

Laboratorio di Applicazioni Mobili: Budget Tracker

Federico Peconi

Docenti: Luciano Bononi Luca Bedogni

Corso di laurea in Informatica, Dipartimento di Informatica, Scienza e Ingegneria

15/09/2017

Introduzione

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

Oggetto

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

Ambiente di sviluppo e requisiti non funzionali

Il software è stato sviluppato utilizzando l'IDE Android Studio 2.3.
Il target supportato per il funzionamento corretto dell'applicazione sono tutti i device Android con versione API19(KitKat) o superiore.



Features

Al fine di garantire il servizio proposto, sono state individuate le seguenti **caratteristiche** da fornire:

- ▶ 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;
- ▶ la possibilità di salvare localmente un file di report.

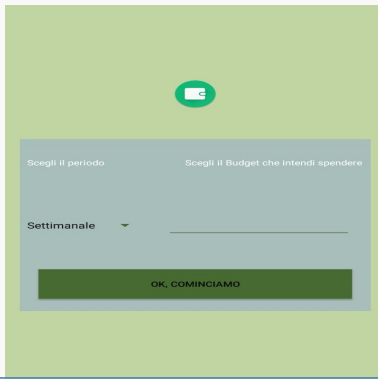
Il Periodo (1)

Come prima cosa, all'utente viene richiesto di scegliere con quale **frequenza** desidera monitorare le sue risorse.

Il periodo può essere o mensile o settimanale.

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.



Il Periodo (2)

Soluzione trovata

BudgetTracker implementa il **service AlarmPush** il quale si occupa di schedulare un allarme, sfruttando il servizio di sistema AlarmManager, al termine di ogni periodo.

Il triggering del segnale comporta il reset del budget corrente al valore di default.

I tipi di Spesa

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. **Pianificata**: sono quelle spese normali per cui vi è un interesse a programmarle nel futuro.
3. **Periodica**: 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.java`** che le accomuna per gli attributi: nome, costo, categoria, data e descrizione.

Caso d'uso : inserimento

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

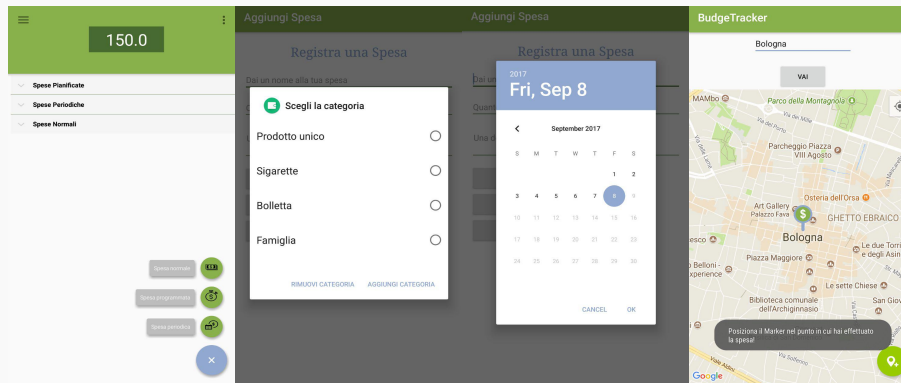


Figure: Varie fasi del salvataggio di una spesa

Visualizzazione (1)

Il sistema fornisce diverse modalità di visualizzazione delle spese inserite.

- ▶ Nella schermata di home queste vengono raggruppate all'interno di un **ExpandableListView** che le mostra in maniera ordinata e divise per tipo.
- ▶ Cliccando su queste é possibile accedere all'**ExpenseActivity** dedicata, per una visione a 360 gradi di quello che é l'oggetto spesa.
Da qui é possibile inoltre cancellarla o visualizzarla geograficamente in maniera generale i.e. immersa nella mappa tra le altre spese.

Visualizzazione (2)

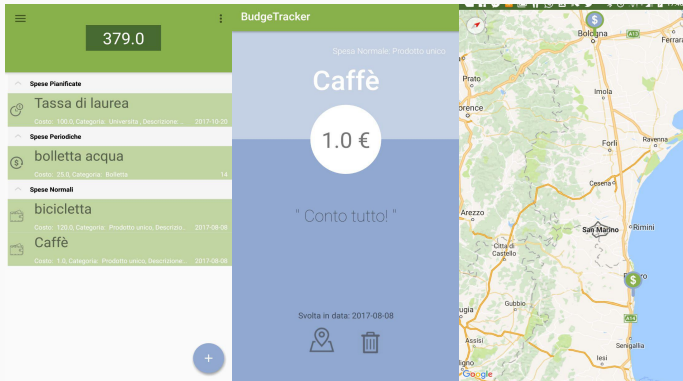
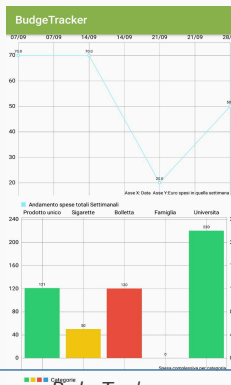


Figure: I vari livelli di visualizzazione

Statistiche

Una sezione raggiungibile dal drawer permette la rappresentazione grafica di statistiche.

- ▶ Il calcolo dei grafici vien fatto utilizzando la libreria **AndroidMPChart**
- ▶ I grafici mostrano l'andamento del Budget nel susseguirsi dei periodi e l'ammontare delle spese fatte suddivise per categorie



PDF Report

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

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

BudgeTracker

Snapshot delle spese

2017-09-08

- Nome:Costo BICICLETTA: 120.0, Categoria: Prodotto unico, Descrizione: Bella e utile, Data: 2017-08-08.
- Nome:Costo CAFFE: 1.0, Categoria: Prodotto unico, Descrizione: Conto tutto!, Data: 2017-08-08.
- Nome:Costo UAH: 120.5, Categoria: Bolletta, Descrizione: Jakah, Data: 2017-08-08.
- Nome:Costo AAA: 120.5, Categoria: Università , Descrizione: Jakah, Data: 2017-08-08.
- Nome:Costo KAKAH: 100.0, Categoria: Università , Descrizione: , Data: 2017-08-08.
- Nome:Costo KAKAH: 50.0, Categoria: Sigaretta, Descrizione: , Data: 2017-08-08.

- Nome:Costo TASSA DI LAUREA: 100.0, Categoria: Università , Description Speriamo entro dicembre , Data: 2017-10-20.

- Nome:Costo BOLLETTA ACQUA: 25.0, Categoria: Bolletta, Descrizione: , Data: 14.

File snapshot.pdf creato e salvato!

Scelte implementative

- ▶ Per quanto minimale, la UI é curata dal punto di vista cromatico scegliendo come colori i Pantoni **Greenery** (colore dell'anno) e **Serenity** .Inoltre, nei limiti delle risorse, la grafica si isipira 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 schedualare 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.

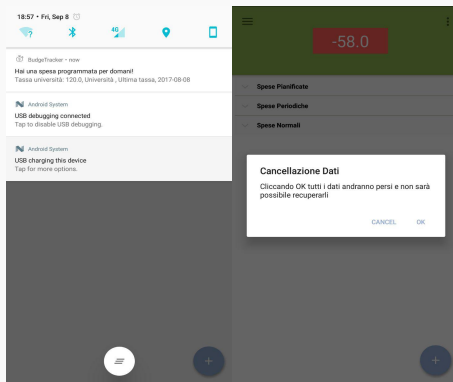


Figure: Notifiche e Earasing dei dati, si noti come nello sfondo il colore del contatore cambia quando il budget corrente va in "rosso".

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 piú concreta;
- ▶ Raffinare quelle che sono le tipologie delle spese.

E' auspicabile implementare **test "di bordo"** per garantire una maggiore affidabilit , aggiungere **controlli di sicurezza** e.g. sql injection, e, ovviamente, altro ancora.

Conclusione

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.