

SUPSI

Progetto Android - Meteo App

Studente/i

Alessandro Bianchi

Maura Clerici

Zyrl Errol Gatchalian

Relatore

-

Correlatore

-

Committente

Vanni Galli

Corso di laurea

Ingegneria Informatica

Modulo

**M02074 - Sviluppo di
applicazioni mobile**

Anno

2019

Data

15 aprile 2019

STUDENTSUPSI

Indice

1	Requisiti	1
2	Implementazione	3
2.1	Applicazione di tipo List-Detail	3
2.2	Aggiunta di nuove location	3
2.3	Utilizzo GPS	3
2.4	Persistenza delle locations	4
2.5	Controllo periodico delle temperature e notifiche	4
2.6	Invio richieste HTTP	4

Capitolo 1

Requisiti

Lo scopo del progetto è lo sviluppo di un'applicazione Android per visualizzare i dati meteo di diverse località, con i seguenti requisiti:

- Applicazione di tipo List - Detail
- Possibilità di aggiungere nuove location manualmente (con popup, nuova schermata)
- Utilizzo del GPS per leggere la posizione corrente e mostrarla in lista
- Salvataggio delle location inserite dall'utente su database SQLite
- Controllo periodico (tramite Background Service) delle temperature; invio di notifiche se la temperatura locale scende / sale sopra una certa soglia

Capitolo 2

Implementazione

2.1 Applicazione di tipo List-Detail

Le applicazioni di tipo List-Detail costituiscono un formato molto popolare per quanto riguarda le applicazioni mobile: si compongono di solito di due schermate principali:

- List: una lista di entries costituita in questo caso dall'elenco delle location di cui l'utente vuole conoscere le informazioni meteo
- Detail: una schermata di dettaglio di una delle entries della schermata List: nel nostro caso, verranno mostrati Di solito è possibile raggiungerla eseguendo un tap sul nome nella lista della precedente schermata

2.2 Aggiunta di nuove location

Grazie alla presenza del bottone "+" presente nella toolbar dell'applicazione, è possibile accedere al dialog per l'inserimento di una nuova location di interesse. Dopo aver inserito il nome della location, è sufficiente premere il bottone "OK" per inserirla nel database.

2.3 Utilizzo GPS

Per l'uso del GPS, l'applicazione fa ricorso alla libreria SmartLocation per gestire in modo semplice e veloce il dispositivo. È necessario, prima di poter fare uso del GPS, implementare la richiesta all'utente dei permessi necessari per il suo utilizzo.

Periodicamente, l'app chiede al GPS la posizione attuale: in questo modo, nel caso l'utente si muovesse, la sua posizione viene mantenuta aggiornata. Questo meccanismo è gestito da un listener che aggiorna la posizione dell'utente ogni 5 secondi.

I dati ritornati dal GPS sono corrispondenti a latitudine e longitudine a cui il dispositivo si trova.

2.4 Persistenza delle locations

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

2.5 Controllo periodico delle temperature e notifiche

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

2.6 Invio richieste HTTP

Le richieste HTTP vengono mandate ogni minuto per ricevere aggiornamenti sulla temperatura della posizione corrente in modo da inviare una notifica nel caso in cui la temperatura supera un certo valore. Inoltre le richieste vengono mandati quando l'applicazione richiede (si preme su una città o posizione corrente) le condizioni atmosferiche di una delle città presenti nella lista.

Per l'invio di richieste http abbiamo dovuto apportare diverse modifiche:

- aggiungere i permessi nel manifest
- creare le classi apposite alle richieste asincrone

Per poter fare la richiesta ci siamo basati maggiormente sulla classe `MeteoFetcher` che si occupa di:

- tenere la chiave per poter fare le richieste (ottenuta da una previa registrazione al sito OpenWeather)
- aprire la connessione verso l'url di OpenWeather
- costruire il messaggio di richiesta
- prelevare i dati di risposta con uno stream di byte
- convalidazione dei dati ricevuti
- parsare il messaggio json ritornato dal sito

- prelevare i dati parsati e metterli in un oggetto model Location temporaneo

Questa classe è gestita da un'altra classe chiamata `MeteoTask` in modo da far partire in background la richiesta affinché l'applicazione non si blocchi fino alla ricezione della risposta da parte del sito.