Mobile Computing

Practical Assignment #2

Weather Forecast

My preferred cities weather

FEUP, M.EIC, 1st year, May 2022

Group #1

Diana Freitas	up201806230
José Rodrigues	up201708806
Juliane Marubayashi	up201800175

Index

1. Introduction **3**

2. Architecture **3**

2.1. Code Structure 3

2.2. Database 3

2.3. Main Page 4

2.4. City Page 4

2.5. Manage Locations 5

3. Interface **6**

3.1 Application pages 6

3.2 Application flow 7

4. Features **8**

4.1 Default City 8

4.2 Weather Icons and Images 8

4.3 Extra information 8

5. Testing **9**

6. Use Cases **9**

7. Conclusion **9**

8. Bibliography **10**

1. Introduction

In the context of the second assignment of Mobile Computing, this project aims to provide an application for weather data consultation.

In this mobile application, the user is able to manage a record of cities he is interested in, among the district capitals of Portugal, for which he can obtain information about the weather. The user can consult the current weather conditions in each of these cities and he is also able to define a favorite location.

Furthermore, the user can consult the weather forecasts for the next 5 days for each of the cities in his list in increments of 3 hours, obtaining information about the temperature, pressure, precipitation, wind, and humidity.

In the following sections, this application will be explained in further details regarding its interface, architecture, main features, and other relevant topics.

2. Architecture

2.1. Code Structure

To better organize the code of the application, the following packages were used:

- ***models***: Includes the *City* class, which represents the main entity of this project;
- ***components***: To improve the clarity of the code, this package includes *Stateless Widgets*, such as cards, that are used repeatedly throughout the code or that were extracted into a single component to improve code readability;
- ***utils***: Includes multiple auxiliary functions used mainly to generate *FutureBuilders* and to map the weather data, received from the API, to the respective element of the *WeatherStatus* enum and to the icon and image that represent that weather.
- ***databases:*** Includes the Singleton SQLite Database used to store the cities and user preferences;
- ***httpRequests:*** Includes auxiliary functions used to fetch the weather from the API.

Furthermore, for each of the main pages of the application (Main Page, City Page and Manage Locations Page) we decided to create a package that stores the main widgets associated with it.

2.2. Database

In order to persist the available locations and the preferences of the user, the package *sqflite* was used to create a Singleton SQLite database. The database consists of a single table **City** which stores each city of the district capitals of Portugal. Besides the name of the city, each record contains an **isOfInterest** field, which indicates if the city is one of the preferred cities of the User, and an **isFavorite** field, which indicates if the city is the default city of the user (the one that is highlighted in the main page of the Application).

2.3. Main Page

The main page starts by instantiating the databases in its state (*_MyHomePageState*) and, after that, it fetches the cities from the database and stores them in a list (*cities*), by querying the database.

The page is responsible for displaying the temperature of the user's favorite city, also allowing him to visualize the temperature in the other locations of his list and to change his favorite location. Let's discuss the three main methods that handle the three sections described earlier.

- ***optionsButton*: This function firstly renders a button (+) on the superior right corner of the screen. By pressing this button the user can visualize, select and unselect the cities of interest displayed in the main page.
- ***mainTemperature***: This is a *Column Widget* that occupies half of the screen and displays the current temperature information of the user's favorite city, where the default city is Lisbon. To fetch the information of a city, HTTP requests are made to the open weather map API.
- ***LocationsList***: This *Widget* occupies the other half of the screen and it displays a list of cards containing the information about the cities of interest selected by the user, as well as the current temperature of each one. By clicking in a card, the user is redirected to the City Page. If no cities are selected, this function is not executed and the *mainTemperature* function renders a *Widget* that occupies an entire screen.

2.4. City Page

On the individual page of each city, a *StatefulWidget* (*CityPage*) is used, receiving as arguments the country and the city that will be used to build the page.

In the construction method, the function that will make the API call and get all the necessary weather data is called first (*get5dayForecast()*), obtaining the data and passing it down to the functions that will build the three sections of the page:

- ***getLocationMain***: This section covers the first half of the screen and displays the current temperature, the current weather conditions, and also displays the city and country in question;
- ***getForecast***: This is a scrollable section, which in a blocks-in-a-bar style displays the weather conditions (in icon form), the temperature, the time of day, and the weekday it represents, in 3-hour increments, 5 days in the future. These forecast blocks can be selected, and the conditions displayed in the *getConditions* section will change accordingly;

- ***getConditions***: This section displays the 6 main components of the weather conditions in the selected weekday and time, showing in a 2-row 3-column table the following metrics:
    - Humidity (%);
    - Feels like (ºC);
    - Precipitation (%);
    - Cloudiness (%);
    - Pressure (hPa);
    - Wind speed (m/s).

All of these pages use a *FutureBuilder* to construct their elements, through the function *getStringFutureBuilder()*.

This page also has an in-built tutorial feature, to teach the users about the scrolling ability of the forecast section, when this page is first accessed on the device it will animate the scrolling of the section all the way through and back to showcase this feature. This only happens once per device.

2.5. Manage Locations

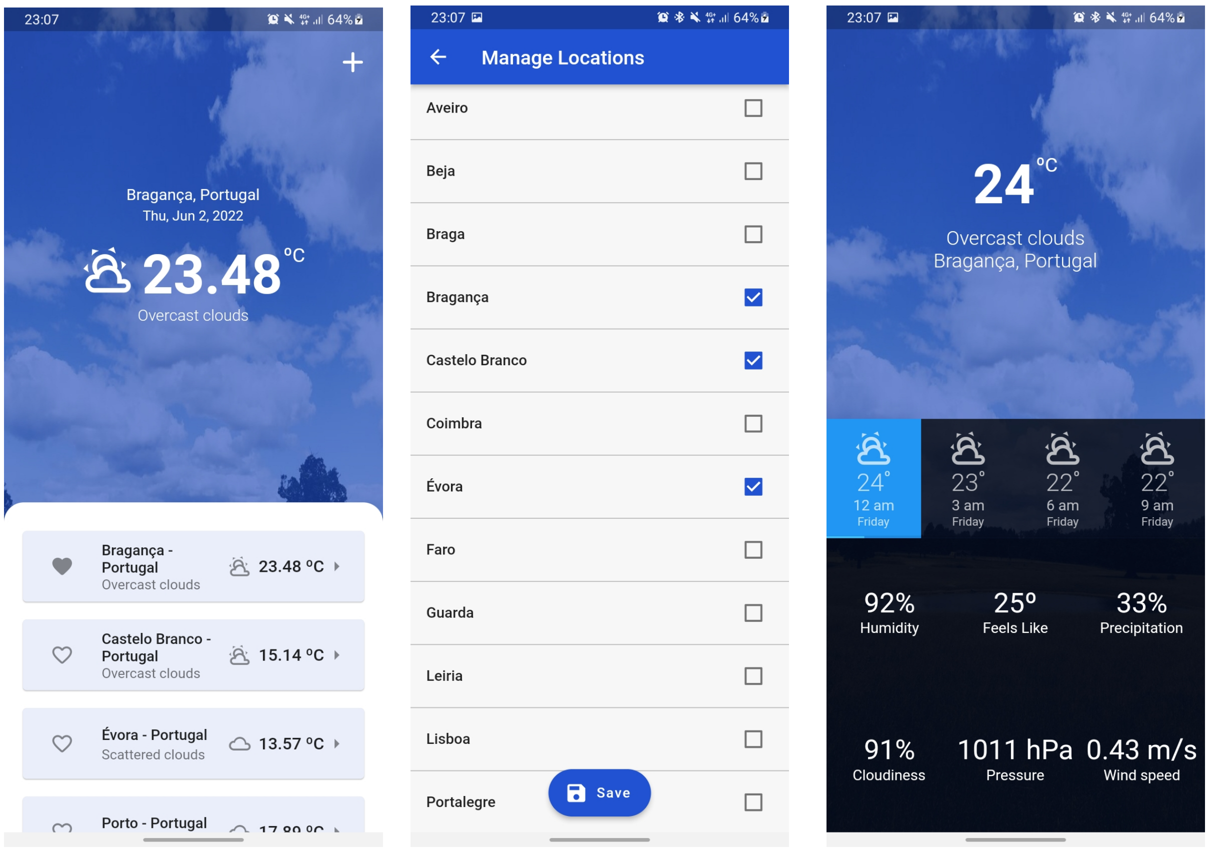
To allow the user to manage his list of preferred cities, a *StatefulWidget* (*ManageLocationsPage*) was used. To init the state of this page, the list of available cities is first fetched from the database. After the cities are loaded, the page will show a checklist where the user is able to check a city to add it to his list or uncheck it to remove it. The cities list is updated every time a checkbox is selected or unselected. To save the updated list the user must press the “Save” button, which will result in a call to the *saveCities* function, which updates the database with the new preferences.

To implement the checklist described above, the following functions were used:

- ***citiesListView:*** Returns a *ListView* in which each item is a *CityCheckBoxTile,* a component that uses a *CheckboxListTile* to display a city of the list;
- ***saveButton:*** Returns a *FloatingActionButton* that, when pressed, calls the *saveCities* function;

3. Interface

3.1 Application pages



Bragança, Portugal
Thu, Jun 2, 2022
23.48 °C
Overcast clouds

Bragança - Portugal	Overcast clouds	23.48 ºC
Castelo Branco - Portugal	Overcast clouds	15.14 ºC
Évora - Portugal	Scattered clouds	13.57 ºC
Porto - Portugal		17.89 ºC

Aveiro	☐
Beja	☐
Braga	☐
Bragança	☑
Castelo Branco	☑
Coimbra	☐
Évora	☑
Faro	☐
Guarda	☐
Leiria	☐
Lisboa	☐
Portalegre	☐

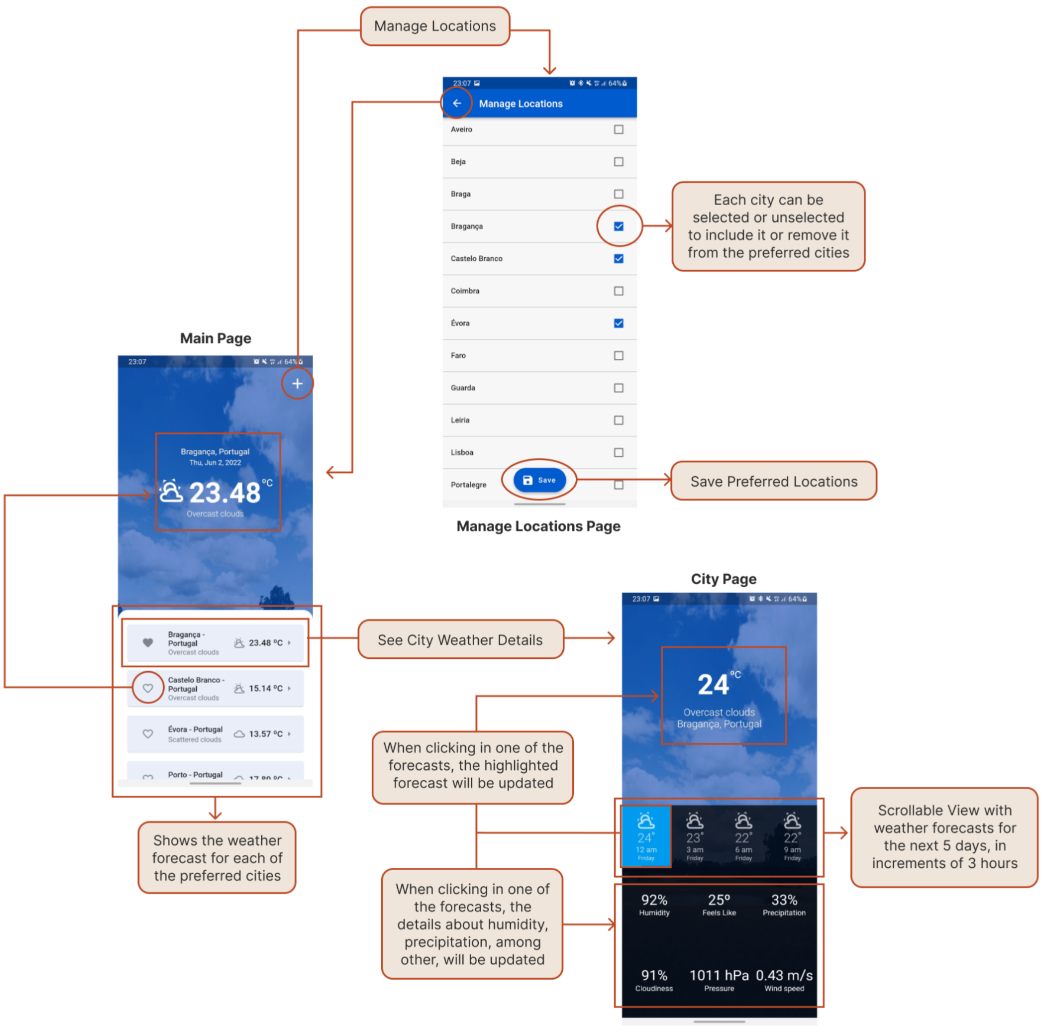
24 °C
Overcast clouds
Bragança, Portugal

24°	23°	22°	22°
12 am	3 am	6 am	9 am
Friday	Friday	Friday	Friday

92%	25º	33%
Humidity	Feels Like	Precipitation
91%	1011 hPa	0.43 m/s
Cloudiness	Pressure	Wind speed

Figure 1: Application pages - from left to right, the screens show the Main Page, the Manage Locations Page and the City Page;

3.2 Application flow



Manage Locations
Manage Locations
Aveiro
Beja
Braga
Bragança
Castelo Branco
Coimbra
Évora
Faro
Guarda
Leiria
Lisboa
Portalegre
Save
Each city can be selected or unselected to include it or remove it from the preferred cities
Save Preferred Locations
Manage Locations Page
Main Page
Bragança, Portugal
Thu, Jun 2, 2022
23.48 ºC
Overcast clouds
Bragança - Portugal 23.48 ºC
Castelo Branco - Portugal 15.14 ºC
Évora - Portugal 13.57 ºC
Porto - Portugal
See City Weather Details
Shows the weather forecast for each of the preferred cities
City Page
24 ºC
Overcast clouds
Bragança, Portugal
When clicking in one of the forecasts, the highlighted forecast will be updated
When clicking in one of the forecasts, the details about humidity, precipitation, among other, will be updated
Scrollable View with weather forecasts for the next 5 days, in increments of 3 hours
24° 12 am Friday
23° 3 am Friday
22° 6 am Friday
22° 9 am Friday
92% Humidity
25º Feels Like
33% Precipitation
91% Cloudiness
1011 hPa Pressure
0.43 m/s Wind speed

Figure 2: Application Flow (Navigational Map)

In this subsection it is explained how a user can navigate through the application.

1. **Main Page:** From the Main Page it is possible to navigate to the Cities List and to the City Page. To visualize and manage the preferred cities, the user shall click on the "+" button in the superior right corner of the page. To visualize a certain city page, the user must click on the respective city card available at the bottom of the screen
2. **Manage Locations Page:** From this section, the user has only the option to navigate back to the main page, by clicking on the arrow button in the superior left corner of the screen.
3. **City Page:** As well as in the Manage Locations Page, from this page the user is only allowed to navigate back to the Main Page.

4. Features

4.1 Default City

As discussed previously, the weather application includes a Main Page and a page that describes the meteorology of a city. However, a user doesn't need to access the City Page to check the weather condition of the current day: he can choose a default city, whose temperature will be displayed in the main panel, by clicking on the heart icon which is displayed in the card of the respective city.

4.2 Weather Icons and Images

In order to display personalized weather icons and images, the iconId returned by the API response, is mapped to each type of weather represented by the WeatherStatus enumeration:

- SUNNY;
- PARTLY_CLOUDS;
- CLOUDY;
- RAIN;
- THUNDERSTORM;
- SNOW;
- MIST;
- PARTLY_RAINING.

The main functions that handle the icons and images are: *getTemperatureIcon(WeatherStatus weatherStatus)* and *getWeatherImage(String iconId).* The first function is responsible for returning the icon corresponding to the given *WeatherStatus* and the second one returns the path to the image that corresponds to the given *iconId.*

4.3 Extra information

On the city page, extra weather information about the weather conditions is displayed, particularly the:

- Humidity. The percentage of water vapor in the air. (%);
- Feels like. This parameter accounts for the human perception of weather. (ºC);
- Precipitation. Probability of precipitation. (%);
- Cloudiness. Percentage of cloud coverage. (%);
- Pressure. Atmospheric pressure on the sea level, in hectopascal. (hPa);
- Wind speed. Speed of the wind in meters per second. (m/s).

5. Testing

To guarantee that the system worked as expected, each feature was continuously tested, making sure that new functionalities did not break the ones that were already there from previous iterations.

In specific, some of the edge scenarios we tested were the following:

- We tried to continuously take all the cities from the list and put them back, and it worked as expected. When no cities were selected, Lisbon was displayed as the default one;
- We tried to unfavorite the current favorite city when it was the only city present, and as expected, the action didn’t do anything, as when only one city is present it is automatically the favorite one;
- We attempted to turn on flight mode (disable internet connection) and access the application. As expected, the app was unable to give us any information as it had no access to the API, but as we intended, it showed an user-friendly message indicating the cause of the error.

6. Use Cases

In our application there is just one actor: the user. To better describe the expected use cases of the application, the following User Stories were defined

1. Visualize the list of cities in Portugal;
2. Visualize the weather conditions of a city of interest: temperature, pressure, precipitation, wind, humidity, temperature feeling and cloudiness;
3. Visualize the temperatures for the next 5 days;
4. Manage the cities of interest;
5. Select the favorite city to be displayed in the main page;
6. Change the favorite city.

7. Conclusion

In conclusion, we believe that the goals of the project were successfully achieved. Furthermore, some extra features were also implemented and the final product intuitively fulfills every task it was meant to. It also allowed us to consolidate the knowledge about all the involved flutter mechanics that were required to complete the project, ending up with an App that is both functional and aesthetically pleasant.

Finally, some enhancements could be done to further improve our application, such as increasing the database of cities by including cities of other countries or increasing the amount of extra weather metrics displayed in the city page to allow for a better understanding of the current weather conditions.

8. Bibliography

- **Project instructions**
    - https://moodle.up.pt/pluginfile.php/220135/mod_resource/content/2/Assgn2_%202022.pdf

- **Weather API**
    - https://openweathermap.org/current
    - https://openweathermap.org/forecast5

- **Flutter Documentation**
    - https://docs.flutter.dev/

- **SQLite (*sqflite* package)**
    - https://docs.flutter.dev/cookbook/persistence/sqlite
    - https://pub.dev/packages/sqflite

- **HTTP Requests (*http* package)**
    - https://docs.flutter.dev/cookbook/networking/fetch-data
    - https://pub.dev/packages/http