Map Connectivity

Relazione progetto

Cheikh Ibrahim · Zaid

Matricola: 0000974909

zaid.cheikhibrahim@studio.unibo.it

Paris · Manuel

Matricola: 0000997526

manuel.paris@studio.unibo.it

Anno accademico
2022 – 2023

Corso di Laboratorio di Applicazioni Mobili
Alma Mater Studiorum · Università di Bologna

Indice

1 Introduzione

L'applicazione `MapConnectivity` è un software sviluppato in Kotlin che permette di registrare e visualizzare su una mappa, tramite una griglia a riquadri e uno schema di colori prefissato (rosso = pessimo, giallo = medio, verde = ottimo), le misurazioni della potenza del segnale LTE, Wi-Fi e il rumore misurato in dB nella zona circostante.
È possibile effettuare misure in tre modi diversi:

1. Manualmente - tramite la pressione di un bottone,
2. Automaticamente - viene effettuata una misura ogni volta che l'utente entra in un nuovo riquadro,
3. Periodicamente - viene effettuata una misura ogni intervallo scelto dall'utente.

Inoltre, è possibile scambiare le proprie misure con un altro dispositivo tramite file o Bluetooth.

1.1 Main Activity

La schermata principale dell'applicazione (Figura 1).
All'avvio, una volta concessi i permessi richiesti, viene mostrata la mappa con la griglia. Inoltre è presente un indicatore che mostra la modalità attuale di visualizzazione, ovvero se i colori mostrati sulla griglia si riferiscono a misure relative a LTE, Wi-Fi o dB.
Il bottone `Misura` effettua una misura calcolando l'intensità del valore LTE, Wi-Fi e dB localizzandoli nella posizione attuale dell'utente.
Sotto la mappa sono presenti, inoltre, due bottoni che portano alle altre schermate dell'applicazione: Impostazioni (Capitolo 1.2) e Scambio dati (Capitolo 1.3).

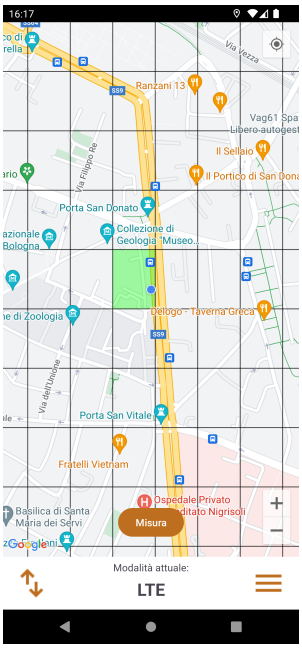


Figura 1: Schermata principale

Premendo su un riquadro (Figura 2), è possibile mostrare una schermata riepilogativa di tutte le misure presenti in quel riquadro. Selezionando una specifica misura, è possibile visualizzarne nel dettaglio i dati oppure eliminarla.

(a) Premendo su un riquadro

(b) Premendo su "Mostra di più"

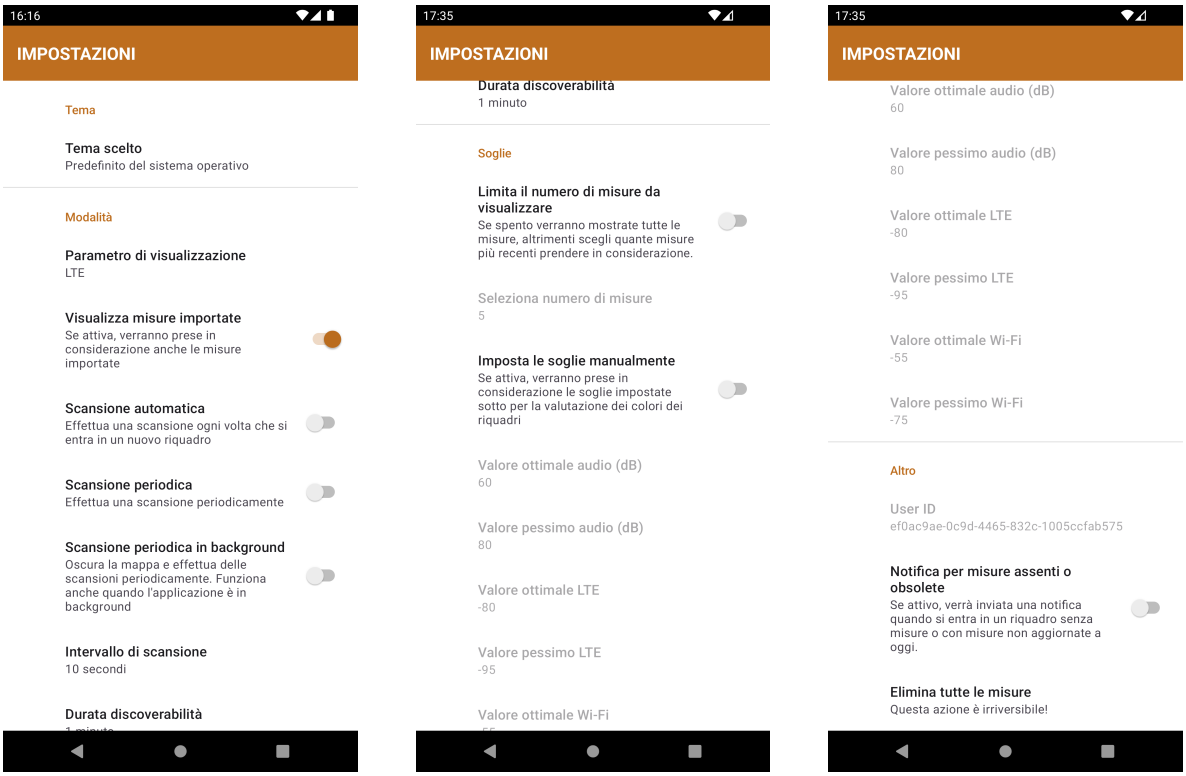
(c) Premendo su una misura

Figura 2: Schermata riepilogativa

1.2 Settings Activity

Schermata contenente le impostazioni per le preferenze e modalità di misurazione (Figura 3). L'utente ha la possibilità di scegliere:

- Il tema, tra chiaro, scuro e predefinito del S.O.,
- Il parametro di visualizzazione, tra LTE, Wi-Fi o dB,
- Se mostrare o no le misure importate,
- Di attivare una particolare modalità di scansione, tra automatica, periodica e periodica in background,
- L'intervallo di scansione, ovvero ogni quanto effettuare una scansione quando è attiva quella periodica,
- La durata della discoverabilità, ovvero quanto rimanere discoverabile durante l'esportazione via Bluetooth,
- Di limitare le misure da tenere in considerazione da visualizzare sulla mappa e in caso affermativo ne può specificare il numero,
- Di impostare delle soglie manuali per la visualizzazione dei colori su mappa,
- Di attivare la funzionalità di notifica quando si entra in un riquadro con misure non recenti al giorno odierno,
- Di cancellare le misure (tutte, solo quelle effettuate dall'utente o solo quelle importate).



(a) (b) (c)

Figura 3: Schermata delle impostazioni

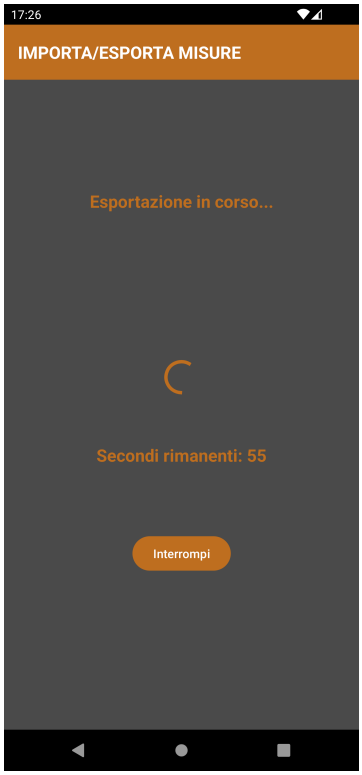
1.3 Swap Activity

Schermata contenente le opzioni di scambio dati (Figura 4a).
L'utente può scegliere di scambiare i dati con un altro dispositivo o tramite l'invio di file o tramite Bluetooth.
Ci sono quindi 4 opzioni:

- Importazione da file: verrà chiesto di selezionare il file `.mapc` dall'archivio del dispositivo.
- Esportazione da file: verrà creato un file `export_data-ora.mapc` contenente tutte le misure presenti nel database, tranne quelle importate, e verrà chiesto all'utente il modo di inviare tale file.
- Importazione da Bluetooth: verranno visualizzati i dispositivi nelle vicinanze discoverabili tramite Bluetooth e selezionandone uno verrà inizializzata la connessione per importare le misure automaticamente (Figura 4b).
- Esportazione da Bluetooth: in base alla durata di discoverabilità presente nelle impostazioni, il dispositivo diventerà discoverabile dagli altri per quella durata e alla richiesta di importazione da parte di un dispositivo, gli invierà tutte le misure presenti nel database, tranne quelle importate (Figura 4c).

(a) Schermata generale

(b) Schermata di importazione tramite Bluetooth



(c) Schermata di esportazione tramite Bluetooth

Figura 4: Schermata di scambio dati

2 Dettagli implementativi

2.1 Permessi

Da Android 11 le richieste dei permessi sono cambiate[1]. Per aggirare questo problema si è deciso di mostrare un dialog che, in caso l'utente vieti i permessi oltre il nuovo limite imposto, gli notifica l'assenza dei permessi e lo indirizza alle impostazioni dell'applicazione per autorizzarli manualmente.
Tale procedura è stata applicata per le richieste dei permessi di posizione, notifica, bluetooth e microfono. Per i permessi iniziali di posizione, dato che sono fondamentali per il funzionamento minimale dell'applicazione, l'utente è forzato a fornire i permessi nelle impostazioni per proseguire con l'utilizzo dell'applicazione.

2.2 Mappa

Per la visualizzazione della mappa, sono utilizzate le `API` di `Google Maps` ed è implementata nella Main Activity tramite un fragment di classe `SupportMapFragment`. Viene inizializzata, tramite la funzione `loadMap`, dopo che l'utente ha fornito i permessi di posizione.
Per localizzare l'utente sulla mappa viene utilizzato un `LocationListener` che viene aggiornato ogni due secondi e il cui ciclo di vita è legato a quello di `resume` e `stop` della Main Activity

2.2.1 Griglia

La griglia è implementata tramite oggetti della classe `Polygon` di Google Maps che vanno a formare i singoli riquadri. È generata dalla funzione `drawGridOnMap` della classe `Map`.
Per avere una griglia fissata alla mappa e non allo schermo in modo da avere la griglia indipendente dal movimento della mappa, si è considerata una coordinata canonica di partenza dalla quale si calcolano le posizioni di tutti i riquadri della griglia.
Quindi per calcolare la posizione di un riquadro in cui è compresa una determinata coordinata (`lat`, `lon`), viene calcolata la coordinata del vertice in alto a sinistra tramite la funzione `generateTopLeftPoint`, e da lì viene disegnato il resto del riquadro. Tale funzione genera il punto tramite le seguenti formule:

$$\texttt{lat}_{\texttt{tlPoint}} = \left\lceil \frac{\texttt{lat} - \texttt{lat}_{\texttt{start}}}{\texttt{offset}} \right\rceil \cdot \texttt{offset}$$

$$\texttt{lon}_{\texttt{tlPoint}} = \left\lfloor \frac{\texttt{lon} - \texttt{lon}_{\texttt{start}}}{\texttt{offset}} \right\rfloor \cdot \texttt{offset}$$

Dove $\texttt{lat}_{\texttt{tlPoint}}$ e $\texttt{lon}_{\texttt{tlPoint}}$ sono le coordinate del punto in alto a sinistra del riquadro che si sta calcolando. `Offset` è la lunghezza di un lato di un riquadro ed è calcolato dal metodo `metersToOffset` che prende in input la grandezza di un riquadro in base allo zoom (`meters`) e la converte in gradi usando la proporzione $1° \ \texttt{lat/lon} = 111km$ circa. $\texttt{Lat}_{\texttt{start}}$ e $\texttt{lon}_{\texttt{start}}$ sono le coordinate della posizione canonica presa come punto di riferimento, ovvero il punto $(0, 0)$.
Una volta definite le funzioni ausiliarie, per rappresentare la griglia viene preso in considerazione il punto in alto a sinistra dello schermo del dispositivo e viene generato il riquadro in cui è compreso. Successivamente viene calcolata l'area dello schermo del dispositivo e si generano riquadri adiacenti finché lo schermo riesce a visualizzarli, prima orizzontalmente poi verticalmente. Ogni riquadro viene aggiunto ad una lista dei poligoni in

[1] Permissions updates in Android 11 - Android Developers

modo da poter essere richiamato successivamente.
Per gestire la colorazione di un riquadro, viene utilizzata la funzione `getQuality`, che prendendo in input le medie dei tre tipi di misurazione in quel riquadro (limitando eventualmente il numero di misure da prendere in considerazione per la media in base al limite specificato dall'utente), li confronta con le soglie pessime e ottimali (di default o impostate dall'utente) e associa ad ogni tipo di misurazione il colore rosso se la media si trova sotto la soglia pessima, verde se si trova sopra la soglia ottimale e giallo se si trova in mezzo.

L'interazione con i riquadri è implementata tramite un listener della mappa, che ad ogni tocco restituisce la posizione in cui si è premuto e questa viene usata per ottenere il poligono relativo. Successivamente prende le misure in quel riquadro e fornisce le informazioni all'utente tramite dei dialog.
Inizialmente viene mostrato un dialog con informazioni generali sulle misure presenti nel riquadro. Premendo su "`Mostra di più`" viene creato un ulteriore dialog che mostra la lista di misure. Selezionando una misura si viene portati in un ultimo dialog che mostra nello specifico tutte le informazioni della misura selezionata con la possibilità di cancellarla. La cancellazione avviene invocando il `DAO` del database e fornendo l'id della misura alla query di cancellazione.

2.3 Notifica

La funzionalità di inviare una notifica se si entra in un riquadro senza misure recenti viene gestita nel metodo `listenerHandler` con parametro `isInFetchingMode` a `false`.
Per rilevare quando l'utente entra in un nuovo riquadro, viene confrontata la sua posizione presa dal `LocationListener` con l'ultima registrata controllando se le due posizioni si trovano in riquadri diversi. In caso affermativo viene chiamato il database per controllare se l'ultima misura in quel riquadro risale al giorno odierno e in caso negativo viene inviata una notifica per segnalare la mancanza.

2.4 Sensori

La gestione dei sensori avviene nella classe `Sensor` che implementa i metodi di misurazione per ciascuno dei tre componenti.
Le misurazioni LTE e Wi-Fi vengono effettuate instantaneamente, mentre per il calcolo del dB si campionano due misure del microfono nell'arco di tre secondi ignorando il primo perché non viene captato nulla. I valori registrati, essendo calcolati in ampiezza, vengono convertiti in dB con la funzione `fetchAmplitude`.

2.5 Metodi di misurazione

All'interno dell'applicazione è implementato un database locale con `Room` che immagazzina le misure registrate. Nel database una misura è memorizzata come oggetto della classe `Measure`, formata dai seguenti campi:

Campo	Descrizione
`id`	ID della misura, autogenerato e univoco
`timestamp`	Data e ora della misura
`lat`	Latitudine della posizione
`lon`	Longitudine della posizione
`lte`	Valore LTE calcolato
`wifi`	Valore Wi-Fi calcolato
`db`	Valore dB calcolato
`user_id`	UUID univoco per ogni utente
`imported`	True se la misura è importata, false se è stata effettuata dall'utente

La creazione di una misura avviene nella funzione `addMeasurement` che crea un oggetto `Measure` e lo popola con l'orario e la posizione attuale, le misure registrate tramite la classe `Sensor`, l'UUID univoco dell'utente e impostando il flag `imported` a `false`. Successivamente lo importa nel database. Infine, in base al flag `isOutside` passato come parametro, che indica se la misura è stata effettuata al di fuori dell'area dello schermo, aggiorna o no la griglia.

2.5.1 Periodic

La funzionalità di scansione periodica è gestita nel metodo `periodicFetch` che, finché è attivo lo switch `periodic_fetch` nelle impostazioni, effettua misure con un intervallo scelto dall'utente.

2.5.2 Periodic in background

La funzionalità viene gestita tramite il servizio `PeriodicFetchService`, la cui attivazione e disattivazione sono gestite dentro alla Main Activity in base allo switch `background_periodic_fetch` nelle impostazioni.
Il servizio, che permette all'applicazione di agire in background, invia una notifica fissa nella Status Bar che segnala che il servizio è in esecuzione e quante misure ha effettuato dall'avvio, successivamente effettua misure con un intervallo scelto dall'utente.
Durante il ciclo di vita del servizio, la mappa nel Main Activity viene oscurata.

2.5.3 Automatic

La funzionalità viene gestita nel metodo `listenerHandler` con parametro `isInFetchingMode` a `true`. Per rilevare quando l'utente entra in un nuovo riquadro, viene utilizzato lo stesso metodo utilizzato per la notifica (Capitolo 2.3).

2.6 Scambio di dati

Lo scambio delle misure viene gestito nell'activity `SwapActivity`.
Il compito principale dell'activity è manipolare oggetti JSON per importare ed esportare misure nel database locale.

2.6.1 Da file

L'importazione è gestita lanciando un intent `ACTION_OPEN_DOCUMENT` per permettere all'utente di selezionare il file. Dopo la selezione, vengono ricavate le misure dal file JSON e importate nel database tramite la funzione `importData`. Per evitare di importare più volte la stessa misura, viene prima controllato che non siano già presenti nel database misure con lo stesso ID utente, orario e posizione.
L'esportazione è gestita dalla funzione `exportData` che prende tutte le misure, escludendo quelle importate, dal database e le inserisce in un file JSON con estensione `.mapc`. Successivamente lancia un intent `ACTION_SEND` per permettere all'utente di condividere il file.

2.6.2 Bluetooth

Lo scambio dati con Bluetooth è implementato tramite un'architettura Client/Server dove chi esporta funge da Server e chi importa funge da Client.
Prima di tutto vengono controllati i permessi relativi al Bluetooth e, in caso non sia attivo, viene chiesto di attivarlo.
L'importazione è gestita a partire dalla funzione `btReceiveHandler` che inizia la scansione dei dispositivi nelle vicinanze che hanno l'esportazione attiva e, grazie ad un oggetto `BroadcastReceiver` che quando riceve l'intent `ACTION_FOUND` relativo a un dispositivo rilevato, lo mostra all'utente aggiungendolo ad una lista di dispositivi rilevati, che viene aggiornata ad ogni ciclo di scansione rimuovendo i dispositivi non più rilevabili. Quando l'utente preme su un dispositivo, viene inizializzata la connessione istanziando un oggetto di classe `ConnectThread` che ferma la scansione dei dispositivi e crea un socket per la connessione utilizzando un UUID per identificare l'app.
Una volta ricevute le misure, in una stringa formato JSON, vengono importate tramite la funzione `importData`.
L'esportazione è gestita a partire dalla funzione `btDiscoverHandler` che, tramite un intent `ACTION_REQUEST_DISCOVERABLE`, richiede all'utente di attivare la discoverabilità per una durata impostata. Se l'utente accetta viene inizializzato il server istanziando un oggetto della classe `AcceptThread`, che crea un server socket con lo stesso UUID del lato client per ricevere connessioni e rimane in ascolto finché un dispositivo si connette o scade il tempo. Una volta accettata la connessione con un client, si connette al suo socket e chiude il server socket creato precedentemente. Successivamente viene creata e trasmessa al client una stringa formato JSON che contiene tutte le misure, eccetto quelle importate, aggiungendo in coda una stringa `-- END --` per segnalare al client quando la trasmissione della stringa è terminata.
Mentre l'esportazione è in corso, viene mostrato un fragment che implementa un countdown sincronizzato con la durata della discoverabilità decisa nelle impostazioni e un bottone per terminare l'esportazione instantaneamente. Al termine del countdown o alla chiusura manuale dell'esportazione la discoverabilità viene disattivata e il fragment viene nascosto.

3 Bug noti

Di seguito vengono elencati, in ordine di gravità, i maggiori bug riscontrati durante lo sviluppo dell'applicazione, non risolti o aggirati.

1. In base al dispositivo usato, si sono notati comportamenti diversi dati dalla diversa gestione di Android e dell'ottimizzazione dei processi.

   (a) Su dispositivo fisico OnePlus, il programma va in crash all'avvio perché l'applicazione va direttamente alla callback `onRequestPermissionResult` prima che l'utente possa fornire o rifiutare i permessi. Per ovviare a questo problema bisogna concederli manualmente dalle impostazioni.

2. Su emulatore, durante la background periodic fetch, l'applicazione va in crash per timeout solo quando si rimane in MainActivity.

3. A volte è possibile che non venga visualizzata la richiesta di permessi per il motivo citato nel Capitolo 2.1. In tal caso è sufficiente riavviare l'applicazione per mostrare il dialog che rimanda alle impostazioni.

4. Le misure vengono effettuate con un fuso orario fisso (UTC) e non viene modificato se il dispositivo cambia manualmente il fuso orario.

5. Quando la mappa viene visualizzata con lo zoom minimo, la griglia non viene mostrata correttamente.