

(Just another)

Weather Forecast

Author: U. Schiara

Date: 23/Jan/2023

Rev.1

TABLE OF CONTENTS

1	Foreword	3
2	Scope	3
3	The main Screen	4
4	Language Selection	5
5	Location settings display	6
6	The Files	8
6.1	Config.txt file	8
6.2	Locations.txt file	9
7	Credits.....	9
8	Conclusion	9
	Figure 1 Main Screen	4
	Figure 2 Language Selection Popup.....	5
	Figure 3 Location Settings Menu	6
	Figure 4 Cities list picker	7
	Figure 5 Location Use/Erase/Store.....	8

1 Foreword

If you look to the web, you can find that the weather forecast applications are one more than one thousand! The problem is that very often they are full of Advertisements and this is really annoying me. Therefore, I've decided to develop one on my own, completely free from ADV.

I've then started to search a web site where I could find a free weather forecast: I successfully found Open-Meteo.com, which offers a free forecast, once asked to their web site, passing to it Latitude and Longitude of the wanted location.

Having set some parameters, like City Name, type of forecast wanted, days of week, hours and some others, the site creates a query that can be used by the AI2 "WEB1 Get" instruction block.

The response is contained in a JSON file, that is unpacked by finding various keywords and retrieving the related values (i.e., temperature, wind speed, status of weather, etc.). The unpacking does not use the dictionary methods, but with a mechanism of interrelated lists.

Since the forecast query needs the Latitude and Longitude values to retrieve the "local" forecast, another web site, where to find the location coordinates of a city has been searched. Despite many sites are offering this service in a paid way, only a few of them offers a free service without creating an account. I found then Geocode.xyz, which offers a completely free geolocalization service. The only constraint is the limited number of attempts per second: if the providing server is busy in the moment in which you ask for a new city, a popup asking the user for a new attempt is raised (a Notifier as warning block). Also this web site answers are provided as a JSON file: in this case the "Dictionary blocks" and the "Decode JSON" blocks have been used.

2 Scope

The aim of this document is to describe the structure of the app and to give some indications how to use it.



This is the icon: by tapping on it, the app starts.

3 The main Screen

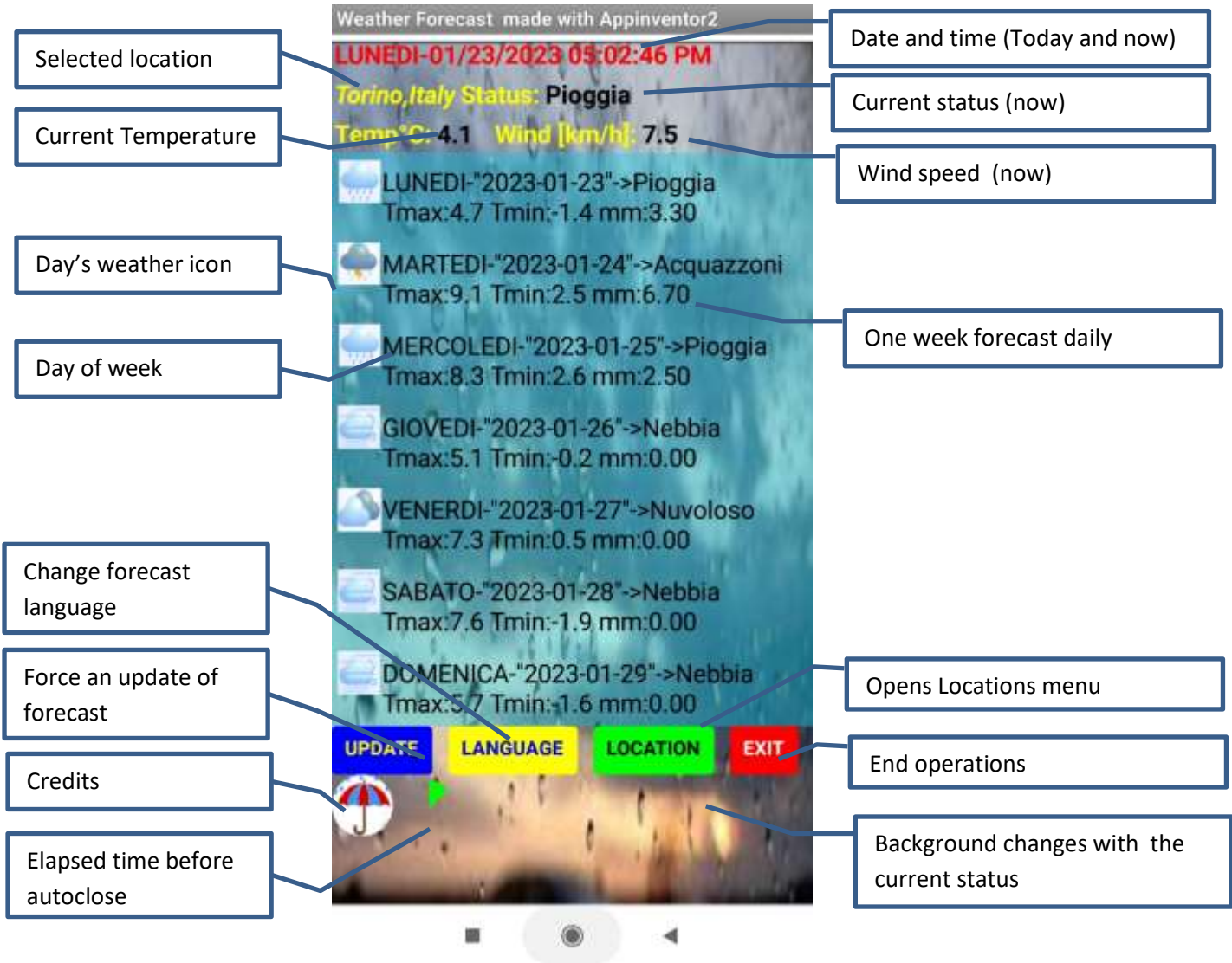


Figure 1 Main Screen

In the Main screen the current and the weekly forecast are shown. The language can be selected between English and Italian (my mothertongue). Once selected the language, the forecast messages change and the new language is stored into a configuration file, so at the next app opening, the language is maintained. The background changes according to the current status, while for the weekly reports, small icons represent the “average” weather status for each day. Minimum and maximum temperatures are shown, together with the precipitation millimeters for that day.

A green pie slice increases every second to warn the elapsing of time, when the pie reaches its end (the pie becomes full round, and changes to red) the app auto-closes. By tapping on every button, a new time slot (30 seconds) restarts.

By tapping on the umbrella, some credits appear for a while.

4 Language Selection



Once tapped on the yellow button the language selection popup appears.

English or Italian are currently available.

If none is selected, the current language remains.

If a language is selected, the forecast messages change, immediately and the new language is stored into the configuration file. See chapter 6.1

Figure 2 Language Selection Popup

5 Location settings display

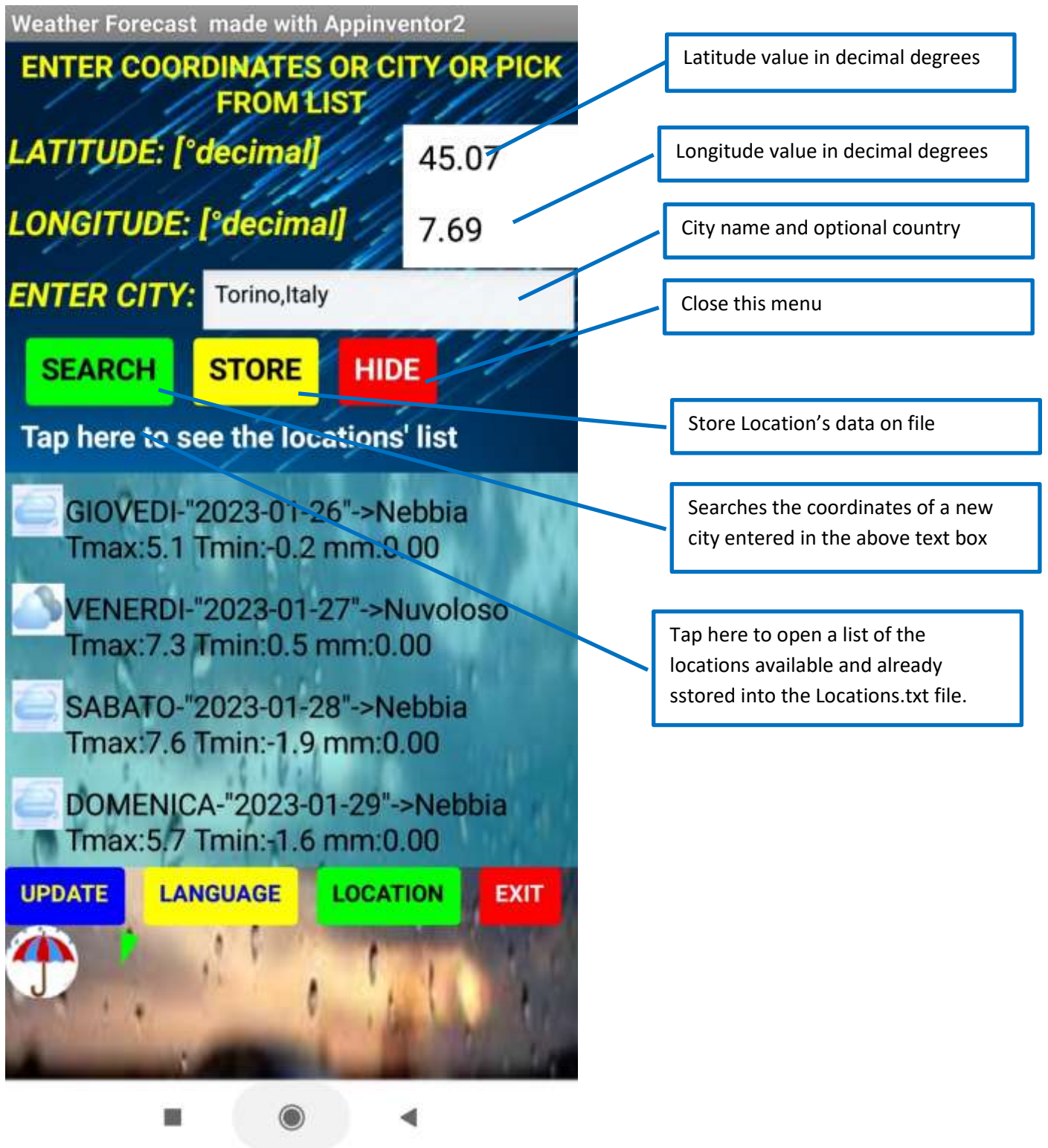


Figure 3 Location Settings Menu

This menu appears when the green LOCATION button has been hit.

Several possibilities are available:

- Latitude and longitude can be modified manually (though it is advisable not to do it). Modified values can be used temporarily when pressing "UPDATE" or stored permanently by pressing "STORE".
- A new city can be searched, by entering the name (and country, optional, separated by a comma ","), then pressing "SEARCH".
 - If the coordinates server is busy, a red popup appears asking for a new attempt, by pressing "SEARCH" again.
 - The red warning lasts on video for a couple of seconds, then disappears.
 - When it has disappeared, by pressing again "SEARCH" a new attempt is done.
 - Until the server is busy the popup can rise again.
 - When the server is available, the new coordinates of the new city are displayed into the two text boxes of Lat and Lon, and the an updated weather forecast for that new city is shown automatically.
 - If the new city is stored ("STORE" button) the relevant dataset is written (appended) into the Locations.txt file and it also becomes the default city, stored into the Configuration.txt file. The next time the app is run, this location is the default one.
- By tapping on the text "Tap Here to see the locations' list" the following page is shown:

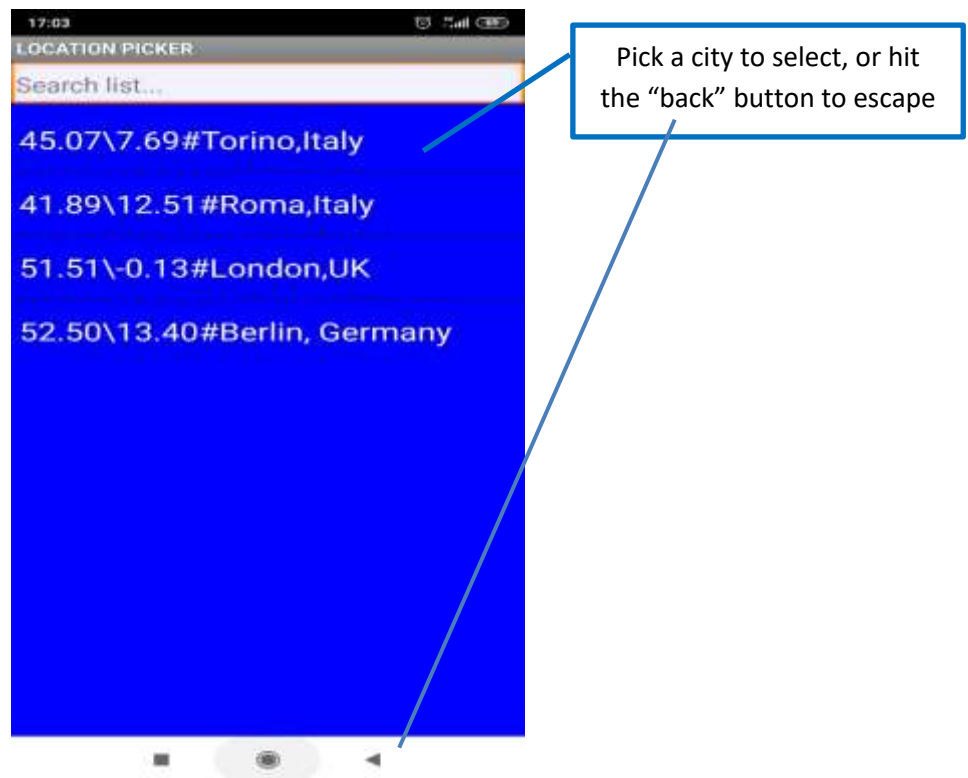


Figure 4 Cities list picker

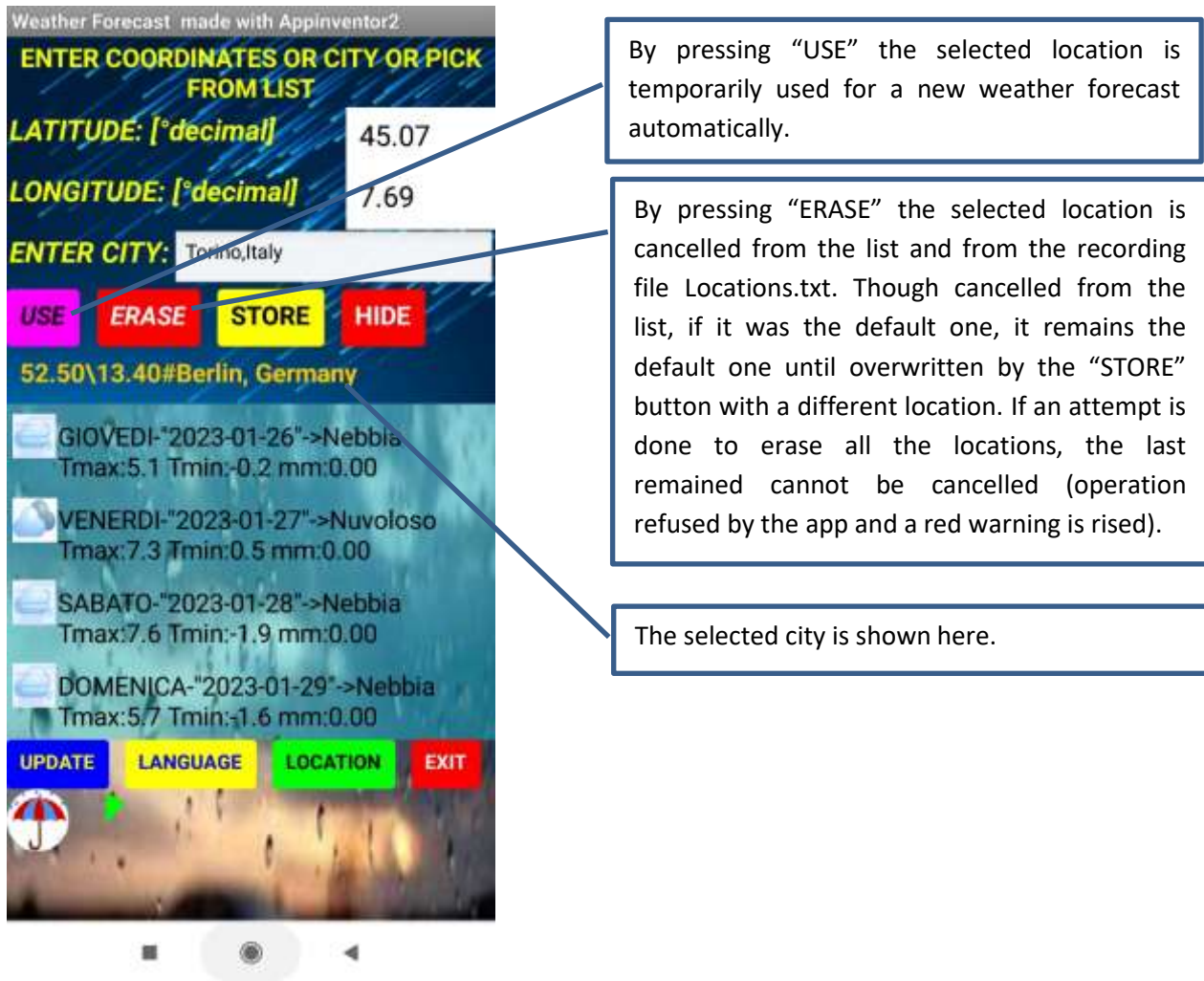


Figure 5 Location Use/Erase/Store

6 The Files

The language dictionaries, the icons, the background images are stored in the Asset directory. The Configuration and the Location files are stored in the "legacy" directory and can be viewed by means of a standard .txt viewer. Please take care not to modify them because their misuse can make the app not working.

6.1 Config.txt file

The contents of this file are read at the beginning of the application run. The first field is the language code (see comments below), then there are the latitude and the longitude of the location. The last data is the city name with optional, the country (for a better location). This file is written every time there is a change in the language, or if a new location is searched and stored (button "STORE"). If the newly stored location does not already exist in the Location.txt file, then it is added (appended) to the Location.txt file as well. If it is already there, it is not written a second time (in the Location.txt file).

Typical content is:

1,45.07,7.69,Torino,Italy;

Comments shall be put after the semicolon ";" while data fields shall be separated by commas ","

1) First field if =1 language file is Italian else if =2 in English

6.2 Locations.txt file

In this file the various locations selectable in a list picker popup, are contained. A dataset for every location is composed by latitude, longitude separated by a reverse slash “\” and the city name and country introduced by a “#”. Each dataset is separated by a semicolon “;”. Please take care not to write a “;” after the last dataset. This file can never be empty.

NOTE: These two files are updated automatically by the app. There is no need to edit them manually. The information above is given only for general knowledge.

Typical content is:

45.07\7.69#Torino,Italy;41.89\12.51#Roma,Italy;51.51\0.13#London,UK

7 Credits

The following extensions have been used in the app:

- TaifunTools: KeepScreenOn
 - <https://puravidaapps.com/>
- OlverlapView: AddLayer
 - VSatish13 www.casagbic.com
- TextBoxTools: ClearFocus
 - Gordon Lu <https://sites.google.com/view/appinventor-aicode?pli=1>

Many thanks to their respective authors.

8 Conclusion

The app is still far from being complete. Improvements can be done:

- by removing bugs (a bug is always lurking)
- by adding more details for each day (i.e., hourly forecast)
- by adding more languages
- by changing the language also of buttons and messages (and not only of the forecasts)

But these improvements, and others, are left to you! You have the source code and you can “enjoy” the challenge to modify it.