

Relazione

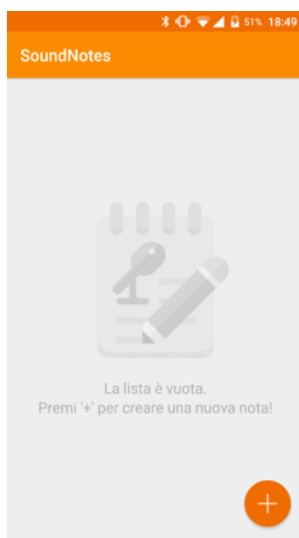
Introduzione

Soundnotes è un'app di note ed un registratore vocale. Il suo obiettivo è quello di rendere più semplice il riascolto delle registrazioni sfruttando il testo scritto nella nota.

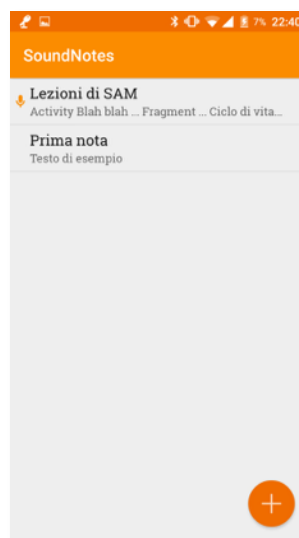
Spesso capita di fare lunghe registrazioni e poi dover perdere tempo a riascoltarle per trovare l'argomento che interessa. SoundNotes associa la registrazione ad un paragrafo di testo; basterà quindi creare registrazioni più brevi mentre si prendono appunti per sapere esattamente che cosa riguardano senza doverle riascoltare per la loro interezza.

Interfaccia grafica

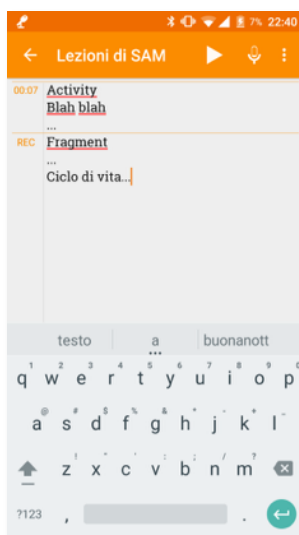
L'interfaccia è essenziale. È una semplice lista di note. Toccandone una si apre un campo di testo che riempie tutto lo spazio. Su tablet le due parti dell'interfaccia sono mostrate contemporaneamente. Le azioni si trovano tutte nell'ActionBar tranne il tasto per creare una nuova nota, un "+" fluttuante in stile Lollipop.



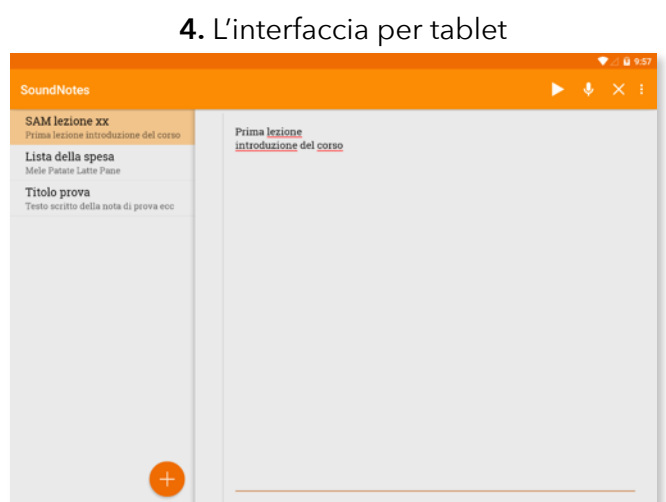
1.
La lista vuota



2.
La lista con alcune note



3.
Il dettaglio
di una nota

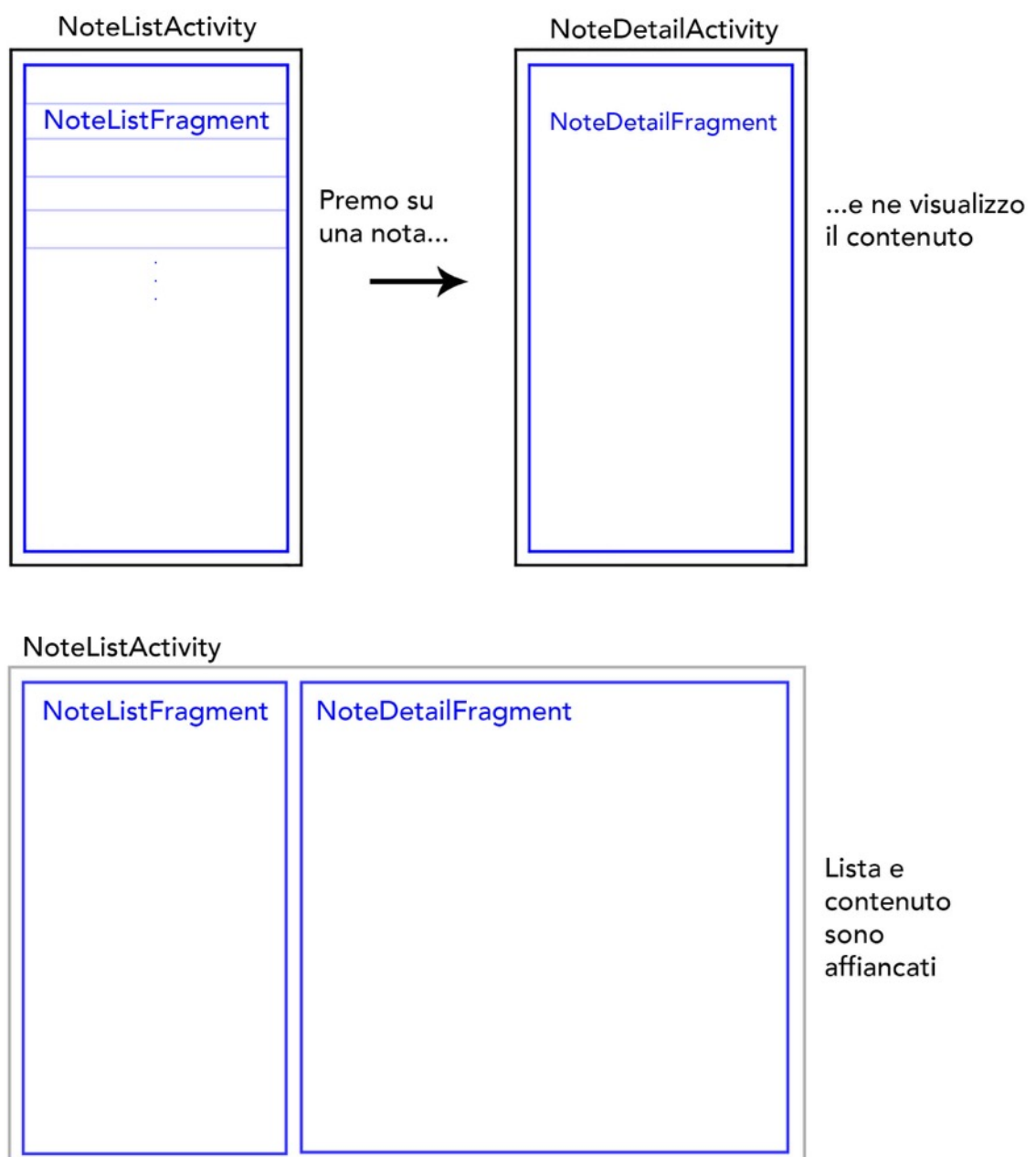


4. L'interfaccia per tablet

L'app è compatibile con smartphone e tablet, per questo motivo l'interfaccia è composta da due Activity e due Fragment. Questi ultimi si occupano di una buona parte della gestione dell'interfaccia del programma, rendendo il codice meno ridondante se si devono realizzare layout multipli, ma non più semplice da gestire. Infatti i Fragment hanno un loro ciclo di vita separato da quello dell'Activity ma strettamente legato da gestire con cura.

L'Activity principale del programma è `NoteListActivity`. È infatti presente su tutti i dispositivi: sia sugli smartphone che sui tablet. Su tablet gestisce anche il campo di testo della nota aperta rendendo `NoteDetailActivity` necessaria solo su smartphone. Su cellulare c'è infatti un passaggio tra un'Activity e l'altra (con relativa animazione) non presente su tablet.

Qui sotto è possibile vedere la relazione tra le Activity e i Fragment in SoundNotes:



Registrazione audio

1. Creazione di una registrazione

Per registrare basta premere il tasto a forma del microfono nell'ActionBar. Questa verrà associata alla riga in cui è posizionato il cursore e viene visualizzata con una linea arancione posta sopra il testo della riga corrente e da una piccola scritta nello spazio vuoto a sinistra: "REC" se si sta registrando o la sua durata al termine della registrazione.

In questo modo più registrazioni nella stessa nota spezzano visivamente il testo in parti che è come se appartenessero a registrazioni diverse.

2. Manipolazione del testo

Una volta create alcune registrazioni queste seguono il testo a mano a mano che l'utente scrive o cancella. Per evitare che vengano perse o si rovini la suddivisione tra le parti di testo che hanno associata una registrazione alcune operazioni sul testo non sono permesse, bloccando di fatto l'utente.

00:09	Nuovo
REC	Altro argomento

Per esempio (immagine a fianco) se il cursore è posizionato all'inizio di una riga in cui c'è una registrazione (|Altro argomento) e si

preme backspace non succede nulla perché unirei il testo delle due linee e perderei una delle due registrazioni. Un altro esempio lo abbiamo quando l'utente seleziona del testo e poi lo cancella o prova a tagliarlo; se il testo selezionato è in parte di una registrazione e in parte di un'altra lo devo bloccare.

Per implementare questo aspetto intercetto l'input dell'utente e lo controllo prima di confermarlo. In RichEditText (eredita da EditText), la classe che gestisce il campo di testo principale della nota, controllo quando l'utente taglia o incolla del testo. La maggior parte del lavoro lo faccio però in ReInputConnectionWrapper (eredita da InputConnectionWrapper): la collego alla RichEditText e molte operazioni che vengono fatte al testo mi vengono notificate prima di essere eseguite. Qui controllo la pressione di "backspace" di cui ho già parlato e l'aggiunta di ogni carattere che potrebbe aumentare il numero di linee perché inserisco un ritorno a capo o perché da qualche parte il testo sfora la lunghezza del display.

Un altro aspetto interessante è la "linea". Android conta le linee considerando il layout del dispositivo: spesso capita che il testo vada accapo per ché supererebbe la fine del display. Io non posso farci affidamento; considero il "ritorno a capo" come il carattere separatore e conto manualmente le linee. Devo comunque poter risalire alle linee dipendenti dal layout per sapere dove disegnare la linea di una registrazione e il tempo a sinistra.

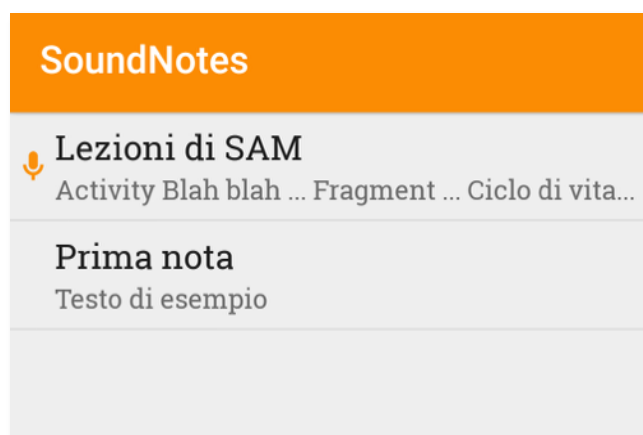
3. RecorderService

RecorderService è un Service deputato a gestire la registrazione dei suoni tramite un MediaRecorder e la loro riproduzione tramite un MediaPlayer. Mi serve un Service per continuare a registrare/riprodurre quando l'app va in background. Infatti il Service è un componente importante in Android di pari importanza all'Activity. Ha un ciclo di vita da gestire con cura per evitare che rimanga attivo e consumi risorse anche quando l'utente ha smesso di utilizzare l'app. Per questo motivo avvio il Service "NOT_STICKY" in modo che il sistema non lo riavvii dopo averlo terminato per bisogno di memoria. In quel caso, se viene chiamato il metodo onDestroy di RecorderService salvo la registrazione prima di terminare ma non posso esserne sicuro. Quello che faccio è dire al sistema che il Service è di tipo "foreground" garantendogli maggior priorità. Guadagno anche un modo semplice per avere una notifica sempre presente mentre sto registrando o riproducendo (Android non permette l'esecuzione di un Service "foreground" senza mostrare una notifica che dica all'utente che qualche componente dell'app è ancora in esecuzione).

Storage permanente

Il salvataggio delle note e delle registrazioni è organizzato interamente tramite file. Salvo tutto nella memoria interna dell'app. Ogni nota creata è associata ad un ID numerico crescente così da permettere la creazione di note con lo stesso nome e da identificarle facilmente. Mi aiuta anche nel salvataggio delle registrazioni, infatti vengono salvate in una cartella che contiene solo quelle, ma ne parlerò dopo.

Il nome della nota sul file system è composto dall'id, il titolo della nota e l'estensione classica dei file di testo, ".txt". Le registrazioni vengono salvate dentro una cartella che ha come nome l'id della nota aperta. Mentre sto registrando il MediaRecorder scrive nel file "temp.aac" che verrà poi rinominato nel nome corretto: posizione della stessa all'interno della nota, seguito dalla sua lunghezza in millisecondi e dall'estensione ".aac". A fianco vi sono alcuni esempi di come apparirebbero sul file system.



Durante la registrazione	Dopo
.../1-Prima nota.txt	.../1-Prima nota.txt
.../2-Lezioni di SAM.txt	.../2-Lezioni di SAM.txt
.../2/0-temp.aac	.../2/0-481516.aac