

App Meteo

Matteo Miggiano Alessio Panarese

April 30, 2021

1 Introduzione

Il progetto, assegnatoci dal docente Vanni Galli, consiste nello sviluppo di un'applicazione mobile in due versioni, una Android e l'altra cross-platform con Xamarin. Ciò al fine di apprendere al meglio le conoscenze base per la programmazione di applicazioni mobile. In particolare, nel nostro caso, essa consiste in un software meteorologico, nel quale è possibile aggiungere località a piacere e visualizzarne le informazioni relative al meteo. Essa può fare inoltre utilizzo del GPS per mostrare i dati del luogo nel quale ci si trova. Tra gli altri requisiti troviamo un sistema di notifiche, per segnalare all'utente se la temperatura diminuisce o aumenta entro una determinata soglia, e l'ausilio di database Room per permettere il salvataggio delle località. Tramite delle richieste effettuate con le API di OpenWeatherMap viene ricevuto un JSON, dal quale vengono poi estrapolati i dati.

2 Descrizione

La schermata iniziale, nonché quella principale, mostra all'utente la lista delle località, la loro temperatura attuale e un'immagine che indica la condizione meteorologica. Cliccando su una di esse, viene visualizzata la pagina dettagliata delle condizioni meteo della località selezionata, comprendenti: temperatura percepita, temperatura minima e massima, umidità e pressione atmosferica.

3 Struttura

3.1 Inserimento località

L'utente è in grado di inserire la località desiderata, premendo l'apposito pulsante '+' e inserendo il nome di essa all'interno della schermata di popup. Per verificare la correttezza del nome della città inserita, viene eseguita una semplice chiamata http che in caso di positività, permette di conoscere anche i relativi dati atmosferici della località selezionata, ed in caso di negatività, viene notificato l'errore all'utente attraverso un toast. Un requisito fondamentale era quello di permettere all'utente finale di poter inserire all'interno dell'applicazione, la

misurazione meteorologica della località presso la quale si trova in quel determinato istante. Questo è reso possibile dall'utilizzo della posizione GPS fornita dal dispositivo sul quale è installata l'applicazione.

3.2 Database

Per la memorizzazione delle località inserite all'interno dell'applicazione da parte dell'utente, è stata utilizzata la libreria Room, la quale permette una facile interazione tra il database ed il codice Java nativo. All'interno di quest'ultimo è stata inserita una classe di tipo singleton, identificata con il nome di DBManager, la quale è il fulcro delle chiamate al database sia per quanto riguarda la lettura che per quanto riguarda la scrittura dei dati.

3.3 Gestione API

Per poter fornire all'utente i dati relativi alle città inserite, sono state utilizzate delle chiamate HTTP alla OpenWeatherMap API. Queste ultime vengono gestite come task asincroni, richiamati al momento del bisogno all'interno delle classi che necessita di ottenere o aggiornare i dati dell'applicazione. Sono state implementate tre classi per gestire le modalità di richiesta necessarie per rispettare i requisiti: una prima classe viene utilizzata per ottenere i dati dall'API utilizzato come input le coordinate della posizione presso la quale si trova l'utente, ed in seguito quest'ultima viene inserita all'interno del database locale dell'applicazione; una seconda classe ha sostanzialmente le stesse funzionalità della precedente, l'unica differenza sta nel fatto che come input alla chiamata viene utilizzato il nome della località inserito dall'utente; l'ultima viene utilizzata per aggiornare i dati delle città presenti nel database locale. Le risposte dell'API vengono fornite sotto forma di json, quindi è stato essenziale effettuare il parsing di queste ultime al fine di ottenere i dati necessari. Per eseguire tale procedura si è preferito non utilizzare librerie alle quali appoggiarsi, ma è stato utilizzato la sola classe JSONObject.

3.4 Notifiche

Un requisito base dell'applicazione è la presenza di notifiche. Queste vengono gestite dal NotificationManager che si occupa di gestire le PeriodicWorkRequest che vengono accodate all'interno di una coda. L'effettiva dichiarazione della notifica viene fatta all'interno della classe UpdateWorker, che come si può notare dal nome, si occupa di aggiornare i dati delle località presenti all'interno del database dell'app. Così facendo, è possibile comparare le misurazioni attuali con quelle precedenti, e nel caso queste ultime abbiano valori superiori o inferiori a valori di soglia stabiliti, verrà notificata all'utente il valore fuori soglia.

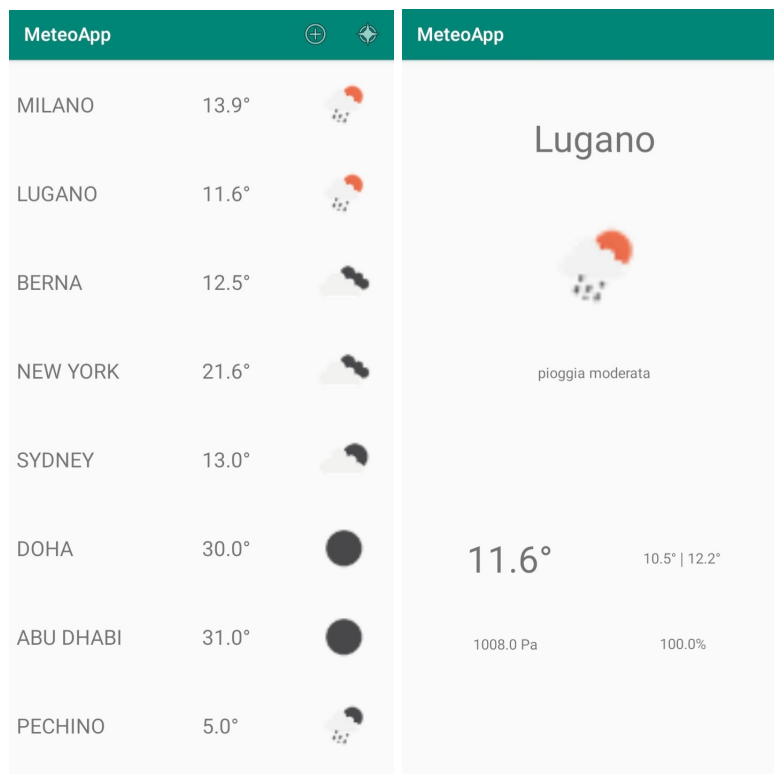


Figure 1: App design