

Documentazione progetto LAM2017 : BudgeTracker

Federico Peconi

8 settembre 2017

Oggetto

Il seguente documento tratta, in maniera sintetica ma senza rinunciare alla completezza, del processo di scelta, progettazione e sviluppo del progetto finale per il corso di Applicazioni Mobili 2016/2017 tenuto dai professori Luciano Bononi e Luca Bedogni all'università Alma Mater Studiorum di Bologna.

L'applicazione in questione implementa un servizio di tracciabilità delle spese (budget tracking) seguendo e rispettando le funzionalità richieste nella proposta di progetto relativa.

Scopo dell'applicazione

Lo scopo è quello di fornire all'utente un'applicazione che permetta di tenere traccia del proprio budget.

Tale intento si articola fornendo le seguenti funzionalità:

- La scelta di un periodo di monitoraggio: settimanale o mensile ;
- l'identificazione ed il salvataggio su più livelli di 3 tipi di spese fondamentali: Spesa Normale, Spesa Pianificata, Spesa Periodica ;
- la visualizzazione ordinata delle spese effettuate e la possibilità di cancellarle ;
- la geolocalizzazione dei movimenti e dei luoghi in cui le spese sono state effettuate o pianificate ;
- il calcolo di statistiche utili come il tasso di spesa per categoria o l'andamento del budget in base al periodo scelto ;
- la possibilità di salvare localmente un file di report.

La parte principale di questo documento sviluppa queste caratteristiche una per una cercando di sottolineare i problemi incontrati, le soluzioni trovate e le possibili aggiunte.

Ambiente di sviluppo e requisiti

Il software è stato sviluppato su di una macchina che monta come sistema operativo UbuntuOS 17.04 attraverso l'utilizzo dell'IDE Android Studio 2.3. I requisiti non funzionali necessari per il corretto utilizzo sono device con versione minima SDK19.

Features

Il periodo

La politica dell'applicazione richiede che l'utente inserisca come prima cosa i dati relativi al periodo in cui vuole tracciare le sue attività ed il relativo budget a disposizione.

Il periodo può essere o mensile o settimanale, la scelta è mutualmente esclusiva e permette poi un'organizzazione più lineare delle attività di gestione nonché un'esperienza lato utente semplice ed intuitiva.

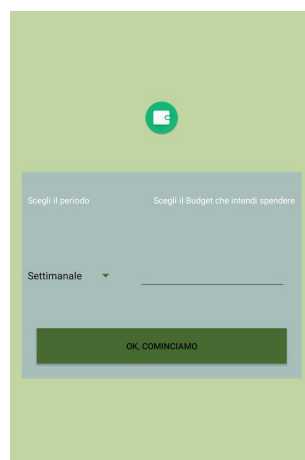
In particolare, tale scelta andrà ad incidere sul modo in cui vengono calcolate le statistiche e la frequenza con cui il sistema rinnova il budget.

Questo design impone di risolvere i seguenti problemi:

1. Come tenere traccia della fine/inizio del periodo;
2. Come salvare lo stato del budget a fine/inizio di ogni periodo.

Per quello che concerne entrambi i punti BudgeTracker implementa il service AlarmPush il quale si occupa di schedare un allarme, sfruttando il servizio di sistema AlarmManager, a fine periodo. Questo allarme verrà poi gestito dal reciver AlarmHandler che a sua volta si occuperà di salvare lo stato di fine periodo nella tabella apposita presente nel database e di resettare il budget al valore di default.

E' da sottolineare che in AlarmHandler non vi è la necessità di resettare l'allarme in quanto questa operazione viene fatta da AlarmPush essendo questo risvegliato da un altro broadcast reciver: Reciver, ad ogni boot (che si suppone avvenire con una frequenza giornaliera).



I tipi di Spesa, inserimento

Come da specifica, nel progetto vengono distinte 3 tipologie di spesa:

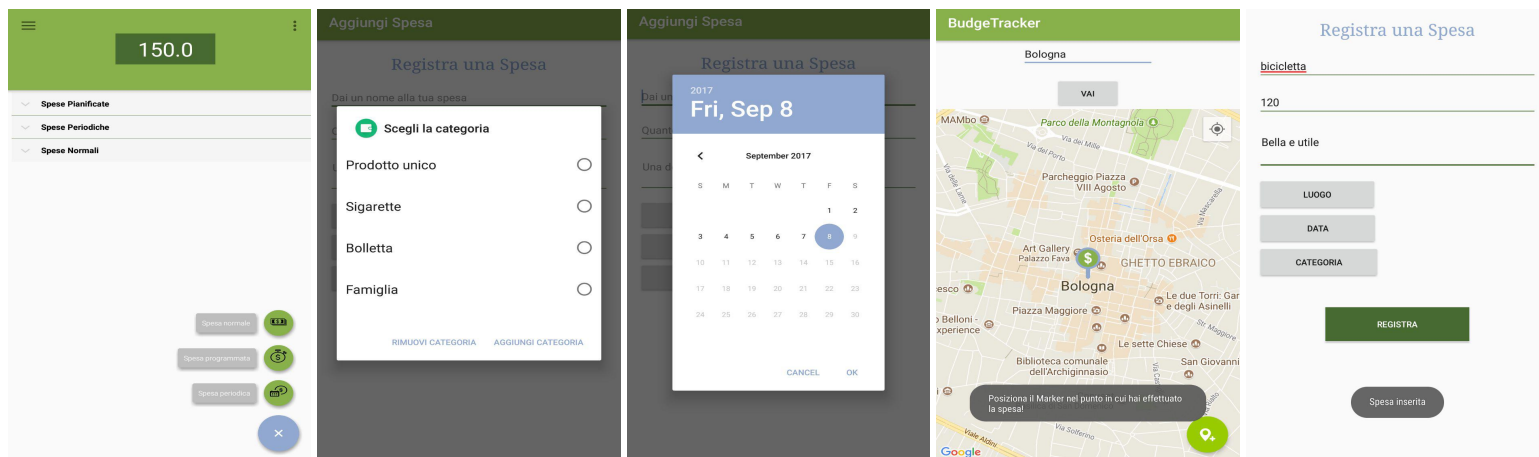
1. Normale: vengono intese "normali" tutte quelle spese di natura quotidiana che l'utente effettua e che non vengono pianificate ;
2. Pianificate: sono quelle spese normali per cui vi è un interesse a programmarle nel futuro.
3. Periodiche: sono banalmente tutte quelle spese che si ripetono periodicamente. Anche in questo caso è possibile scegliere tra periodo mensile e periodo settimanale.

La logica dell'applicazione verte intorno a questi 3 oggetti.

Sono tutte figlie della classe expense che le accomuna per gli attributi: nome, costo, categoria, data e descrizione; i primi 4 campi sono obbligatori durante l'inserzione.

La possibilità di inserire il luogo in cui sono state o saranno effettuate è presente solo per le spese pianificate e spese normali. La data, invece, diventa un giorno settimanale o mensile per le periodiche.

Il caso d'uso base è proprio quello di registrare queste spese che saranno salvate nelle relative tabelle all'interno del database.



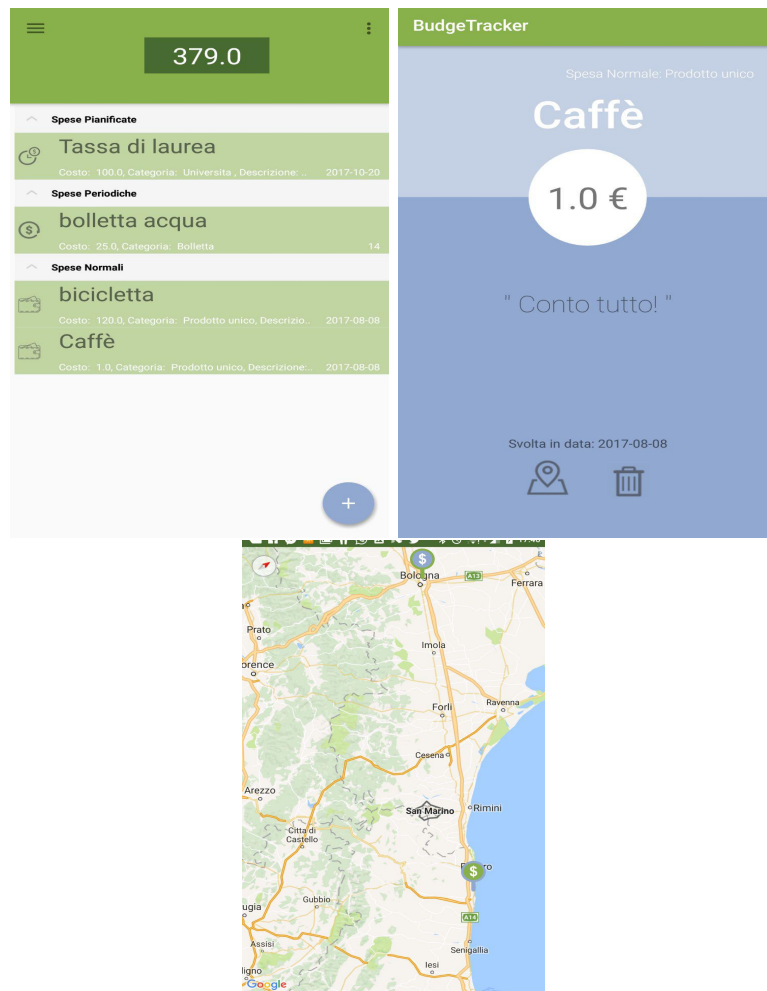
Visualizzare le spese

ExpandableListView e Geolocalizzazione

Le spese registrate sono caricate dinamicamente attraverso l'utilizzo combinato di una LinkedHashMap e di un Adapter su di un elemento ExpandableListView che consente di visualizzarle divise per tipo.

L'ordine segue la data d'inserimento e cliccando su queste è possibile accedere all' ExpenseActivity dedicata, per una visione a 360 gradi di quello che è l' oggetto spesa.

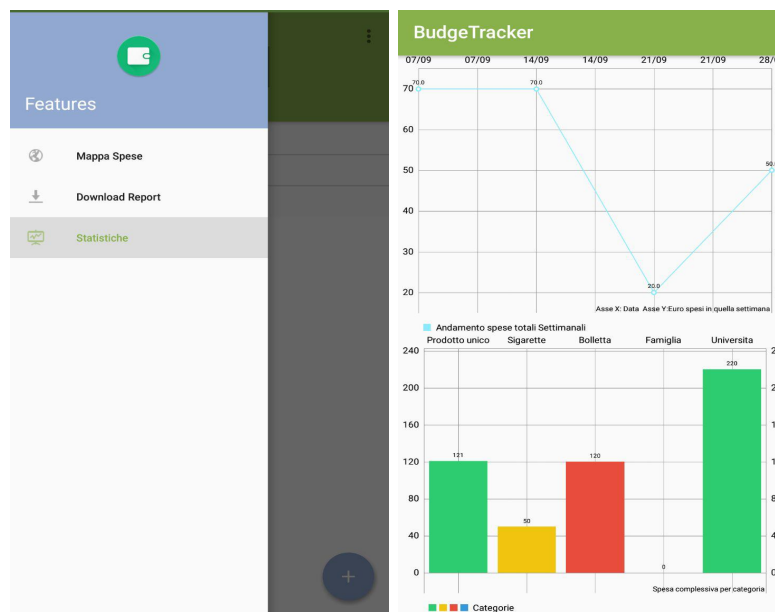
Da qui è possibile inoltre cancellarla o visualizzarla geograficamente in maniera generale i.e. immersa nella mappa tra le altre spese.



Statistiche

Una feature che contribuisce a rinforzare lo scopo dell'app oltre che ad arricchirla graficamente è l'introduzione di statistiche espresse tramite grafici. Questi sono di due tipi: lineChart e barChart e sono oggetti nativi della libreria MPchart che vengono utilizzati per rappresentare, rispettivamente, l'andamento del budget nel susseguirsi dei periodi e l'ammontare delle spese suddiviso per categorie.

I dati vengono richiamati dal database.



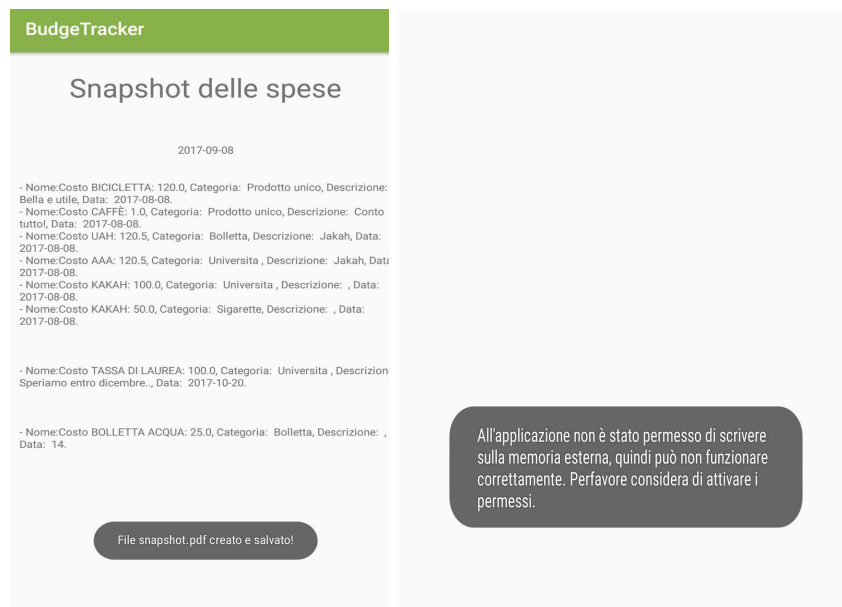
E.g. : vediamo che l'utente si è reso conto di essersi lasciato andare un po' troppo rispetto ai suoi standard.

PDF report

Sempre nel NavigationDrawer è possibile accedere all'activity che permette di salvare uno snapshot dei dati utili e di salvarli sotto forma di PDF in locale.

Tale funzionalità è stata implementata attraverso l'uso dell'oggetto pdfDocument nativo in Android API > 18.

Come per la geolocalizzazione, i permessi per la scrittura vengono richiesti anche a run-time seguendo la politica ANDROID > 5.



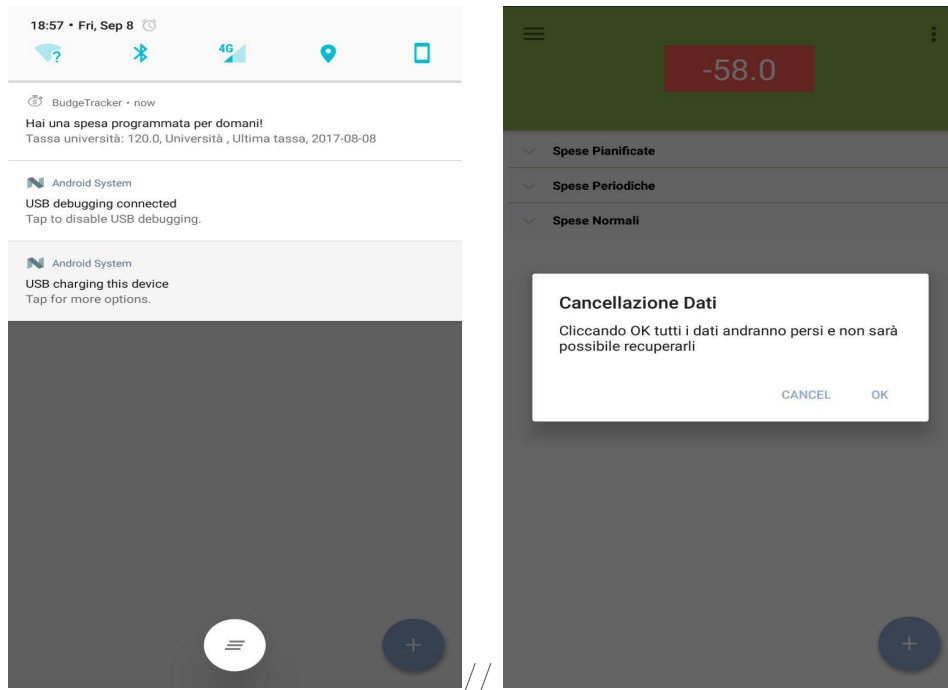
Altre funzionalità e scelte implementative

Per quanto minimale, la UI è curata dal punto di vista cromatico scegliendo come colori i Pantoni Greenery e Serenity, di cui il primo è stato eletto colore dell'anno. Inoltre, nei limiti delle risorse, la grafica si ispira alle buone norme di material design.

Sia per le spese programmate che per quelle periodiche vengono schedulati allarmi impostati a un giorno prima della ricorrenza seguendo la stessa logica adottata per schedulare la fine di un periodo.

Tutte le quantità sono gestite con una precisione di tipo Double.

L'utente, nel caso del bisogno, ha inoltre la possibilità di cancellare tutti i dati fino ad ora accumulati e ricominciare da 0 il percorso ripartendo dalla schermata di benvenuto.



Conclusioni e possibili aggiunte

Possibili aggiunte dal punto di vista dei requisiti funzionali:

- Arricchimento delle informazioni correlate alle spese: foto, video, condivisione sui social..;
- Integrazione del sistema con carta di credito per rendere l'applicazione più concreta;
- Raffinare quelle che sono le tipologie delle spese.

Inoltre é auspicabile implementare test "di bordo" per garantire una maggiore affidabilità, aggiungere controlli di sicurezza e.g. sql injection, e tanto altro ancora.

L'applicazione presentata soddisfa le funzionalità ritenute obbligatorie dalla

specifica e fornisce un ambiente effettivamente utilizzabile per il monitoraggio delle spese riuscendo così a realizzare per intero il suo obiettivo.