn in : <i>medium_weather_app</i> and all necessary files
	Forbidden functions : None

You also have a search bar to get the weather by entering a city name, country, region, etc.

You will use the weather API to get the weather.  
You will also need the Geocoding API to obtain city names from coordinates or coordinates from city names.

When you query the API for weather by city name, you will often receive a list of cities matching the name you entered.

You must display a suggestion list of cities, with each suggestion showing:

- The city name.
- The city's region.
- The country of the city.

This will allow the user to choose the correct city.

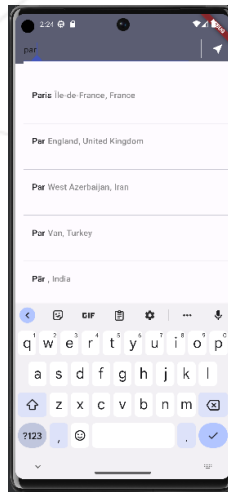
The suggestion list should update dynamically as the user types in the search bar.

When the user selects a city from the list, fetch the weather for that specific city.

The user should also be able to search without selecting a city from the list.



Your app should look something like this:




For this project, we will use:

- [Geocoding API](#) from Open-Meteo to get coordinates from city names and populate the search list.

# Chapter VI

## Exercise 02: Fill the Views

	Exercise :
Fill the Views	
Turn-in directory : <code>mobileModule02</code>	
Files to turn in : <i>medium_weather_app</i> and all necessary files	
Forbidden functions : None	

Now it's time to populate the views with the data.

In your first tab “Current”, you need to display:

- The location (city name, region, and country).
- The current temperature (in Celsius).
- The current weather description (e.g., cloudy, sunny, rainy).
- The current wind speed (in km/h).

In your second tab “Today”, you should display:

- The location (city name, region, and country).
- A list of the day's weather, showing:
  - The time of day.
  - The temperature at each hour.
  - The weather description (cloudy, sunny, rainy, etc.) at each hour.
  - The wind speed (in km/h) at each hour.

In your third tab “Weekly”, display:

- The location (city name, region, and country).

- A list of the weather for each day of the week, including:
  - The date.
  - The minimum and maximum temperatures of the day.
  - The weather description (e.g., cloudy, sunny, rainy).

Be mindful of the following points:

- When you run the application, it should start on the first tab “Current”.
- When you perform a search, you should remain on the tab where you initiated the search.
- When switching tabs, always display the data from the last search.

For now, don’t focus on the design—just ensure that the information is displayed correctly.

You will enhance the design in the next module.

Your app should look something like this:




For this project, we will use the following APIs:

- **Weather Forecast API** from Open-Meteo.

## Chapter VII

### Exercise 03: What's wrong with you?

	Exercise :
What's wrong with you?	
Turn-in directory : <code>mobileModule02</code>	
Files to turn in : <i>medium_weather_app</i> and all necessary files	
Forbidden functions : None	

Great job so far!  
But there's still more to be done.

You've already handled the case where the user denies location access, but there are other situations to account for:

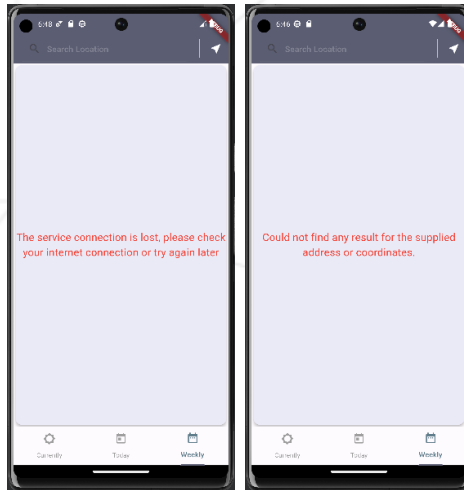
- The user enters a city name that doesn't exist.
- The connection to the API fails.

In both cases, you must inform the user that the city name is invalid or that there was a connection issue.

The message should remain visible until the user enters a valid city name or the connection to the API is restored.

Never underestimate the user's ability to break your application!  
You must always be prepared to handle any edge cases.

Your app should display something like this:



# Chapter VIII

## Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.



The evaluation process will happen on the computer of the evaluated group.