UNIVERSITÀ DI PISA

Dipartimento di Informatica
Corso di Laurea in Informatica

Relazione progetto

StepsSound

Sviluppo Applicazioni Mobili

Prof. Vincenzo Gervasi

Carmine Vitiello
578070- Corso A

Anno Accademico 2021/2022

Obiettivo del progetto

L'applicazione ha lo scopo di migliorare le prestazioni sportive di un utente attraverso una gestione efficiente della riproduzione della musica che viene ascoltata durante una sessione sportiva, dal momento che secondo alcuni studi la musica aiuta l'uomo durante i vari stadi emotivi, allora è possibile condizionare le emozioni per migliorare i traguardi dell'individuo.

Dettagli implementativi generali

L'applicazione si forma di quattro principali activity, cioè:

- GoalActivity elenca attraverso un recyclerview i parametri delle sessioni sportive svolte in precenza che sono salvate un RoomDatabase;
- ListTracesActivity mette a disposizione un riproduttore audio per ascoltare le tracce che l'applicazione cattura attraverso il Content Resolver di sistema e poi salvate sul RoomDatabase, per una futura ed eventuale gestione delle playlist da implementare;
- RunningActivity viene invocata attraverso la MainActivity facendo pressione sul tasto di colore verde centrale all'inquadratura. Verranno mostrati i dati che sono tracciati durante la sessione sportiva, cioè
    - l'accelerometro che terrà traccia della velocità media dei movimenti dell'utente;
    - Il contatore dei passi;
    - Un cronometro per tenere traccia della durata.

  Sul fondo dello schermo ci saranno due pulsanti di cui uno verde per fermare momentaneamente la sessione sportiva o in caso riprenderla se fosse già in pausa. A destra c'è tasto rosso per terminare la sessione ed in caso che l'utente fosse soddisfatto dei traguardi raggiunti potrà salvare i dati nel suo storico che potrà ispezionare nella GoalActivity.

Infine l'ultima activity da citare è la SummaryActivity che mostrerà i dati della sessione sportiva in corso che sta per essere conclusa dall'utente, per poi chiedere se la sessione debba essere salvata o meno.

Queste sono le activity principali voglio concentrare l'attenzione sul MediaPlayer in modo tale da spiegare come è stato implementato e gestito.

Dettagli MediaPlayer

La riproduzione dell'audio è stata gestita attraverso al classe MediaPlayerService, come si può capire dal nome è un Service, visto che attraverso la documentazione e varie opinioni su internet è stato intuito che fosse una delle implementazioni migliori, perchè attraverso il service si evitava codice ridondante è si possono controllare alcune situazioni comuni, come la ricezione di chiamate oppure situazioni simili nelle quali il canale audio veniva occupato da processi prioritari. In questo componente ci sono delle funzionalità che verranno implementate in futuro, come:

- La gestione del cambio di transizione audio attraverso la tecnologia chiamata VolumeShaper, oppure attraverso TimerTask;
- Una gestione più accurata di notifiche attraverso il Media Controls oppure il Media Session in modo tale da far visualizzare la traccia riprodotta anche quando il dispositivo fosse bloccato.

Dettagli RoomDatabase e ViewModel

Per la gestione dei dati persistenti è stato utilizzato un RoomDatabase, cioè un componente della libreria Jetpack nativa di Android che è stata ideata per aiutare lo sviluppatore, perchè la maggiorparte delle volte potrebbe perderesi sul linguaggio SQL attraverso l'impiego di un database SQLLite, infatti, il database moderno aiuta lo sviluppatore a gestire i dati in maniera opportuna, visto che attraverso varie attenzioni si cerca di limitare errori come memory leak, che potrebbero essere scaturiti da errori di distrazione, oppure da sviluppi incoscienti, cioè azioni sconsiderate che possono sembrare adeguate durante una prima valutazione, ma in realtà portano ad errori di inconsistenza, incoerenza e riduzione delle prestazioni. Il RoomDatabase si forma di alcuni costrutti tra cui il **D**ata **A**ccess **O**bject (oppure comunemente chiamato **DAO** e l'entity che controllano principalmente l'interazione tra il Database e le varie chiamate per recuperare, modificare o inserire dati. Quando vengono utilizzate queste tecnologie è probabile che si accostino al ViewModel e al ViewModelProvider, che gestiscono la parte visualizzata dall'utente, cioè i dati che riescono ad essere mostrati sull'interfaccia.
Attraverso gli ultimi costrutti citati si evita errori causati dal caricamento continuo di dati durante i vari cicli di vita delle activity, visto che durante la morte di un'activity i dati al suo interno vengono distrutti. Al posto di utilizzare sistemi pesanti computazionalmente per le prestazioni del dispositivo oppure implementati manualmente, si cerca di utilizzare un provider che ottimizzi nel migliore dei modi i caricamenti delle informazioni rispettando tutte le fasi di un ciclo di vita di un'activity.

Librerie di terze parti

Il progetto sfrutta una libreria di terze parti chiamata Lottie che porta varie modifiche ai componenti simile alla libreria di design di google material, che viene utilizzata parzialmente nel progetto per visualizzare la bottom navbar. Lottie oltre ad aggiungere componenti statici offre anche componenti animati, infatti, nella MainActivity si trovano due animazioni che attraverso la loro configurazione è possibile cambiare velocità e modalità di riproduzione della sequenza animata. Per utilizzare questi componenti è necessario aggiungere una dipendeza al file Gradle del progetto e poi attraverso un file di formato Json, che rappresenta i dati essenziali per visualizzare l'animazione, è possibile mostrale il componente nella sua interezza.