

DropAlert. Applicazione Android per il rilevamento delle cadute.

Corso di Sistemi Operativi 2

Prof. D. Riccio

Alessandro Serrapica N97000213

Luglio 2016

Indice

1	Introduzione	1
2	Descrizione dell'applicazione	2
2.1	La MainActivity	3
2.2	Il DropAlert Service	5
2.3	La Calibrazione	6
2.4	La AlarmActivity	7
3	Casi d'uso	8
4	Testing dell'applicazione	8

1 Introduzione

L'applicazione DropAlert é un'applicazione utile al rilevamento delle cadute dello smartphone o del suo possessore.

Il riconoscimento di una caduta avviene sfruttando l'accelerometro di cui oramai é dotata la maggior parte degli smartphone.

Un accelerometro é un dispositivo facente parte della famiglia dei MEMS (Micro ElectroMechanical Systems). Esso é composto da una parte interna libera di muoversi e da una parte esterna, fissata sulla mainboard dello smartphone, alla quale é collegata mediante delle molle. Il rilevamento dei movimenti é determinato dalla generazione di una carica elettrica (come un microscopico condensatore) tra la parte esterna ed interna che può essere quindi letta ed interpretata dal processore dello smartphone.

La misura dell'accelerazione avviene lungo i tre assi x , y e z .

Appoggiando lo smartphone su di un tavolo possiamo misurare l'accelerazione gravitazionale $9,8m/s^2$ lungo l'asse z . In realtà riferendoci in particolar modo agli accelerometri degli smartphone si può notare come tale misura non sempre è precisa e come alcuni smartphone tendano a valori più o meno discostanti dal valore atteso. Ciò è dovuto principalmente a tre motivi :

- la qualità dell' accelerometro;
- la posizione dell' accelerometro all'interno dello smartphone;
- l'implementazione software dei controller che gestiscono il sensore che spesso varia di costruttore in costruttore.

Risulta chiaro, quindi, come il proposito di sviluppare un'applicazione che basa il suo funzionamento sull'accelerometro e rivolta ad un numero considerevolmente elevato di smartphone possa scontrarsi con sensori poco precisi e dal comportamento non sempre predicibile.

2 Descrizione dell'applicazione

Il progetto assegnato prevedeva in origine di strutturare l'applicazione in due parti principali: un'activity con la possibilità di poter impostare le notifiche in caso di caduta e un service che si occupasse di rilevare le cadute e lanciare le relative notifiche.

In particolare, le notifiche assegnate sono :

- notifica sonora
- notifica via mail
- notifica via sms

A queste si è scelto di aggiungere diverse funzionalità volte a rendere l'applicazione più completa ed effettivamente utilizzabile.

Sono state quindi aggiunte le seguenti funzionalità:

- un registro per tenere traccia degli eventi implementato sfruttando un database SQLite.
- la possibilità di forzare la riproduzione dell'allarme sonoro anche se l'audio del telefono è disattivato.
- la possibilità di allegare alle email e agli sms la posizione geografica dello smartphone, sotto forma di un link a GMaps.
- la possibilità di avviare automaticamente il 'DropAlert Service' all'avvio di Android.
- la possibilità di posticipare l'invio di email e sms di 15 secondi per evitare falsi allarmi.
- la possibilità di calibrare il proprio smartphone per migliorare il funzionamento generale dell'applicazione.

Di seguito, si analizzeranno le principali componenti dell' applicazione e le relative funzionalità.

2.1 La MainActivity

La MainActivity costituisce l'activity principale dell'applicazione e in particolare quella mostrata all'avvio dell'applicazione.

L'activity é suddivisa in due zone principali: all'interno della prima é possibile gestire tutte le impostazioni dell'applicazione e le impostazioni di notifica; nella seconda, invece, si trova il registro degli eventi.

La persistenza dei dati nella prima parte, ovvero nelle impostazioni, é gestita tramite le 'SharedPreferences'. Esse consentono di accedere ad una zona di memoria dedicata all'applicazione, all'interno della quale é possibile salvare variabili semplici come interi e stringhe. Nel registro degli eventi, invece, soprattutto per fini didattici, si é scelto di implementare la persistenza dei dati tramite database 'SQLite'.

Nel progetto, quindi, é stata aggiunta una classe in grado di gestire completamente il database e una classe 'Registro' di supporto alle interrogazioni.

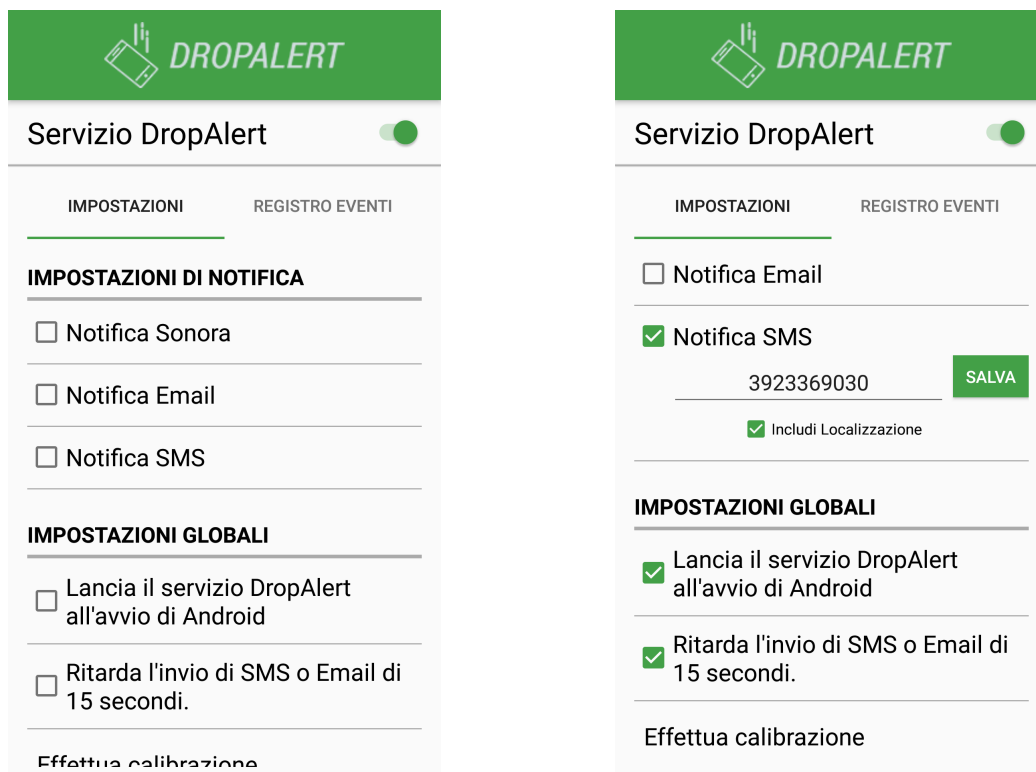
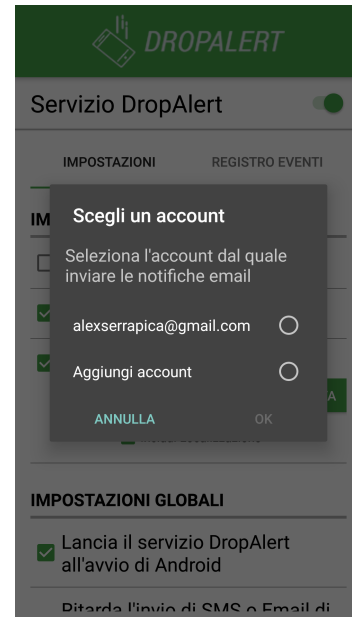


Figura 1: Screenshot della mainactivity

Per l'invio delle notifiche email si è scelto di utilizzare un account GMail residente sullo smartphone. Questa richiesta non è restrittiva dato che in ogni telefono Android è quasi obbligatorio, ormai, configurare un account. Una volta scelto l'account, l'applicazione richiederà il token di autorizzazione a Google tramite il quale sarà possibile accedere all'account GMail per inviare le mail. Il token verrà salvato nelle 'sharedpreferences' in modo da evitare di dover ripetere l'operazione più volte. Proprio per questo motivo è stata data la possibilità di cambiare account, e quindi di prelevare il relativo token d'accesso.



Inoltre in rispetto delle linee guida Android, dalla versione 6.0 in poi di Android (API 23) non è più sufficiente inserire alcuni permessi nel file 'AndroidManifest.xml' ma è necessario richiederli a runtime. Questo vale per quei permessi che Google ritiene 'pericolosi', tra cui l'invio degli SMS e l'accesso alla posizione del telefono.

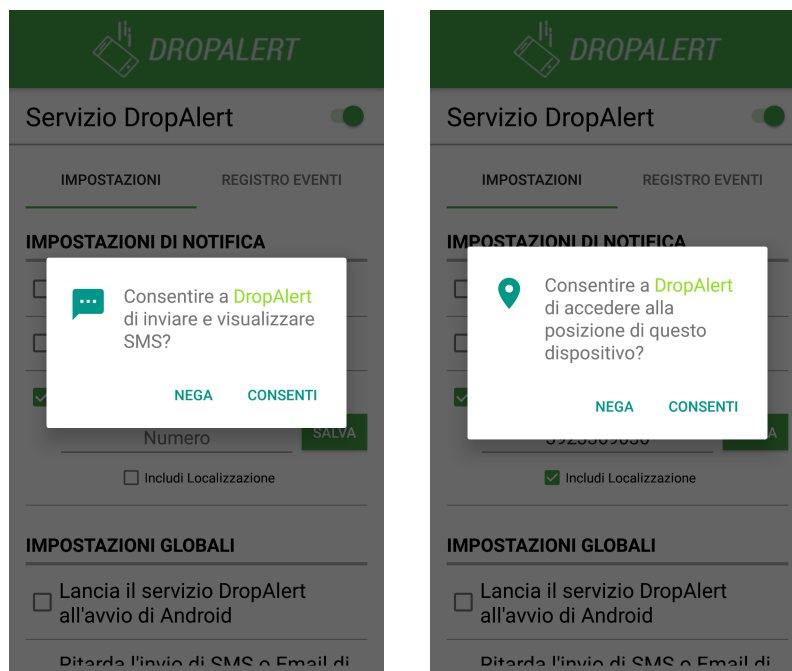
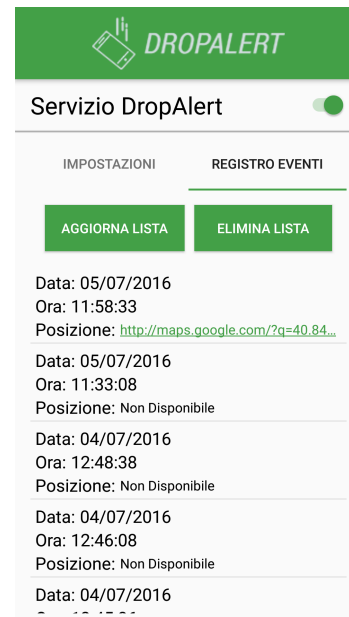


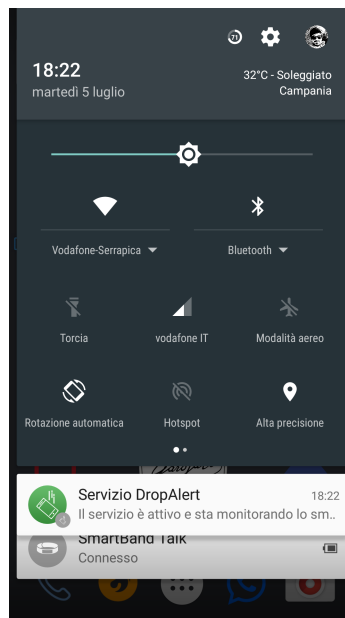
Figura 2: Screenshot delle richieste dei permessi

Il registro degli eventi, invece, é stato implementato, come anticipato, utilizzando una base di dati SQLite. Ogni riga della nostra tabella SQL sarà infatti mostrata come riga stilizzata di una ListView. Al fine di associare ogni riga ad una entry della ListView, dopo avere inserito il risultato della query in una mappa Java, é stato utilizzato l'oggetto 'SimpleAdapter' che va così a costruire la lista come in figura.



2.2 Il DropAlert Service

Il DropAlert Service può essere considerato come il cuore del progetto. Esso é incaricato di monitorare l'accelerometro e confrontare la differenza di due accelerazioni misurate in un intervallo di $60\mu s$ con un valore soglia. Tale valore é, in assenza di calibrazione, pari a $2g$. In particolare, il service é un 'foreground service', un service in grado di sopravvivere all'activity e alla relativa applicazione e che soltanto in casi estremi il sistema operativo Android decide di terminare. L'unico vincolo nella creazione di un 'foreground service' sta nel dover creare una notifica permanente all'interno dell'area delle notifiche del sistema.



La parte più importante del service è costituita dal controllo dell'accelerometro. La classe 'DropAlertService', infatti, oltre ad estendere, appunto, la classe Service, implementa l'interfaccia 'SensorEventListener' e sfrutta il metodo 'onSensorChanged' per monitorare i cambiamenti di stato dell'accelerometro.

```
@Override
public void onSensorChanged(SensorEvent event) {
    double x = event.values[0];
    double y = event.values[1];
    double z = event.values[2];
    lastAcc = acceleration;
    acceleration = Math.sqrt(x * x + y * y + z * z);
    double diff = acceleration - lastAcc;
    if (diff >= soglia) {
        if (!isAllarme()) {
            setAllarme(true);
            Intent dialogIntent = new Intent(this, AlarmActivity.class);
            dialogIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(dialogIntent);
        }
    }
}
```

2.3 La Calibrazione

Come anticipato, gli accelerometri presenti sugli smartphone differiscono in qualità e metodologia di implementazione. Queste differenze hanno costituito un iniziale impedimento nello sviluppo dell'applicazione. La soluzione migliore si è rilevata essere quella di offrire la possibilità di effettuare una calibrazione, in modo da poter consentire a ciascun utente di impostare una soglia personalizzata in base ai valori riportati dall'accelerometro. La calibrazione consiste in 3 simulazioni di caduta dello smartphone. Viene quindi calcolato un valore medio tra le 3 rilevazioni e ridotto di una tolleranza del 10%. Il valore calcolato viene quindi salvato nelle 'SharedPreferences' dell'applicazione e utilizzato come soglia d'allarme nel 'DropAlert Service'.

I risultati ottenuti, come si vedrà nel paragrafo dei Test, sono positivi e ciò è dovuto in particolar modo alla procedura di calibrazione.



Figura 3: Screenshot della procedura di calibrazione

2.4 La AlarmActivity

La 'AlarmActivity', é l'activity dedicata alla gestione dell'allarme sonoro e dell'invio delle notifiche mail e sms. Essa é richiamata direttamente dal 'DropAlert Service' e gestisce autonomamente anche la localizzazione dello smartphone mediante i 'Google Play Services'. L'activity, inoltre, si occupa di ritardare l'invio delle notifiche, qualora l'apposita checkbox sia stata selezionata, e gestisce eventuali errori come assenza di linea o dispositivo offline.

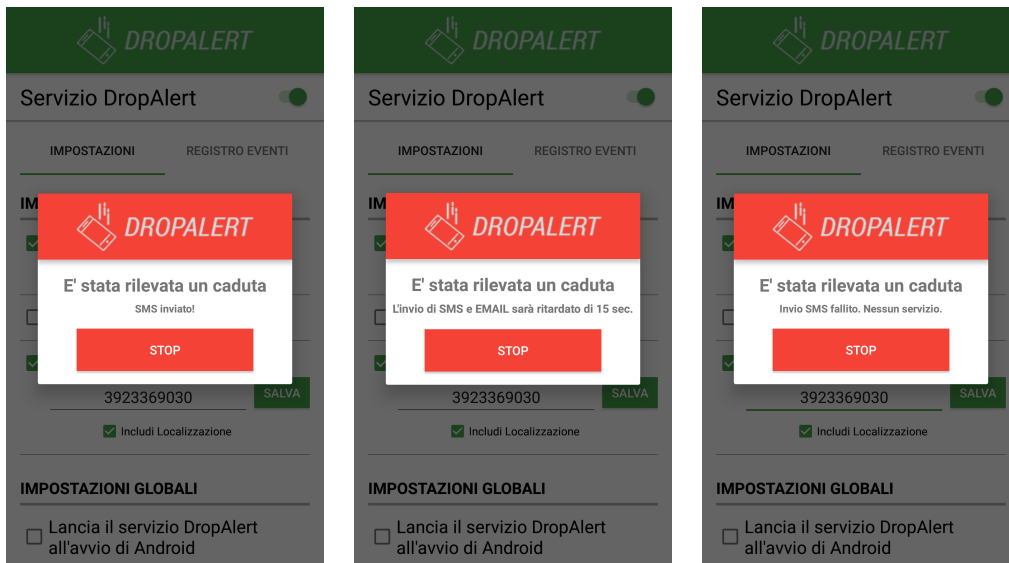


Figura 4: Screenshot della AlarmActivity

3 Casi d'uso

In questo paragrafo verrà riportato l'use case principale dell'applicazione DropAlert. Il diagramma mostra le macro-operazioni che svolgono utente e il 'DropAlert Service'.

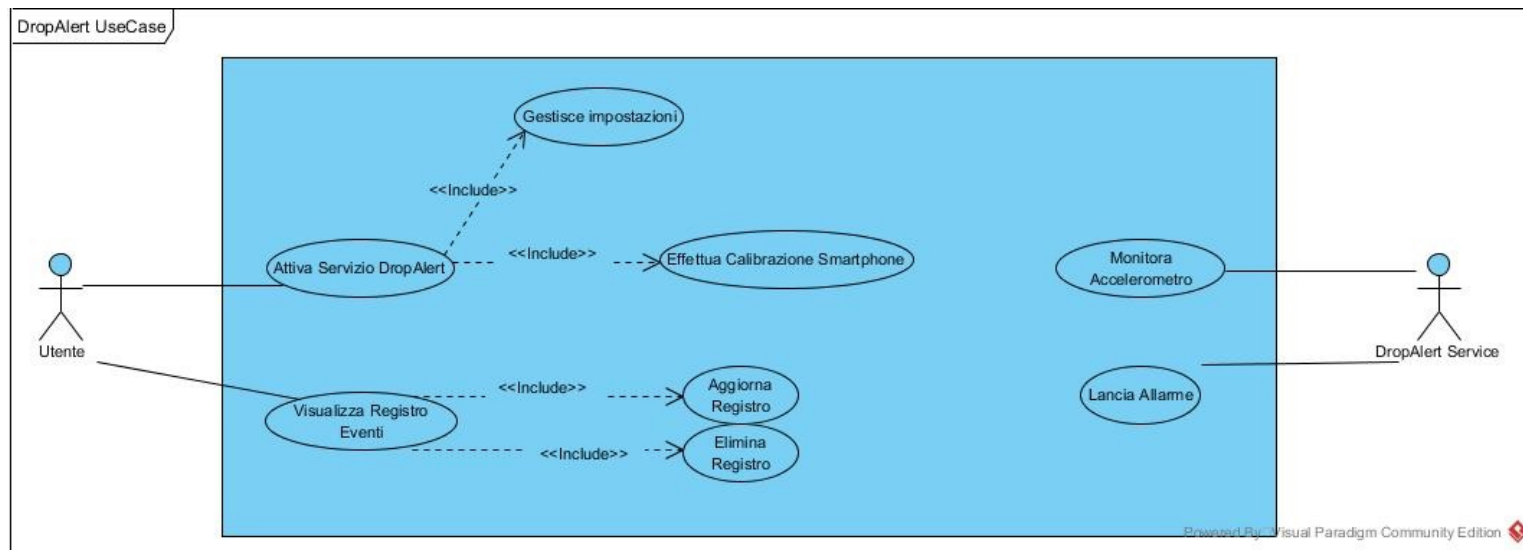


Figura 5: Use Case principale

4 Testing dell'applicazione

DropAlert è utilizzabile su dispositivi che supportino almeno le *API16(Android4.1.2)*. La scelta è stata effettuata al fine di garantire la compatibilità dell'applicazione su di un elevato numero di dispositivi: attualmente DropAlert è compatibile con più del 95% dei dispositivi Android. La versione target è invece *Android 6.0(API23)*.

I test effettuati sono stati svolti su numerosi smartphone che hanno costituito un campione di prova piuttosto eterogeneo. Infatti, è stato possibile testare l'applicazione su device di alta e bassa fascia dalle *API16* alle *API23*. Sebbene solo alcuni abbiano mostrato un comportamento simile nel gestire l'accelerometro e quasi nessuno riportava dati simili la funzione di calibrazione ha permesso di avere su tutti i dispositivi risultati positivi.

Le funzionalità basilari, quali l'invio di SMS e email, la riproduzione sonora dell'allarme e il registro degli eventi funzionano su tutti i dispositivi nel 100% dei casi. Tutti i layout grafici sono riprodotti ugualmente su tutti i device al netto della riproduzione grafica di quelle view come checkbox e switch che variano in base al tema basilare del dispositivo (come ad esempio Holo).

Per quanto riguarda la rilevazione della caduta dello smartphone è bene differenziare i test in due casi principali:

- se é stata effettuata la calibrazione seguendo le linee guida, la caduta viene rilevata nel 90% dei casi;
- se non é stata effettuata la calibrazione, la caduta viene rilevata nel 60% dei casi.

Come indicato nella schermata di calibrazione, essa consente di tener conto anche di cover e protezioni usualmente utilizzate per lo smartphone. Tenere conto di questo fattore consente una maggiore affidabilit , in quanto in base al materiale della cover gli urti potrebbero essere pi  o meno attutiti.

In conclusione i risultati ottenuti possono essere considerati soddisfacenti, in quanto l'applicazione nel complesso offre piena funzionalit  su tutti i dispositivi che si é scelto di supportare.