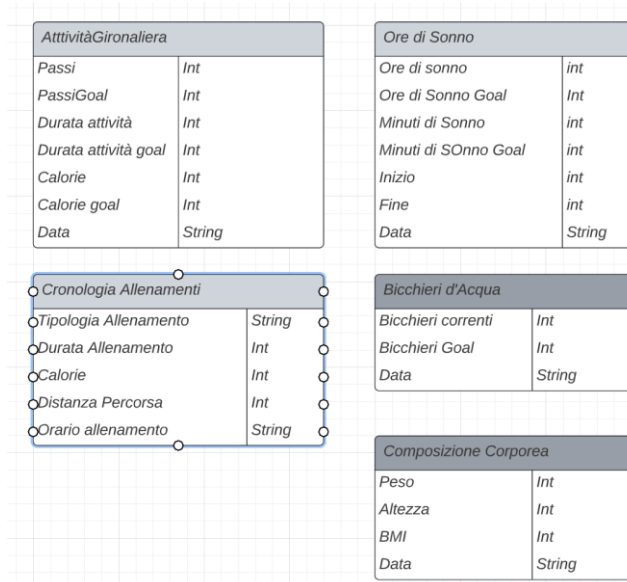


Relazione progetto di SAM

App “myHealth”

In questa relazione darò in modo molto veloce una ampia prospettiva delle funzionalità dell'applicazione.

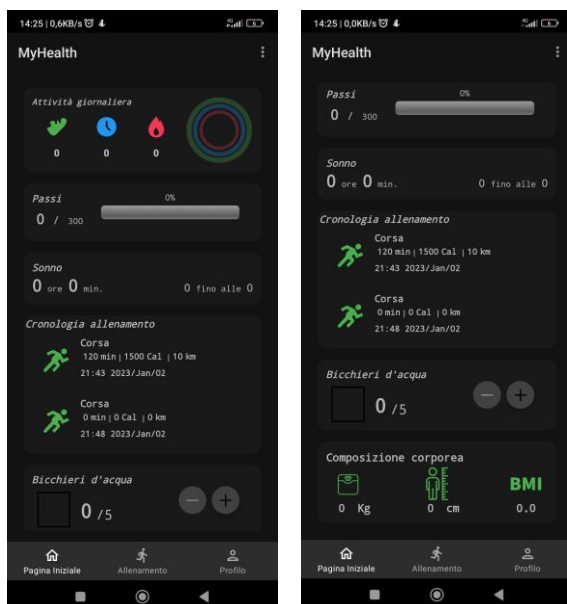
Schema di DB



Il DB utilizzato è molto semplice. Come struttura è stato utilizzato un SQLiteOpenHelper per semplificare il tutto, e le tabelle in figura sono tutte presenti all'interno della stessa classe "myDataBase" che estende SQLiteOpenHelper. All'interno della classe "myDataBase" sono presenti i metodi necessari per: aggiungere nuovi record nelle varie tabelle, recuperare particolari record in base ad un parametro dato, aggiornare record già esistenti. I record nelle varie tabelle sono organizzati in modo tale che ci sia sempre una sola riga per ogni data. Ed inoltre i record nelle tabelle sono aggiunti,

rimossi e aggiornati in modo tale che tutto sia sempre ordinato in base alla data, e l'ultima riga della tabella sia la data del giorno corrente.

Schermata principale



La schermata principale è formata da una Activity. Questa Activity contiene una "bottomNavigationView" che ci permette di spostarci tra i 3 fragment principali contenuti in una fragmentManager.

L'HomeFragement (Pagina iniziale, quella nelle immagini sopra) è formata principalmente da una RecyclerView, il cui numero di elementi è fisso, e non cambiano posizione (scelta

fatta per semplificare il tutto, ma non sarebbe troppo complicato implementare una cosa del genere).

L'app fornisce la funzionalità di conta passi, ed allo stesso tempo vengono generati artificialmente dati come durata attività e calorie consumate. Tutti i dati raccolti vengono poi mostrate nelle varie View mostrate dalla RecyclerView (in particolare quelle in posizione 0 e 1). Quando l'app è visibile all'utente allora i dati vengono raccolti direttamente dalla MainActivity che registra e deregistra il sensore opportuno quando necessario.

Tutti i dati raccolti vengono inseriti in un ViewModel utilizzato come banca Dati dell'applicazione. L'Home fragment registra degli observer sugli elementi di questo ViewModel (che sono MutableLiveData) di conseguenza ogni volta che cambiano i valori al loro interno, allora la RecyclerView viene avvisata e di conseguenza può ridisegnare le cose che sono cambiate.

Mentre quando l'app non è visibile all'utente allora entra in gioco un ForeGroundService che ha il compito di contare i passi effettuati quando l'app è in esecuzione ma non visibile, oppure quando è chiusa. Alla sua distruzione questo service invia un Intent in broadcast che verrà colto dalla MainActivity, che di conseguenza andrà ad inserire i dati raccolti nel ViewModel.

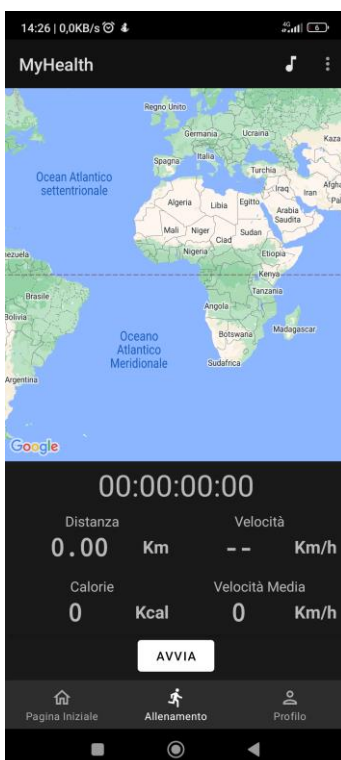
Sappiamo benissimo che i Service vengono eseguiti continuamente in background, ed in particolare il tipo di service che uso è sempre attivo in cerca di passi da registrare. Questo potrebbe causare l'utilizzo eccessivo di batteria, quindi nel caso in cui l'utente non vorrebbe usufruire di una feature del genere, allora ho anche dato la possibilità di disattivarla evitando. Disattivandola allora l'app sarà in grado di registrare i passi solo quando aperta e visibile all'utente.



Attraversi il menu è possibile impostare dei valori obiettivo (come si vede nella seconda immagine), implementato con un DialogFragment. Sempre con un DialogFragment si dà la possibilità all'utente di inserire dei dati nella View Composizione Corporea mostrando poi a schermo i risultati ottenuti.



Cliccando sulle varie View, ed in particolare sulla prima (quella in posizione 0 nella Pagina Iniziale) è possibile aprire una nuova Activity nella quale l'utente ha la possibilità di vedere un grafico a barre su 3 dati diversi (Passi, durata attività, Calorie bruciate) mostrate in base ai 7 giorni passati. Grazie al pulsante di condivisione in alto a destra nell'immagine l'utente ha la possibilità di condividere lo screenshot della schermata e farne quello che vuole (anche caricarselo su GoogleDrive).



Infine, il secondo fragment dà la possibilità all'utente di avviare un allenamento monitorandone durata, distanza percorsa, velocità, calorie bruciate e velocità media. Il tutto viene fatto tramite utilizzo delle funzionalità di GPS e alla API di google Maps. La funzionalità di tracciamento della posizione dell'utente è disponibile solamente se l'utente concede i permessi necessari per farlo.

Nel caso in cui i permessi vengano dati allora sarà di nuovo un `ForegroundService` a tenere traccia delle coordinate per percorso fatto, della durata dell'allenamento, della velocità e di varie altre informazioni.

Naturalmente il Service le genera solamente i dati necessari, sarà poi il `TrackingFragment` e raccoglieere questi dati (tramite delle observe su `MutableLiveData`) processarli e mostrarli correttamente a schermo.

Al termine dell'allenamento tutte le info a lue relative verranno salvate nella tabella corrispettiva nel DB.