



Piscines: Quick presentation

Summary:

Version: 1.00

Contents

I Piscine

2

II Instruction

3

Chapter I

Piscine

What is a piscine?

Much like the selection piscine, the project is split in many small projects (modules), and with each day a new notion on the language/paradigm/technology you chose to work on. Validating a piscine requires more than just a day or two of investment. To ensure that this piscine is done correctly, it is recommended that you take note of everything listed below:

- You can't validate the piscine if you work only on one or two projects.
- Trying to turn in every day: even if something isn't finished, it's better to try and turn it in.
- At least two week of work: going into the piscine shouldn't be taken lightly.
- Each initialized depot will be automatically closed after one day.
- You have two days to complete the required peer-evaluations in each module.
- It is possible to retry your modules without waiting.



It is important not to do these modules alone. For this reason it is highly recommended to approach these piscines with several students.

Chapter II

Instruction

To validate this piscine you must do the following projects:

- [Mobile - 0 - Basic-of-the-mobile-application](#)
- [Mobile - 1 - Structure and logic](#)
- [Mobile - 2 - API and data](#)
- [Mobile - 3 - Design](#)
- [Mobile - 4 - Auth and DataBase](#)
- [Mobile - 5 - Manage data and display](#)

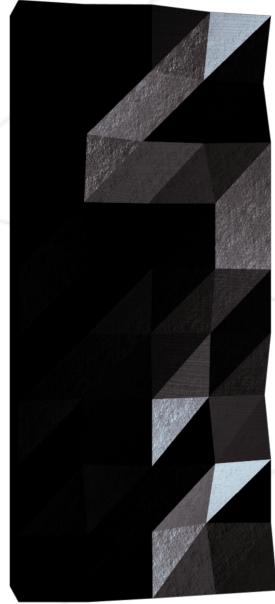
You must validate each project in that order. Indeed, it will not be possible to register for a project if the previous project is not valid.



You must be registered to this project.



It is important to refer to this document to access your projects.
You can click on each module on this page to access it or use the
list of projects available in your intra.



Piscine Mobile - 0

Introduction to Mobile Development

Summary: This document contains the subject matter for Module00 of the Piscine Mobile.

Version: 2.3

Contents

| | | |
|------------|--|-----------|
| I | Instructions | 2 |
| II | Introduction | 3 |
| II.1 | What is a mobile application? | 3 |
| II.2 | What is Flutter? | 3 |
| III | Exercise 00: A basic display | 4 |
| IV | Exercise 01: Say Hello to the World | 6 |
| V | Exercise 02: More Buttons | 8 |
| VI | Exercise 03: It's Alive! | 10 |
| VII | Submission and peer-evaluation | 12 |

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

Introduction

II.1 What is a mobile application?

A mobile application is a software program designed to run on mobile devices, such as personal digital assistants, enterprise digital assistants, or mobile phones.

These applications can either be pre-installed on phones during manufacturing or delivered as web applications using server-side or client-side processing.

Mobile applications often differ from desktop applications, which run on desktop computers, and web applications, which run in mobile web browsers rather than directly on the mobile device.

II.2 What is Flutter?

Flutter is Google's mobile UI framework for crafting high-quality native interfaces on iOS and Android in record time.

Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

Chapter III

Exercise 00: A basic display

| | |
|---|---|
|  | Exercise : |
| | A basic display |
| | Turn-in directory : mobileModule00 |
| | Files to turn in : ex00 and all necessary files |
| | Forbidden functions : None |



As explained in the main project, we are using Flutter for these projects, so we will sometimes use terms specific to this framework. It is up to you to adapt and find the equivalent in the framework you choose to use.

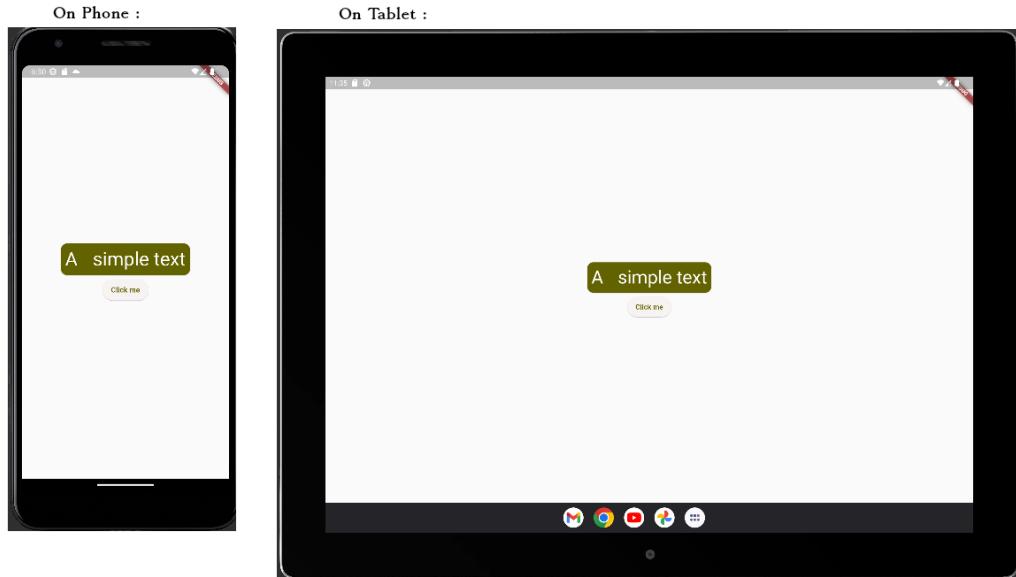
For your first exercise, you will need to create a new ex00 project using the tools provided by the framework of your choice.

If you are using Flutter, it is important to understand the structure of a Flutter project and, for this exercise, what **widgets** are and their different **states**.

For now, your project must contain a single page with some widgets:

- A text widget with a button below it, both centered horizontally and vertically.
- When the button is clicked, you must display “Button pressed” in the debug console.
- Your application must be responsive.

It should look something like this:



On the debug console: **I/flutter (5024): Button pressed**

Chapter IV

Exercise 01: Say Hello to the World

| | |
|--|---|
| | Exercise : |
| | Say Hello to the World |
| | Turn-in directory : mobileModule00 |
| | Files to turn in : ex01 and all necessary files |
| | Forbidden functions : None |

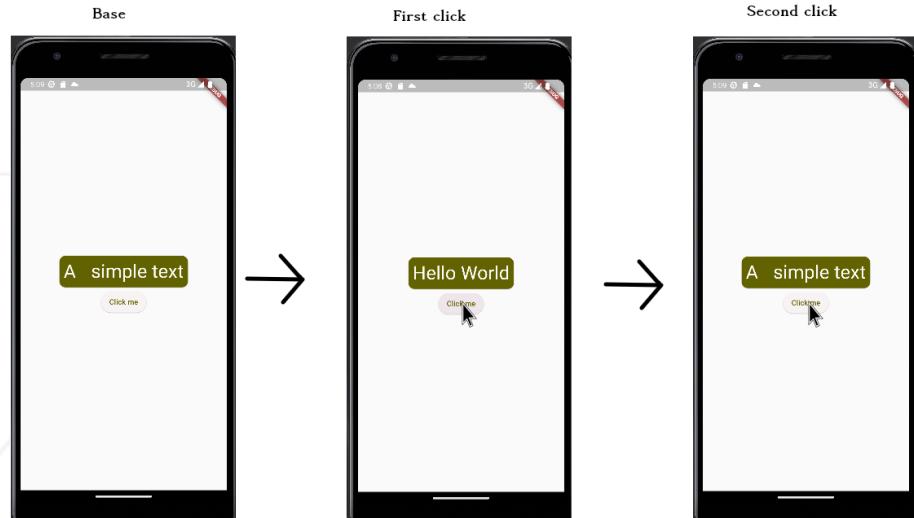


For this exercise, you will need to retrieve the code from your previous exercise and create a new ex01 project.

Now you will need to ensure that the text displayed in the application changes when you click the button.

It should display “Hello World!” instead of the initial text.

Each time you click the button, the text should toggle between the initial text and “Hello World!”.



Chapter V

Exercise 02: More Buttons

| | |
|---|---|
|  | Exercise : |
| | More Buttons |
| | Turn-in directory : mobileModule00 |
| | Files to turn in : ex02 and all necessary files |
| | Forbidden functions : None |

Now that you understand the basics of displaying text and buttons, create a new project called ex02.

In this new project, display an **AppBar** at the top of your screen with the title “Calculator”.

Like this:



You also need to add two **TextFields** (one to display the expression and one to display the result) and several buttons. For now, just display “0” inside both **TextFields**; you will handle this in the next exercise.

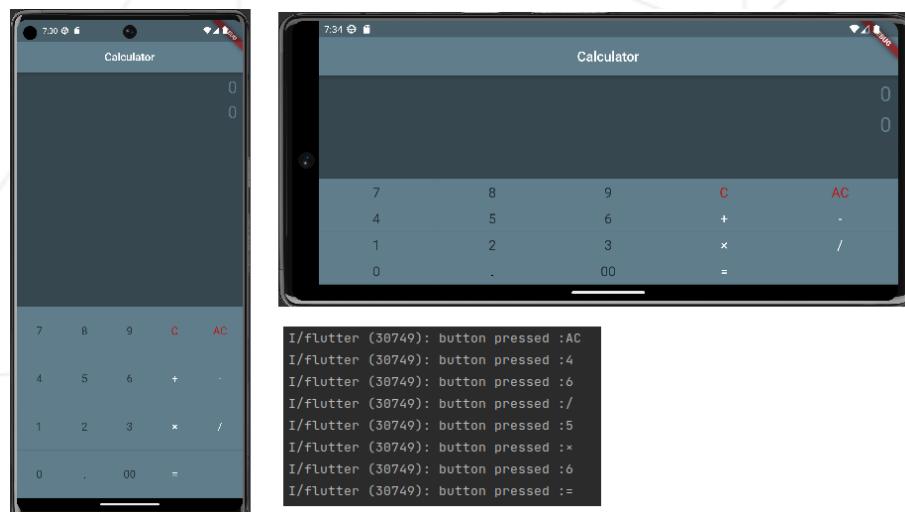
Buttons:

- Numbers from 0 to 9.
- “.” for decimal numbers.
- “AC” to reset the expression and result.
- “C” to delete the last character of the expression.
- “=” to display the result of the expression.
- Operators: “+”, “-”, “*”, “/”.

Add a debug feature. For each button you press, display the text of the button in the debug console.

Once all the buttons are properly set, make sure the display is responsive for all devices (phone, tablet, etc.).

You should have something like this:



Chapter VI

Exercise 03: It's Alive!

| | |
|---|--|
|  | Exercise : |
| | It's Alive! |
| | Turn-in directory : <code>mobileModule00</code> |
| | Files to turn in : <code>calculator_app</code> and all necessary files |
| | Forbidden functions : None |



For this exercise, you will need to retrieve the code from your previous exercise and create the `calculator_app` project. You can use the `math_expressions` library or an equivalent library.

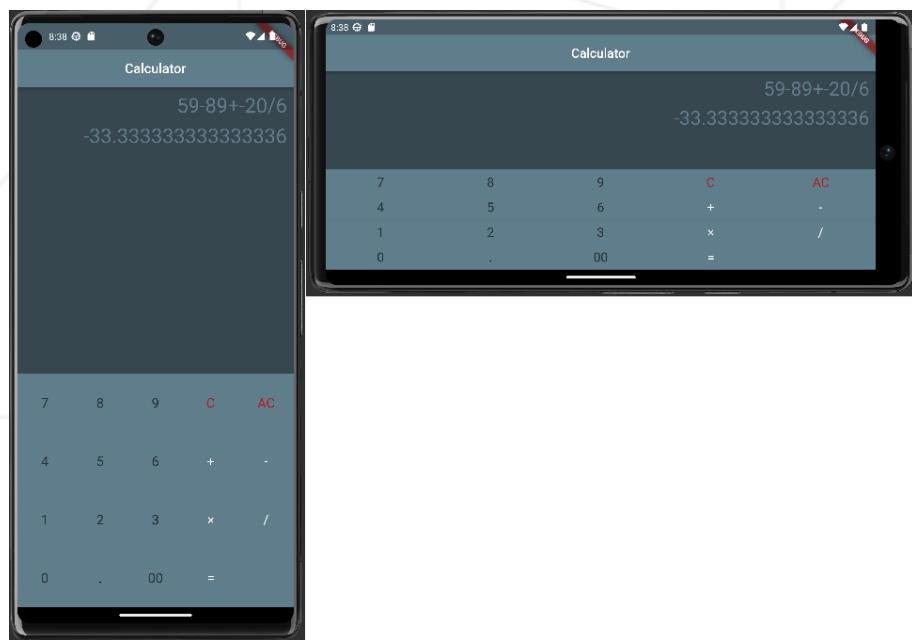
Now it's time to make your calculator work!
You need to add the logic behind it.

The TextFields created in the previous exercise must now display the expression and the result of the expression.

- You must be able to perform the following operations:
 - Addition
 - Subtraction
 - Multiplication
 - Division
- You can perform multiple operations in one expression (e.g., $1 + 2 * 3 - 5 / 2$).
- You must be able to enter a negative number (by pressing the “-” button before the number).
- You must be able to enter decimal numbers.

- You must be able to delete the last character of the expression.
- You must be able to clear the whole expression and result.

Your result could look something like this:



Be careful—if you don't thoroughly test your code, you might encounter issues. For example, an incorrect expression, division by 0, or very large numbers can cause problems.



Your application must NEVER crash!

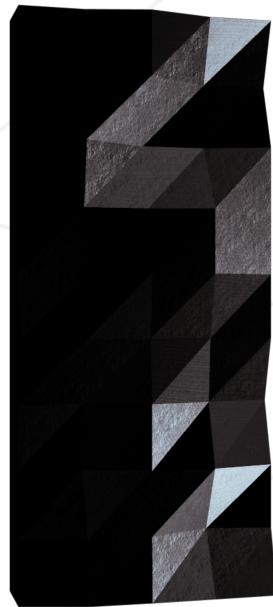
Chapter VII

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.



Piscine Mobile - 1

Structure and Logic

Summary: This document contains the subject matter for Module01 of the Piscine Mobile.

Version: 2.4

Contents

| | | |
|-----|--------------------------------|---|
| I | Instructions | 2 |
| II | Specific Instructions | 3 |
| III | Introduction | 4 |
| IV | Exercise 00: BottomBar | 5 |
| V | Exercise 01: TopBar | 7 |
| VI | Submission and peer-evaluation | 9 |

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

The project in this module will continue in the next module, so it is important to complete it thoroughly.

Chapter III

Introduction

Chapter IV

Exercise 00: BottomBar

| | |
|---|---|
|  | Exercise : |
| | BottomBar |
| | Turn-in directory : <code>mobileModule01</code> |
| | Files to turn in : <code>weather_app</code> and all necessary files |
| | Forbidden functions : None |

In Module01, you will begin preparing your weather application. This application, like all the others you will complete by the end of the Piscine, must be responsive.

To start, create a new project called `weather_app`.

Now, you will create the structure of your application with the following components:

- An AppBar that includes a search TextField and a geolocation button.
- A BottomBar with 3 tabs: “Currently”, “Today” and “Weekly”.
- Each tab should have its own content.



Your application must be responsive.

The BottomBar:

- Create a BottomBar with 3 tabs.
- Each tab should have a name and an icon (Currently, Today, Weekly).
- You should be able to switch between tabs by clicking on them or by swiping; both methods should work.

- When you switch tabs, the content of the page should change. For now, simply display the name of the tab as text, nothing more!
- When the application starts, the first tab (Currently) should be selected by default.



In Flutter:

Use the "TabBar" widget to create a TopBar with tabs.

Use the "TabBarView" widget to create the different views.

Use the "BottomAppBar" widget to create the BottomBar.

Chapter V

Exercise 01: TopBar

| | |
|---|---|
|  | Exercise : |
| | TopBar |
| | Turn-in directory : <code>mobileModule01</code> |
| | Files to turn in : <code>weather_app</code> and all necessary files |
| | Forbidden functions : None |

The TopBar:

- It includes a search TextField.
- It includes a geolocation button.

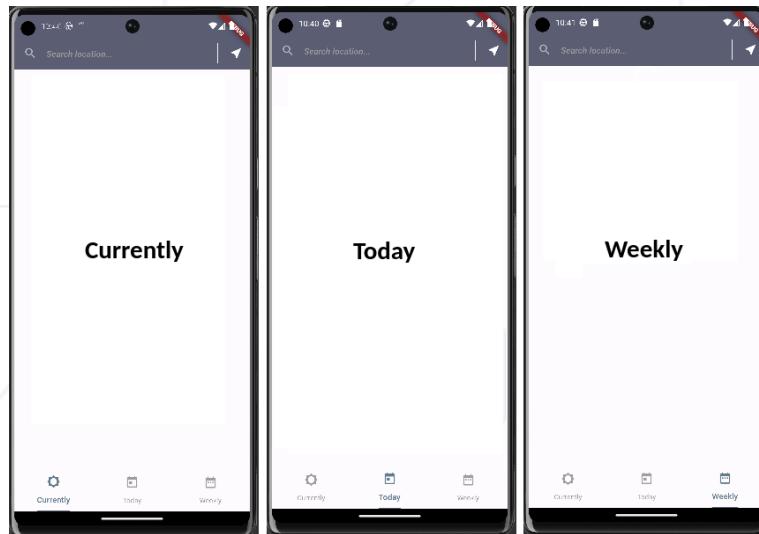
Both the TextField and the geolocation button should work correctly!

The application should display either the text entered in the TextField or the geolocation data, depending on which one is used.

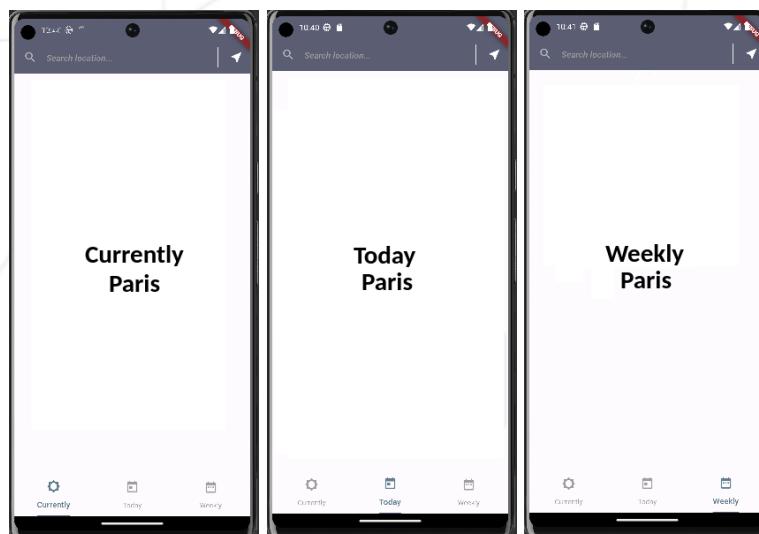
Details:

- If you enter text in the search TextField, the application should display the tab name along with the entered text in all tabs.
- If you click the geolocation button, the application should display the tab name along with the word “Geolocation” in all tabs.

At this stage, your application should look like the following basic layout:



And if you search for a location in the search bar, it should look like this:



Chapter VI

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.



Piscine Mobile - 2

API and Data Management

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

Version: 2.2

Contents

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

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 : <code>medium_weather_app</code> and all necessary files |
| | Forbidden functions : None |

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.

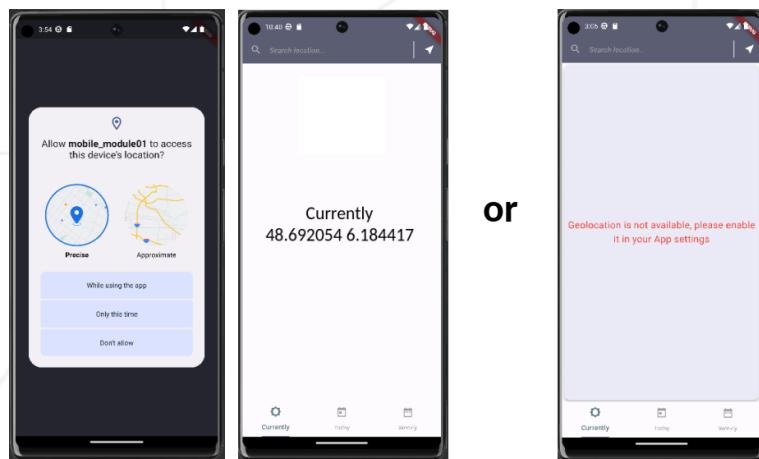
- If the user grants permission, you will retrieve the coordinates and use them to fetch the weather.

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:



or

 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 : <code>medium_weather_app</code> 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.

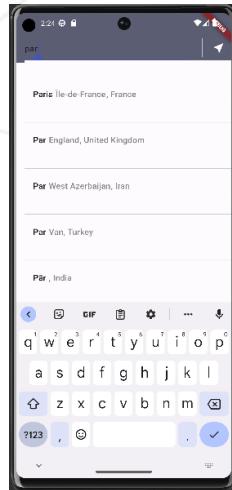
This will allow the user to choose the correct city.

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 : <code>medium_weather_app</code> 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).

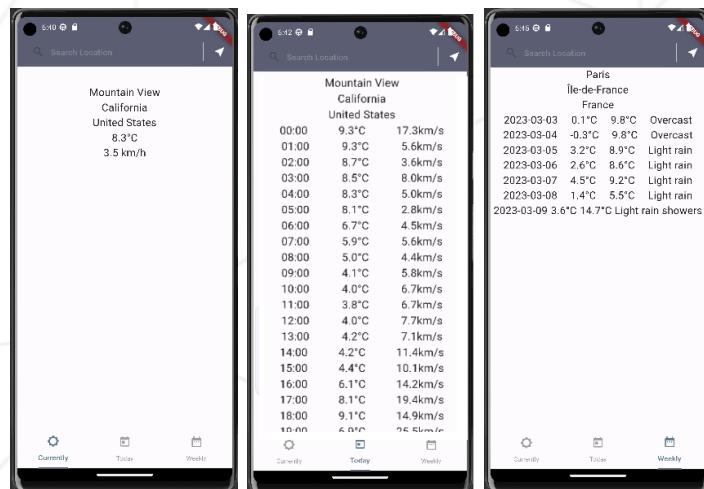
Be mindful of the following points:

- 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 : <code>medium_weather_app</code> 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:

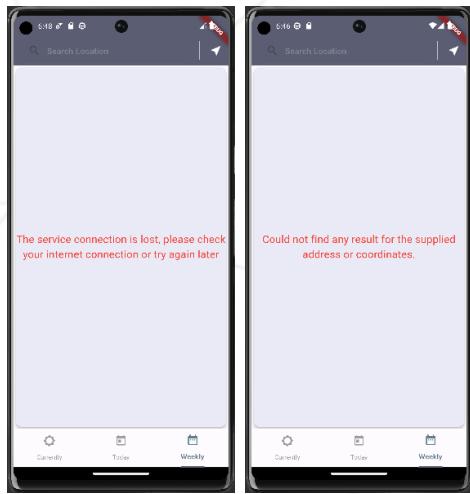
- The user enters a city name that doesn't exist.
- The connection to the API fails.

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.



Piscine Mobile - 3

Design

Summary: This document contains the subject for Module 03 of Piscine Mobile.

Version: 1.2

Contents

| | | |
|------|--------------------------------|----|
| I | Instructions | 2 |
| II | Specific Instructions | 3 |
| III | Introduction | 4 |
| IV | Exercise 00: Search bar | 5 |
| V | Exercise 01: Background | 6 |
| VI | Exercise 02: Current weather | 7 |
| VII | Exercise 03: Today's weather | 9 |
| VIII | Exercise 04: Weekly weather | 11 |
| IX | Submission and peer-evaluation | 13 |

Chapter I

Instructions

- If you have problems installing the tools needed for your project on the 42 computers, you can use a virtual machine. In this case, you will have to :
 - install the virtual machine software on your computer.
 - install the operating system of your choice.
 - install the tools needed for your project.
 - Make sure you have the space on your session to install all of this.
 - You must have everything installed before the evaluation.
- 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

In this module, you will continue developing your project from the previous module. As in the previous module, for clarity and to avoid confusion, you will copy your previous project into a new folder and continue working on it.

Chapter III

Introduction

Congratulations! You have completed the previous module and have a working app. Now it's time to make it more visually appealing and functional!

You are free to choose any design you want, as long as you follow the constraints imposed by the subject.



The choice of a color palette is important for any application. While this will not be graded, it's worth noting that the theme of an application is rarely chosen at random.

Chapter IV

Exercise 00: Search bar

| | |
|--|--|
| | Exercise : |
| | Search bar |
| | Turn-in directory : <code>mobileModule03</code> |
| | Files to turn in : <code>advanced_weather_app</code> and all necessary files |
| | Forbidden functions : None |

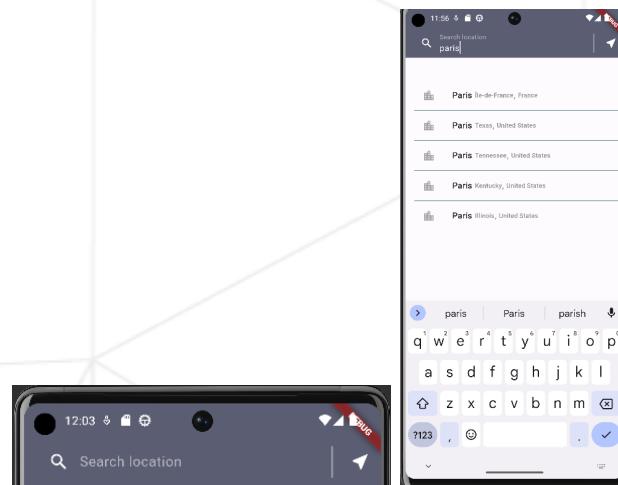
Let's start with the search bar!

The search bar and the list of suggestions should be visually appealing and easy to see.

Ensure that the geolocation button is also clearly visible.

- The user must understand the purpose of the search bar and what they are searching for.
- You should not display more than 5 suggestions at the same time.
- The city name should be clearly visible.

Refer to the following image for an idea of what we are aiming for:



Chapter V

Exercise 01: Background

| | |
|--|--|
| | Exercise : |
| | Background |
| | Turn-in directory : <code>mobileModule03</code> |
| | Files to turn in : <code>advanced_weather_app</code> and all necessary files |
| | Forbidden functions : None |

Choose a background image for your app.

It must follow these rules:

- It must be relevant to your app.
- It should not obstruct the display of information.
- It must cover the entire screen.
- When switching between tabs, the background must remain fixed and should not change.

Your background should not be duplicated—it should be present once at the base of your application and visible throughout.

Chapter VI

Exercise 02: Current weather

| | |
|---|--|
|  | Exercise : |
| | Current weather |
| | Turn-in directory : <code>mobileModule03</code> |
| | Files to turn in : <code>advanced_weather_app</code> and all necessary files |
| | Forbidden functions : None |

Now it's time to display your tabs with the data you have collected.
Start with the current weather tab.

You need to display the following information:

- The location (city name, region, and country)
- The current temperature.
- The current weather description.
- The current weather icon.
- The current wind speed.

All information must be clearly visible and well-placed.
Ensure the user can easily see the information at a glance.

Users should not need more than 3 seconds to understand what the application is displaying.

Refer to the following image for an example of what we are aiming for:



Chapter VII

Exercise 03: Today's weather

| | |
|---|--|
|  | Exercise : |
| | Today's weather |
| | Turn-in directory : <code>mobileModule03</code> |
| | Files to turn in : <code>advanced_weather_app</code> and all necessary files |
| | Forbidden functions : None |

Now, let's focus on the "Today" tab!

In this tab, you will display the weather conditions for the current day.

Using the data you have collected, display the following information:

- The location (city name, region, and country)
- A chart with a temperature curve for the day.
- A list containing the following details:
 - The time of day.
 - The temperature.
 - The weather condition (icon or text).
 - The wind speed.

Specific rules:

- The chart must display the hours and temperature.
- The list must be scrollable.

Refer to the following image for an example of what we are aiming for:



Chapter VIII

Exercise 04: Weekly weather

| | |
|--|--|
| | Exercise : |
| | Weekly weather |
| | Turn-in directory : mobileModule03 |
| | Files to turn in : <i>advanced_weather_app</i> and all necessary files |
| | Forbidden functions : None |

Now, the final tab, “Weekly” tab!

In this tab, you will display the weather conditions for the next 7 days.

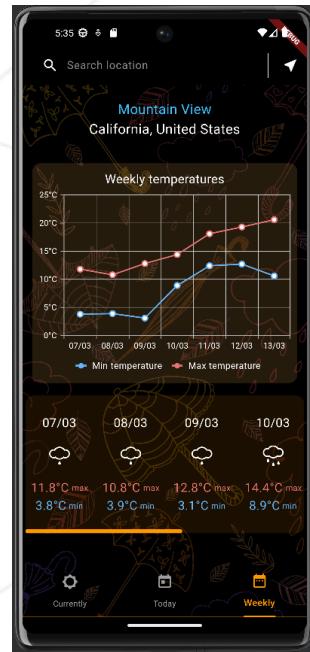
Using the data you have collected, display the following information:

- The location (city name, region, and country)
- A chart with two curves: one for the minimum temperature and one for the maximum temperature for each day.
- A list containing the following information for each day:
 - The day of the week.
 - The minimum temperature.
 - The maximum temperature.
 - The weather condition (icon or text).

Specific rules:

- The chart must display the days of the week along with the min/max temperatures.
- The list must be scrollable.

Refer to the following image for an example of what we are aiming for:



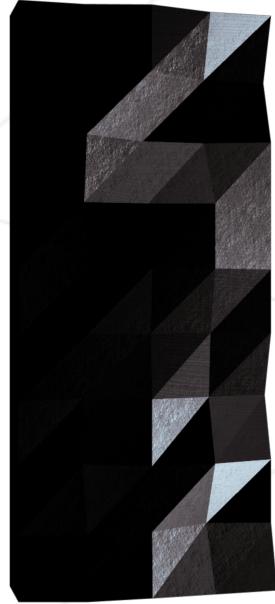
Chapter IX

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.



Piscine Mobile - 4

Auth and Database

Summary: This document outlines the subject for Module 04 of the Mobile Piscine.

Version: 1.2

Contents

| | | |
|-----|---------------------------|---|
| I | Instructions | 2 |
| II | Specific Instructions | 3 |
| III | Exercise 00: Login Page | 4 |
| IV | Exercise 01: Profile Page | 6 |

Chapter I

Instructions

- If you have problems installing the tools needed for your project on the 42 computers, you can use a virtual machine. In this case, you will have to :
 - install the virtual machine software on your computer.
 - install the operating system of your choice.
 - install the tools needed for your project.
 - Make sure you have the space on your session to install all of this.
 - You must have everything installed before the evaluation.
- 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

To begin this module, you will create a new project app named “diary_app” in a new repository called “mobileModule04”.

This project will continue into the next module.

Your new application will be a diary app.

It will allow you to create, read, and delete diary entries.

The diary will be protected by an authentication system.

All entries will be stored in a database.

In this module, your task is to set up the foundation of your application with an authentication system and database.

Chapter III

Exercise 00: Login Page

| | |
|---|---|
|  | Exercise : |
| | Login Page |
| | Turn-in directory : <code>mobileModule04</code> |
| | Files to turn in : <code>diary_app</code> and all necessary files |
| | Forbidden functions : None |

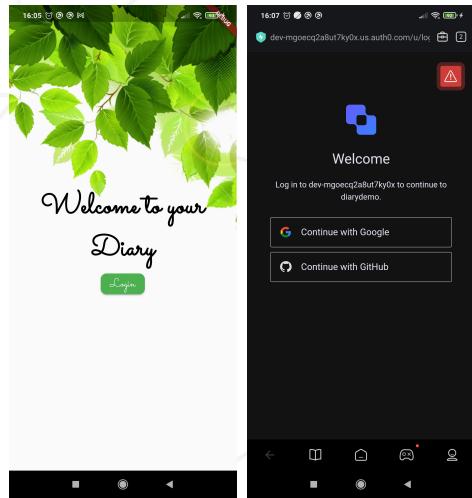
Start by creating the login page.

This page must include:

- A login button that either redirects to an authentication page or directly to the diary page if the user is already logged in.
- The option for the user to log in using a Google or GitHub account.

You must choose an authentication system where users can be stored and managed.
You may use an authentication system like Firebase, AWS, etc.

For your diary app, the login page should look something like this:



Chapter IV

Exercise 01: Profile Page

| | |
|---|---|
|  | Exercise : |
| | Profile Page |
| | Turn-in directory : <code>mobileModule04</code> |
| | Files to turn in : <code>diary_app</code> and all necessary files |
| | Forbidden functions : None |

The next step is to create the profile page.

This page should only be accessible if the user is logged in.

The authentication system should redirect the user to this page after logging in, or directly if they are already logged in.

Before creating the page, you will need to set up a database to store all diary entries.

In the database, you need to store:

- The user's email address.
- The date of each entry.
- The title of each entry.
- The user's feeling of the day.
- The content of the entry.

The database structure should resemble this:

The screenshot shows the Google Cloud Firestore interface. On the left, there's a sidebar with a home icon, followed by 'notes' and a document icon. Below that, there's a button '+ Commencer une collection' and a section for 'notes'. To the right, there's a list of documents under the heading 'notes'. One document is expanded, showing its details. At the top right, there's a blue cloud icon with the text 'Plus de fonctionnalités dans Google Cloud' and a dropdown menu.

| notes | PfrTSJfPJQD1IvYW5ji |
|-------|---|
| | i0g540lpkNJBrlEB9vXC q5BS3Ri3FUbxch1mQ4m rqCAGbHt6HajRUdmmxyU yQHan5Dh6TigTnDDkDbB |

| | PfrTSJfPJQD1IvYW5ji |
|----------------------------|---------------------------------------|
| + Commencer une collection | |
| + Ajouter un champ | |
| date: | 16 mars 2023 à 20:20:04 UTC+1 |
| icon: | "satisfied" |
| text: | "hfyxhdufnfi fjjfbfufbdux jjffudbxjb" |
| title: | "test2" |
| usermail: | [REDACTED] |

You must be able to create, read, and delete entries, so you need to implement the logic for these operations.

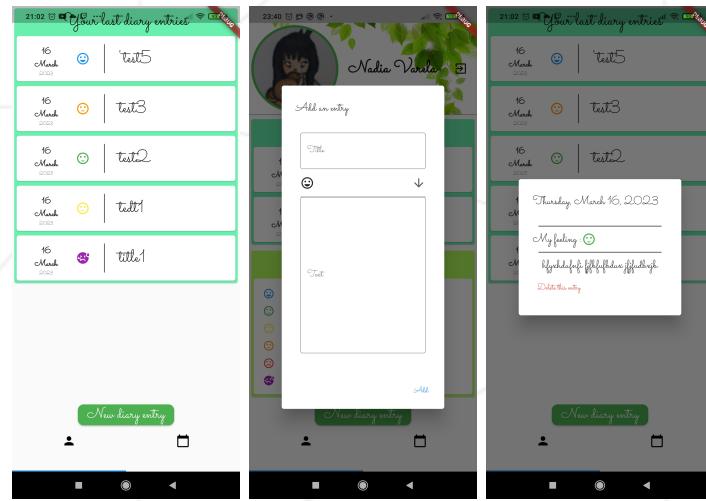
For now, create a profile page that includes:

- A list of all diary entries.
- A button to create a new entry.
- When a user taps on an entry, they should be able to read it.
- A button to delete an entry.

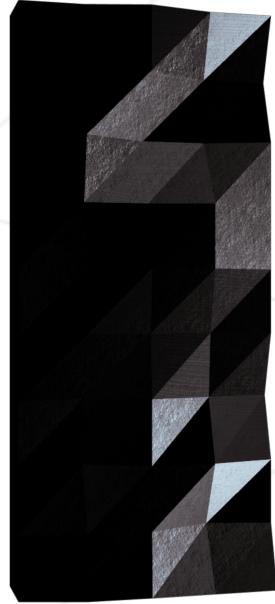
When a new entry is created, the list must be updated.

When an entry is deleted, the list must also be updated.

The profile page should look something like this:



For the evaluation, you must create a Google account for your evaluator to test your app. Ensure that this account contains some diary entries.



Piscine Mobile - 5

Manage Data and Display

Summary: This document outlines the subject for Module 05 of the Mobile Piscine.

Version: 1.00

Contents

| | | |
|-----|---------------------------|---|
| I | Instructions | 2 |
| II | Specific Instructions | 3 |
| III | Exercise 00: Profile Page | 4 |
| IV | Exercise 01: Agenda Page | 6 |

Chapter I

Instructions

- If you have problems installing the tools needed for your project on the 42 computers, you can use a virtual machine. In this case, you will have to :
 - install the virtual machine software on your computer.
 - install the operating system of your choice.
 - install the tools needed for your project.
 - Make sure you have the space on your session to install all of this.
 - You must have everything installed before the evaluation.
- 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 project is a continuation of Module 04. You will need to implement a new feature in the application you have already developed. For clarity and to avoid confusion, copy your previous project into a new folder and continue working from there.

Your application must now have 3 pages:

- The first page is the login page, which should contain the buttons for login.
- The second page is the profile page.
- The third page is the agenda page.

The app requires an internet connection to work.

Chapter III

Exercise 00: Profile Page

| | |
|---|--|
|  | Exercise : |
| | Profile Page |
| | Turn-in directory : <code>mobileModule05</code> |
| | Files to turn in : <code>advanced_diary_app</code> and all necessary files |
| | Forbidden functions : None |

Now that you have a diary application that allows adding and deleting entries, the next step is to add more features to the profile page.

The profile page must display at least the following information:

- The user's name.
- A logout button that logs out the user and redirects them to the login page.
- A list showing the last 2 entries added by the user, including the date, feeling, and title of each entry.
- The ability to select an entry to view its details and delete the entry.
- The total number of entries.
- A list showing the feelings and their percentage of use across all entries.
- A button to add a new entry.

All information must be updated in real time when an entry is added or deleted.

Your profile page should look something like this:



Chapter IV

Exercise 01: Agenda Page

| | |
|---|--|
|  | Exercise : |
| | Agenda Page |
| | Turn-in directory : <code>mobileModule05</code> |
| | Files to turn in : <code>advanced_diary_app</code> and all necessary files |
| | Forbidden functions : None |

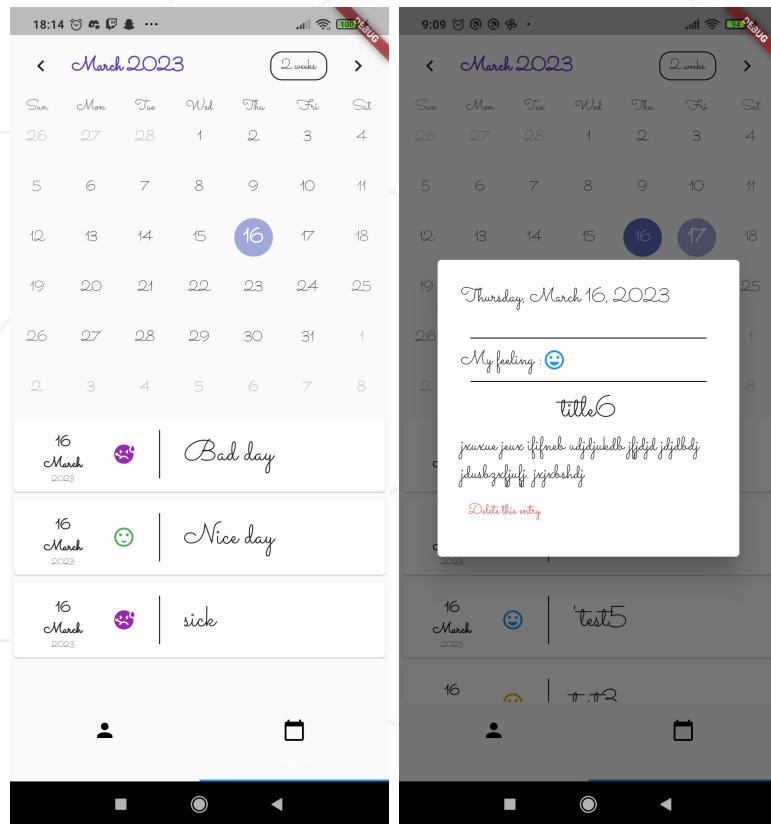
Now, let's implement the agenda page!

This page must display a calendar.

Requirements:

- When you open this page, the calendar should display the current date.
- You can select a date from the calendar.
- When you select a date, a scrollable list of entries from that date should appear.
- You should be able to select an entry to view its details.
- If you delete an entry, the list must be updated accordingly.

Your agenda page should look something like this:



For this project, we used the "table_calendar" Flutter dependency.
[pub.dev](https://pub.dev/packages/table_calendar)