



# A Context-Aware Application to Monitor the Air Quality

Giacomo Cabri<sup>(✉)</sup>  and Gabriele Nocetti

University of Modena and Reggio Emilia, Modena 41125, Italy  
giacomo.cabri@unimore.it, 270435@studenti.unimore.it

**Abstract.** The widespread use of mobile devices allows developers to reach and inform more users than ever before through apps. This work presents the design and development of an application that allows anyone to monitor air quality levels, including pollen and pollution. The developed application retrieves official data and uses elements of context awareness to customize the user experience based on the *user's sensitivity* to specific particles. During the development process, special attention will be paid to interfaces and data presentation based on user characteristics and preferences. The project will focus on the Italian territory. Due to the richness in terms of biodiversity of the country, similar solutions exist only at the regional level without taking into account the user's sensibility. Our contribution is to show an application that join general purpose with context-awareness aspects.

**Keywords:** Pollen · Context-awareness

## 1 Introduction

Airborne particles, specifically *pollen-like* particles, can cause *allergic symptoms* in sensitive subjects. There appears to be a relationship, currently under study, between the level of concentration of these particles in the air and the symptoms experienced by individuals [2, 8, 13, 15]. This paper is not going to approach the stage of study or analysis to obtain future predictions of these phenomena. Instead, the intent is to develop a *tool* that, having reliable data and predictions available, provides an opportunity for anyone to learn about this topic.

Taking advantage of the diffusion of smartphones, in this paper we present the design and development of a context-aware *mobile application*. Choosing this development target opens up many possibilities for gathering information to integrate a *context-aware* system. The goal is to combine the world of air quality, in order to inform users interested in this topic, with the opportunities offered by the development of mobile platforms to customize the experience.

An application covering this topic (air quality) can certainly lead to the implementation of features to provide benefits for the user, such as allowing him or her to monitor levels of particles in the air and to receive alerts if changes are expected in the coming days. Applying *context awareness* to the alerting

functionality can *change* its behavior and operation, to the advantage of the user with *notifications* of information that may actually be useful.

The contribution of this paper is to show the design and the implementation of an application that from the one hand is general and can be applied in different areas, but at the other hand it is based on context-awareness, which makes it adaptable to specific situations.

The rest of the paper is organized as follows. After discussing some related work (Sect. 2), we will propose the requirements of our application (Sect. 3). Then, we will present the implementation (Sect. 4), detailing the user interface (Sect. 5). Finally, we conclude the paper with some remarks and future work (Sect. 6).

## 2 Related Work

In the context of air quality monitoring applications, we consider smartphone apps that can be installed by anyone and that provides data and/or predictions of the particles present in the air of a specific area.

The first example [9] is only available in the Austrian territory and, thanks to the improved *data prediction* model, has also allowed the creation of a personalized real-time experience based on the user's symptom registration.

Another system, called RealForAll [16], uses an *AI-based* subsystem to achieve independence of *aerobiology* experts and allows the use of a symptom calendar for easy comparison of measurements.

Other examples are available for everyone in the apps markets (*App Store* & *Play Store*), and according to [17] they may have disadvantages. Many of these apps, covering vast territories or even the whole world, have the risk of presenting, in a modern and attractive interface, approximate data that are not specific to a particular area. The problem of the correctness of the collected data must be addressed and one should try not to simplify them during the presentation phase.

In these applications, common *context-awareness* elements can be found:

- presenting the user with data retrieved as close as possible to their *location*;
- providing the ability to *indicate*, on a day-to-day basis, what symptoms the user is experiencing or what pollen particles is *sensitive* to.

Solutions related to the chosen territory, Italy, are often limited to local or, at most, regional areas due to the biodiversity of the area in question. Moreover, mobile applications dealing with these territories currently lack context-awareness elements to help the user visualize the data, something that are present in more modern applications that cover larger territories. In conclusion, solutions that are limited to work in one or more countries are the most effective and accurate because of the data provided by national services. In contrast, those that cover the whole world offer data covering a wider area but with lower precision.

Our proposal will focus on the *Italian territory*, using the national data offered. Therefore, our goal is to provide a *ready-to-use* application with context-awareness elements, working with the data obtained. Nevertheless, the proposed model of application aims to be general and can be applied to other countries.

### 3 Application Requirements

This section presents the functionalities and requirements necessary to ensure the proper operation of the application. After a general explanation of the existing functionalities, a focus will be placed on the elements of context awareness present.

#### 3.1 Features of the App

The application consists of several *screens*, the most important of which is the home screen. This main page is designed to present the user with only the information they need at that moment, allowing them to explore additional fields by navigating through additional screens. Secondary screens include a page that groups particles of a specific type, followed by a study page that presents data and information related only to the particle being examined. The user can select a particle monitoring station of his or her choice through a search screen to view the data obtained from it. The screen that allows the user to report any symptoms caused by particles in the air and the settings screen complete the list.

Regarding the types of particles presented, in addition to those belonging to the common group called “pollen”, it was decided to include *pollutant particles*. With the aim of providing a comprehensive picture of air quality and preventing possible allergic effects caused by it [1], pollution levels will be included as types of particles.

Many of the screens described above are necessary to present the user with the key functionalities of the project. They are listed below in order of *dependency*:

1. visualization of the latest available data;
2. calculation of the not available particle value forecasts for the next few days;
3. geolocalization of the user to provide information based on their location;
4. navigation between available monitoring stations;
5. diary, providing the user with the ability to report particles to which they are most sensitive;
6. sending notifications about changes in particle values.

#### 3.2 Context Awareness Structure

Data that define the context of the application can be divided according to how it is collected. The first type covers data derived from device information, such as time, day, and location. The second includes the user’s personal data, which can only be obtained through the user’s interaction with a screen.

In Fig. 1, it is possible to see how these data is structured to enable the implementation of the context aware service. One can see the requirements for gaining the data and their strict dependence on it to achieve the end result (providing a personalized notification to the user). It is therefore clear that the user's personal status data *depend* on particle level information, which in turn is retrieved through the location of the device.

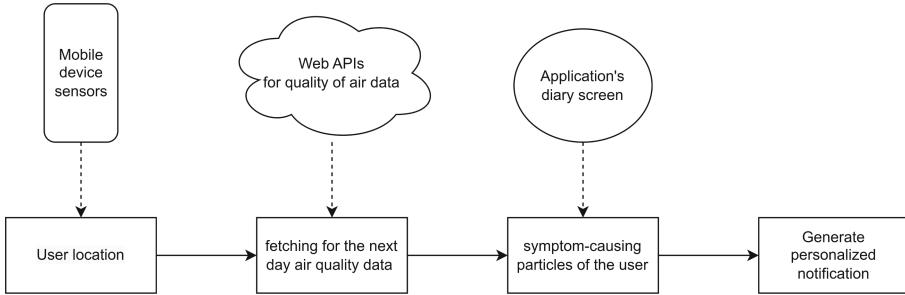


Fig. 1. Context-Aware components

### 3.3 Notification System

This context-based system will allow the user to receive a maximum of one personalized notification per day. When the application detects changes (increases or decreases) in the levels of the most prevalent particles in today's and tomorrow's data at the last recorded location, the notification is sent. Related to this are two concepts that need to be explained and defined to understand how the system works.

It was decided to *group* particles belonging to a common family into types: spores, pollutants, tree pollen and grass pollen. This choice is not intended to replace the level representation of each particle present, listed by the user, but to facilitate the visualization of information. The subdivision, also applied at the structural level, will condition all the functionality presented, including the context-based notification system.

Each particle is associated with a daily value and a set of *threshold values* that will determine the level reached by that particle on that day. Chosen as absent, low, medium and high, once all levels of a type are collected, the average level of each type will be obtained. In connection with the notification system, the way to determine whether to send a notification is based on the presence of changes in the values belonging to the mean or high levels of the most prevalent type on the current or next day.

At this point, the integration of the user's personal data into the notification system occurs in two stages. The *first* stage involves data collection, which is done through the *symptom report page* of the application. Once a day, the user can fill

in the screen by entering the number of hours spent outdoors and the intensity of symptoms felt. This data will be processed along with particles belonging to the most prevalent type detected that day. At the end of this stage, a *weight* is assigned to the detected particles (which may have caused those symptoms). During the *second* stage, when the notifications are generated, these weights will be applied when comparing particle levels. The ways in which they may affect will be explained in the following sections.

The notification system includes an additional notification, concerning only changes in *pollutant levels*, independent of personal data. In conclusion, the system is working from the first app-launch, but will be customized once the user interacts with the diary screen.

## 4 Application Implementation

After an introduction regarding the framework used for development, the implementation of the features in the application will be broken down and described below.

### 4.1 The Chosen Framework

The *Flutter framework* was selected for the implementation of all application features. Focusing on the development of mobile platforms for *Android*, the “material” library [5] included in the framework made it possible to create interfaces that were *OS-consistent* and easy to use.

The framework is based on the *Dart* programming language, which is oriented toward web and mobile development and places great emphasis on handling asynchronous operations and *Future* data types, suitable for the structure of the chosen application.

### 4.2 Data Retrieval

The sources selected, based on the requirement of data reliability, are two. *POLLnet* [14], the Italian aerobiological monitoring network of the National System for Environmental Protection (SNPA), for pollen and spores. The open source *Open-Meteo* [11] project concerning air pollutants. The last one allows interfacing with data from the Copernicus Atmosphere Monitoring Service (CAMS) [3]. Both of these sources will be queried through their *APIs*, returning *JSON data* processed with *Dart*. The threshold values for particle levels were retrieved from the *POLLnet* service and the guidelines [12] of the World Health Organization (for air pollutants).

As noted in Sect. 2, the basic requirement for the data used in this context is *reliability*. This requirement is luckily met, in the borders of *Italy*, by the *POLLnet* service. When delineating the quality of the data, which includes both the diversity of pollen types identified and the additional information attributes,

it becomes evident that by restricting the scope to the Italian territory, a degree of *accuracy* quite different from that of other international services is achieved.

Among these considerations, there is also to be added the *amount* of monitoring stations located across the various regions of Italy. Taken together, these make up a total of 64 stations. This collective infrastructure, combined with the rich quality of the data, guarantees end users of the application (which is based on this network) a *remarkable* standard of accuracy and data quality.

It is important to emphasize, as mentioned in the abstract, the fact that we work only with data obtained from official sources. Among these, the most important ones are those related to the forecast of particle levels in the coming days. For air pollutants, CAMS provides readily available and already calculated data. However, getting data predictions is different for those concerning pollen and spores, which are obtained from POLLnet. Since POLLnet provides data only up to the previous week, we needed to find a way to calculate data for the following days. This was done following guidelines [7], but it is only meant to be a temporary solution until more reliable data regarding pollen and spores forecasts are obtained.

After obtaining the device location, data are *retrieved* from Open-Meteo using latitude and longitude data. Retrieving data from POLLnet is more challenging, as it requires determining the closest monitoring station to the device location using the *Haversine formula*, which calculates the distance between two geographic coordinates.

To ensure a satisfying *user experience* in different situations, especially considering that the application is intended for mobile devices, two specific use cases in particular were covered. First, in case there is no GPS signal, the system ensures that users have access to data from all monitoring stations, in *browse-mode*. Second, even when GPS functionality is active, if the nearest monitoring station has no data available, the app will show information from the nearest *working* monitoring station to users.

### 4.3 Presentation Layer

In terms of data visualization, the user has access not only to the “Home” summary screen, but also to the tab for each detected particle. This tab features a *graph*, drawn using external library<sup>1</sup>, that shows the user the latest official data retrieved based on the chosen monitoring location.

Each screen of the application presents particles categorized by type and sorted by intensity level. In order to achieve a *common view* of these screens, it was decided not to apply differences in the sorting depending on the sensitivity of the user regarding certain particles. Screenshots of the application pages are available in the following sections.

---

<sup>1</sup> [https://pub.dev/packages/fl\\_chart](https://pub.dev/packages/fl_chart).

## 4.4 Storing the Data

External plugins can be used in Flutter to provide data storage services. The first plugin<sup>2</sup> uses Android’s “Shared Preferences” to store *non-sensitive* information, such as the flag that enables or disables notifications. To speed up the application and reduce the number of calls to external sources, a *caching service* was implemented using another plugin<sup>3</sup>. Finally, a plugin that prioritizes *security* was chosen to store the most sensitive data, such as the user’s last detected location and the particles to which the user is most sensitive. This plugin<sup>4</sup> works based on the *KeyStore system* when in the *Android* environment.

## 5 An Overview of the User Interface

### 5.1 Homepage and Details of Particles

Every time the application starts, the home screen is presented. Depending on the location of the device, it shows the highest-level particle type in the foreground. The screen is divided into *tabs* representing today and the following days, and is enhanced by a small box showing the *weather values* for the selected area.

Each type of particle can be viewed in detail on its specific page, which also shows the *distance* and accuracy of the data by location. In addition, a *detailed tab* for each individual particle can be viewed, with a corresponding graph. This tab shows the data source, threshold values for the levels, and a brief description. In the upper right corner, the search icon allows users to search for monitoring stations and view data for that location. The button in the lower right corner provides access to the screen for *reporting* any symptoms caused by particles in the air. It is only visible on the home screen and becomes available again the next day after being used.

All the screens just described can be viewed in Fig. 2, and all the *logos* representing the types were generated by Midjourney [10].

### 5.2 Symptom Report Page and Diary

Thanks to the screen showed in Fig. 3 left, the user can choose the level of *reported symptoms* and the number of *hours spent outdoors*. When finished, the process explained in Sect. 3.3 is performed and the data are saved. Only data on particles exceeding a minimum level of presence in the air are saved. For demonstration purposes, the medium level was chosen.

These data, sorted from the most recent, can be viewed in the diary screen from the settings menu. Protected by *biometric* authentication (fingerprint), the data can be viewed and, if necessary, deleted.

<sup>2</sup> [https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences).

<sup>3</sup> [https://pub.dev/packages/flutter\\_cache\\_manager](https://pub.dev/packages/flutter_cache_manager).

<sup>4</sup> [https://pub.dev/packages/flutter\\_secure\\_storage](https://pub.dev/packages/flutter_secure_storage).

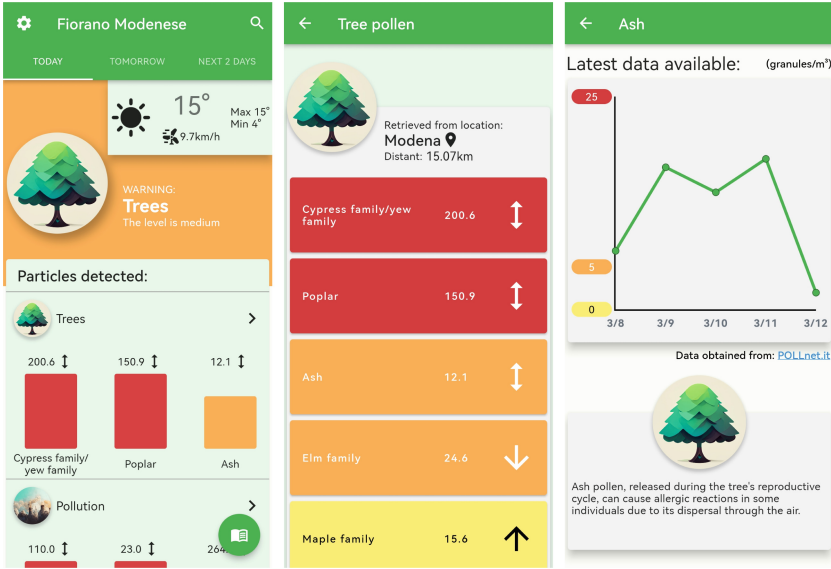


Fig. 2. Homepage (left), view of the type (center) and particle details (right)

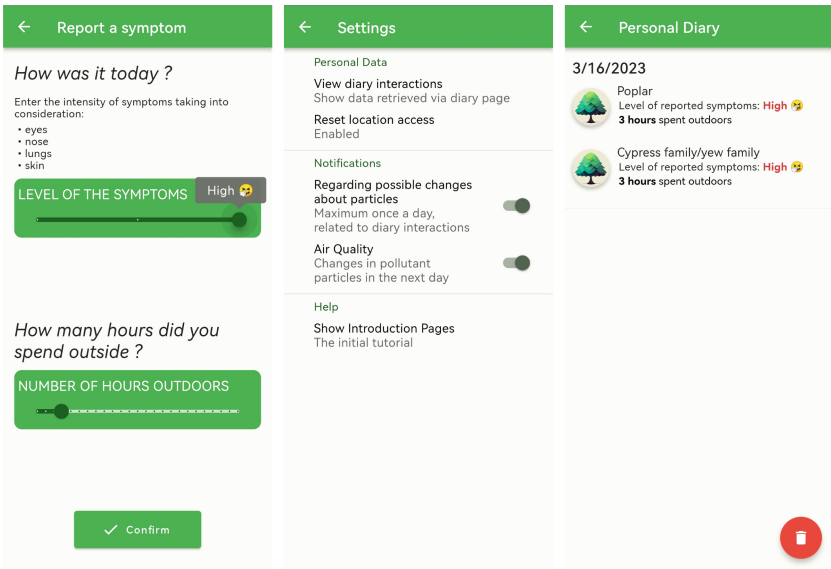


Fig. 3. Symptom reporting page (left), settings (center) and diary screen (right)

### 5.3 Notifications

Notifications are customized according to the last recorded location and particles saved in the *diary*. This may be clearer after an example.

The device is located in Treviso (Fig. 4), where the level of polluting particles is higher than tree pollen. A notification is displayed, warning of increased pollution, because the user has enabled air quality notifications. In addition to this, it is noted that the level related to the *cypress family* is also increasing, and in this case there are 2 possible scenarios:

1. if the diary is *empty*, priority will be given to the pollution type;
2. if the diary has *marked* the cypress family enough times, the notification will alert the sensitive user about the increase of that particle.

In the second scenario, the diary is filled with the values in Fig. 3. The high level of symptoms presented, combined with the number of hours spent outdoors, gave the particles of the cypress family *more weight* than the increase in pollution. In addition to these exemplary cases, it was essential to ensure the perfect functionality of the custom notification distribution logic in a wide range of scenarios.

For this purpose, along with the development of the necessary operating code, an exhaustive set of *unit tests* [6] was made, examining the behavior of the notification system in every possible permutation. This testing includes a total of 256 distinct *configurations* based on the different *intensity ranges* that 8 particles of 2 different pollen types can manifest on a given day (i.e., absent, low, medium, high). The tests also include the interaction of 8 potential “weights” values of the particles, generated by interactions with the *journal* screen, all of which are accurately stored on the device.

The weight combinations subjected to testing includes values associated to two distinct types of pollen particles, in order to introduce a *dynamic* element that influences the behavior of the notification system. These weights are systematically grouped across the *full spectrum* of possible pollen intensity combinations, producing a complete examination of system *responsiveness*.

Together with other *practical* tests conducted on devices of different brands, all of these tests played a key role in the development, significantly contributing to the refinement and optimization of the notification system.

### 5.4 Testing on Different Devices

The application was developed to work on screens with resolution higher than  $320 \times 480$  px. The interface was tested on devices belonging to all major Android smartphone manufacturers and did not present any particular problems.

The notification system, specifically when *showing* notifications, presented different behaviors depending on the device. From these tests it appears that on stock Android systems, without any special modifications by the producer, notifications are displayed almost all the time. The more the producer customizes



**Fig. 4.** Particle levels today (left) and tomorrow (center) in Treviso, notification when diary is empty (upper right) and when cypress family is in the diary (lower right)

her Android system in the area of background app management, the less notifications are likely to be displayed often. To solve this problem, it is often necessary for the user to change the system settings that regulate the background app’s behavior. The results collected from the tests turn out to be very similar to those of the “Don’t Kill My App” project [4].

## 6 Conclusions

In conclusion, this paper describes an *Android-based* application that allows a large number of users to *monitor* particle levels in the air over the Italian territory. A *context-aware notification system* has also been implemented, which customizes the user experience according to the user’s sensitivity to certain particles. The *innovative* approach of this system is highly appreciated as it is applied in a field of great interest, such as the health of ordinary citizens. Even more so in the area of Italy, where solutions of this type are only present at a regional level and lack context-awareness elements.

Thanks to the *cross-platform* framework on which the application is based, additional work in the near future includes support for Web, Apple, and Windows devices. Currently, the project can be compiled in JavaScript and run on any browser without modification. As a result, *platform-specific* functions, such as notifications and location permissions, will need to be converted to the new ones.

A further step would be to expand the service to operate *globally*, leveraging the CAMS pollution level retrieval service, which already offers data covering the whole world. This more complex future development will have to follow the same logic chosen for the Italian territory, relying on official data from the relevant county where it is possible.

However, the most important upcoming work will be to analyze the functionality of the application with the support of *aerobiology experts* and ensure that it is not only technologically reliable, as demonstrated, but also scientifically valid.

## References

1. Bartra Tomàs, J., et al: Air pollution and allergens. *J. Investig. Allergol. Clin. Immunol.* **17**, suppl. 2, 3–8 (2007)
2. Brito, F.F., et al.: Olea Europaea pollen counts and aeroallergen levels predict clinical symptoms in patients allergic to olive pollen. *Ann. Allergy Asthma Immunol.* **106**(2), 146–152 (2011)
3. Copernicus - C.A.M.S. <https://atmosphere.copernicus.eu/>. Accessed 14 Mar 2023
4. Don't kill my app project. <https://dontkillmyapp.com/>. Accessed 16 Mar 2023
5. material library - Dart API. <https://api.flutter.dev/flutter/material/material-library.html>. Accessed 14 Mar 2023
6. Online repository of the project. [https://github.com/noceg43/pollen\\_app](https://github.com/noceg43/pollen_app). Accessed 21 Sept 2023
7. Guidelines from POLLnet (in Italian). [https://www.isprambiente.gov.it/files2017/pubblicazioni/manuali-linee-guida/445364\\_Manuale\\_linee\\_guida\\_151\\_17.pdf](https://www.isprambiente.gov.it/files2017/pubblicazioni/manuali-linee-guida/445364_Manuale_linee_guida_151_17.pdf). Accessed 14 Mar 2023
8. Jantunen, J., Saarinen, K., Rantio-Lehtimäki, A.: Allergy symptoms in relation to alder and birch pollen concentrations in Finland. *Aerobiologia* **28**, 169–176 (2012)
9. Kmenta, M., Bastl, K., Jäger, S., Berger, U.: Development of personal pollen information-the next generation of pollen information and a step forward for hay fever sufferers. *Int. J. Biometeorol.* **58**, 1721–1726 (2014)
10. Midjourney website. <https://www.midjourney.com/>. Accessed 17 Mar 2023
11. Free Open-Source Weather API — OpenMeteo. <https://open-meteo.com/>. Accessed 14 Mar 2023
12. Organization, W.H., et al.: Who global air quality guidelines: particulate matter (pm2.5 and pm10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide: executive summary. WHO global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide (2021)
13. Rapiejko, P., Stanlaewicz, W., Szczygielski, K., Jurkiewicz, D.: Threshold pollen count necessary to evoke allergic symptoms. *Otolaryngologia Polska= The Polish Otolaryngol.* **61**(4), 591–594 (2007)
14. SNPA - Sistema nazionale protezione ambiente (in italian). <https://www.snambiente.it/dati/pollini-e-spore-fungine/>. Accessed 14 Mar 2023
15. Tegart, L.J., et al.: 'pollen potency': the relationship between atmospheric pollen counts and allergen exposure. *Aerobiologia* **37**(4), 825–841 (2021)
16. Tešendić, D., et al.: Realforall: real-time system for automatic detection of airborne pollen. *Enterprise Inform. Syst.* **16**(5), 1793391 (2022)
17. Travaglini, A., et al.: Approximate or accurate? efficacy of daily use of weather and air quality mobile applications for pollen allergy sufferers? *Pediatric Allergy Immunol.* **33**, 41–43 (2022)