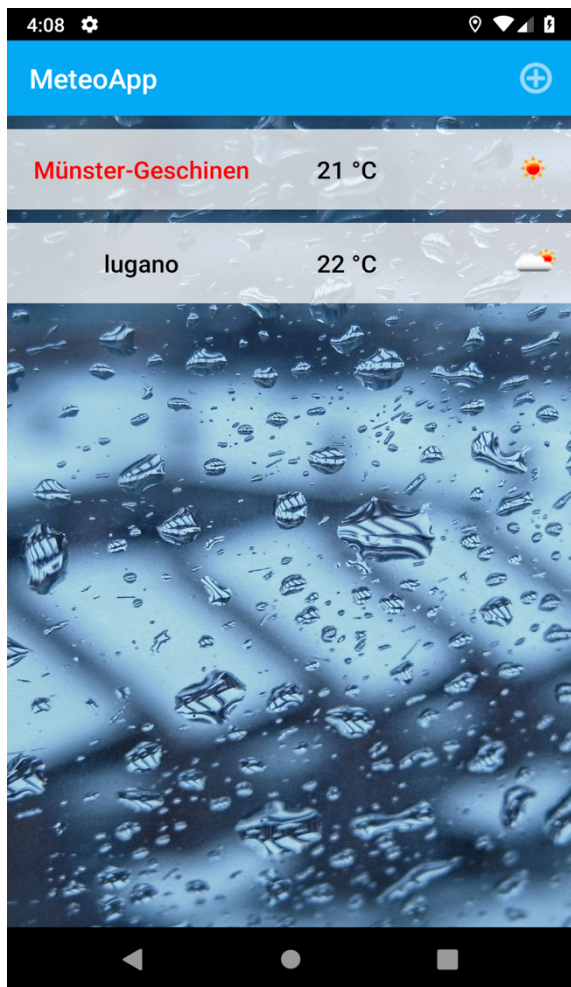


Progetto Android: Meteo App

Cem Koca, Brian Pulfer, Filippo Pura



1.0 - Introduzione

Il seguente documento rappresenta una breve introduzione sugli obiettivi richiesti e la loro realizzazione per il progetto del corso di laurea Sviluppo di Applicazioni "Mobile". L'applicazione in questione da sviluppare è "Meteo App", durante il semestre primaverile 2019 dagli studenti del terzo anno Koca, Pulfer e Pura presso il DTI della SUPSI.

2.0 - Obiettivi

L'obiettivo richiesto è la realizzazione dell'applicazione nativa "Meteo App" per la piattaforma Android, utilizzando l'IDE Android Studio. L'app deve essere in grado di monitorare il clima delle località aggiunte dall'utente e della posizione corrente, se questa è fornita tramite l'API dato.

L'utente deve avere la possibilità di aggiunta delle località desiderate, dove in seguito devono essere monitorate. Inoltre, l'applicazione dev'essere in grado di notificare all'utente la realizzazione di un evento, scelto dal programmatore. La posizione corrente dell'utente deve essere sempre distinguibile dalle altre località inserite.

3.0 - Sviluppo

3.1 - Strumenti di sviluppo

L'applicazione è stata sviluppata sull'IDE Android Studio, utilizzando i sistemi operativi MacOS X e Windows 10 ed è stata emulata sul dispositivo Nexus 5 e provata direttamente sul dispositivo Samsung S7 Edge.

Sono stati utilizzati Java e XML come linguaggi di programmazione, mentre per la persistenza dei dati SQLite.

Il lavoro di gruppo richiede l'integrazione continua, nel nostro caso si è deciso di utilizzare GIT, il cui repository è reperibile al seguente link: <https://github.com/FPura/MeteoApp>.

3.2 - Sistema Lista - Dettaglio

L'applicazione prevede due pagine, dove con la prima si ha la lista delle località aggiunte, la possibilità di aggiungerne nuove e la loro rimozione attraverso una pressione di almeno 1 secondo. Le due operazioni mostrano un pop-up, dove con la prima viene richiesto la località da aggiungere, mentre con la seconda viene richiesto la conferma della rimozione. Se abilitato, viene aggiunto automaticamente la posizione corrente e per essere distinta il nome viene scritto in rosso. A differenza delle altre località, che vengono salvate nel DB o rimosse a dipendenza dell'operazione, la posizione corrente non è salvato e non è possibile la sua rimozione.

La seconda pagina mostra unicamente la meteo della località selezionata.

3.3 - Geolocalizzazione

Per poter fornire all'utente le informazioni meteorologiche relative alla sua posizione corrente, senza che esso debba specificare la sua locazione, è necessario all'applicazione poter accedere alla posizione corrente del dispositivo.

In Android, il permesso per la localizzazione è di tipo **dangerous**, e necessita perciò dell'autorizzazione dell'utente a runtime o tramite attivazione manuale dalle impostazioni. Questa operazione viene eseguita facilmente, dichiarando nel **manifest** i permessi desiderati e sovrascrivendo la funzione che ritorna il risultato.

La libreria utilizzata invece è la *Smart Location Library*, che permette di estrarre le coordinate geografiche del dispositivo in maniera accessibile.

L'aggiornamento della posizione corrente avviene ogni minuto, tempo stabilito da noi inizializzando l'API.

3.4 - Database

La persistenza dei dati dell'applicazione richiede un meccanismo di salvataggio. È stato scelto la libreria SQLite (DBMS), che fornisce ottime funzionalità, è open-source ed è un data base relazionale.

I file salvati da SQLite sono persistenti nel **sandbox** dell'applicazione, ovvero un piccolo filesystem fornito dal sistema operativo Android nella memoria flash del dispositivo. Il sandbox non è accessibile ad altre applicazioni al di fuori della proprietaria.

3.5 - Estrazione delle informazioni sul meteo

L'estrazione dei dati meteorologiche, dato il nome o la sua coordinata avviene l'API di OpenWeatherMap. La sua integrazione è alquanto banale, avviene nel file build.gradle.

Questa API consente di mandare richieste HTTP e ritornano dati meteorologiche desiderate. Nella nostra applicazione, usufruiamo solo di tre funzionalità, le prime due riportano informazioni meteorologiche sulla posizione data in base al nome o alle coordinate, mentre l'ultima ritorna l'icona della meteo richiesta.

3.6 - Notifiche

In Android la metodologia corretta per mandare notifiche in background, anche quando l'applicazione non è visibile sarebbe tramite i **Service**. Un service è un componente di applicazione che permette l'esecuzione in background di varie operazioni anche a lungo termine. Nel nostro caso, l'API che utilizziamo Smart Location esegue già questa funzionalità, quando le viene chiesto di monitorare la posizione in continuo.

Abbiamo deciso di usufruire la possibilità che ci è stato dato da Smart Location, infatti dentro la funzione che viene richiamata ogni minuto, facciamo un controllo sulla temperatura, e se questa è inferiore ai 13 °C viene lanciata una notifica all'utente, che lo avverte della temperatura bassa nella sua posizione corrente.

4.0 - Conclusione

In conclusione, gli obiettivi prefissati sono stati raggiunti e abbiamo anche aggiunto qualche funzionalità extra, come la possibilità di cancellare località, il controllo sull'aggiunta di doppioni o località fantasma e la gestione della rotazione del dispositivo.

Il progetto in questione è stato molto interessante e istruttivo, sicuramente queste nozioni ci saranno utili in futuro per sviluppare applicazioni native su Android.