

Laboratorio di Applicazioni Mobili 2013/2014
Relazione finale relativa al progetto

Budget Tracking

Nome e cognome: Giulia Franchini

Matricola: 0000626125

Email: giulia.franchini2@studio.unibo.it

Link al progetto: <https://github.com/franka92/BudgetTracking>

Scopo dell'applicazione

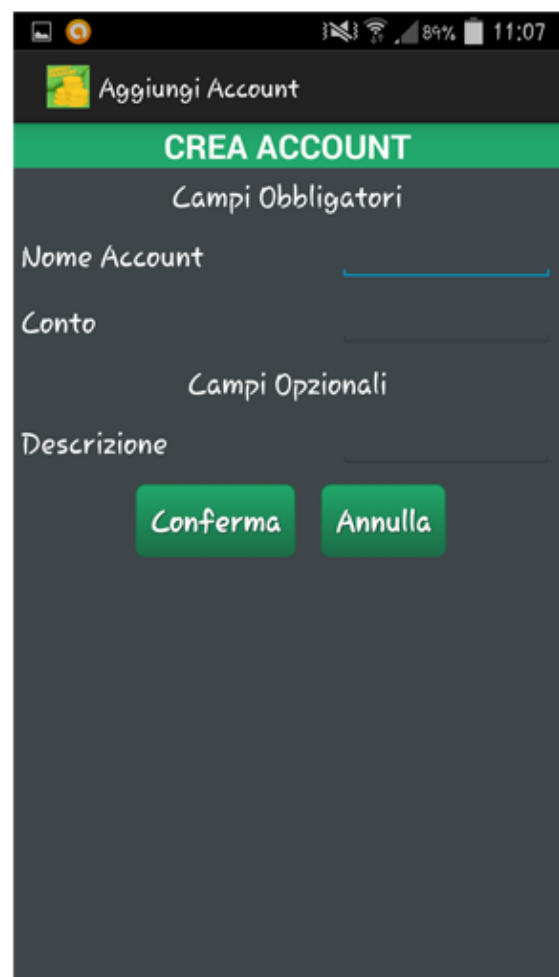
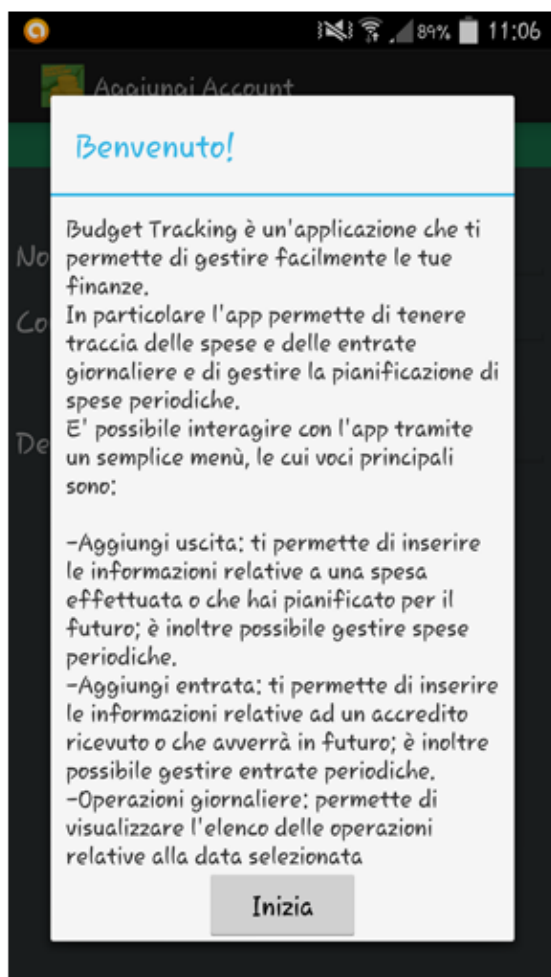
Budget Tracking è un'applicazione che permette all'utente di gestire facilmente le proprie finanze. In particolare l'app permette di tenere traccia delle spese effettuate e delle entrate giornaliere, e di gestire la pianificazione di operazioni periodiche o future.

Caratteristiche e requisiti

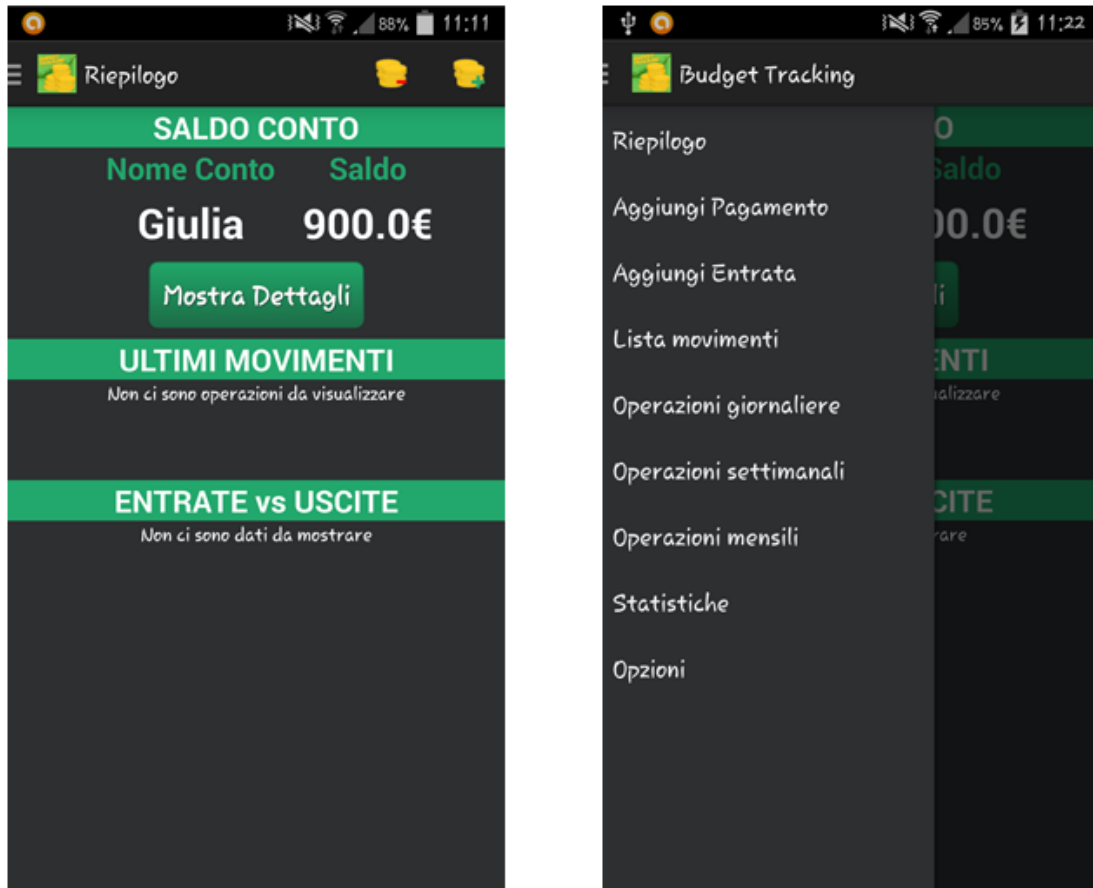
L'ambiente di sviluppo utilizzato per la realizzazione del progetto è stato Android Studio (Beta) 0.8.2 per Windows. L'applicazione è stata testata e sviluppata per funzionare con versioni di Android ≥ 4.3 (Jelly Bean).

Principali funzionalità e caratteristiche

Al primo avvio dell'app viene richiesto all'utente di inserire alcune informazioni di base per poter iniziare ad utilizzare l'applicazione.



Una volta configurato il proprio account verrà mostrata una schermata di riepilogo, che rappresenta la vista principale che viene visualizzata ad ogni avvio dell'app.



È possibile interagire con l'applicazione tramite un semplice menù laterale (Navigation Drawer), le cui voci principali sono le seguenti:

1. Aggiungi Pagamento

Permette all'utente di inserire le informazioni relative ad una spesa effettuata o pianificata per il futuro. In particolare è possibile inserire le informazioni relative a:

- Oggetto (nome)
- Importo (in euro)
- Categoria
- Data in cui è stato effettuato il pagamento
- Mezzo di transazione utilizzato
- *Luogo (opzionale)*
- *Breve descrizione (opzionale)*
- *Foto dell'oggetto acquistato (opzionale)*

Aggiungi Operazione

AGGIUNGI PAGAMENTO

Oggetto: _____

Importo (€) : _____

Categoria: abbigliamento

Data: 2014-12-13

Mezzo di transazione: contanti

Luogo:

☒ Posizione corrente ☐ Inserisci luogo

Descrizione:

☐ Ripetizione

Aggiungi Foto

Conferma Azzerà

Aggiungi Operazione

AGGIUNGI PAGAMENTO

Oggetto: _____

Importo (€) : _____

Categoria: abbigliamento

Data: 2014-12-13

Mezzo di transazione: contanti

Luogo:

☒ Posizione corrente ☐ Inserisci luogo

Descrizione:

50 pezzi. dvd-R

☐ Ripetizione

Aggiungi Foto

Conferma Azzerà

Per l'inserimento del luogo è possibile scegliere tra due opzioni, ovvero rilevare automaticamente la posizione corrente o ricercare manualmente un indirizzo.

Aggiungi Operazione

AGGIUNGI PAGAMENTO

Oggetto: _____

Importo (€) : _____

Categoria: abbigliamento

Data: 2014-12-13

Mezzo di transazione: _____

CERCA LUOGO

via innerio

Seleziona un luogo (cliccandoci sopra)

- Via Innerio, Bologna, Italy
- Via Innerio, 25127 Brescia BS, Italy
- Via Innerio, 43123 Parma PR, Italy

☐ Ripetizione

Aggiungi Foto

Conferma Azzerà

Aggiungi Operazione

AGGIUNGI PAGAMENTO

Oggetto: _____

Importo (€) : _____

Categoria: abbigliamento

Data: _____

Mezzo di transazione: _____

CERCA LUOGO

via innerio

Seleziona un luogo (cliccandoci sopra)

- Via Innerio, Bologna, Italy
- Via Innerio, 25127 Brescia BS, Italy
- Via Innerio, 43123 Parma PR, Italy

☐ Ripetizione

Aggiungi Foto

Conferma Azzerà

È possibile inoltre specificare se l'operazione che si sta inserendo è periodica e selezionare il tipo di ripetizione previsto tra giornaliera, settimanale, mensile

Premendo su "Conferma" il pagamento viene salvato sul database e, se la data inserita è minore o uguale alla data odierna, viene effettuato anche un aggiornamento del saldo attuale.

Premendo su "Azzerà" vengono puliti tutti i campi di inserimento ed eliminata anche l'eventuale foto inserita.

2. Aggiungi Entrata

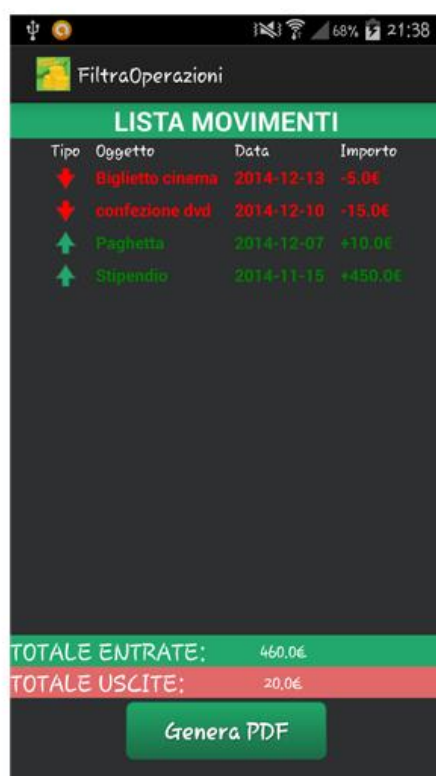
Permette all'utente di inserire le informazioni relative ad un accredito ricevuto o che avverrà in futuro. Il principio di funzionamento è analogo a quello sopradescritto per l'aggiunta di un pagamento.

3. Lista Movimenti

Permette all'utente di visualizzare un elenco completo di tutte le operazioni inserite, senza filtri di alcun tipo. Per ciascuna operazione vengono mostrate le informazioni relative a:

- Tipo di operazione (Pagamento o Entrata)
- Oggetto
- Importo
- Data

Cliccando su una delle voci dell'elenco si apre una Dialog con la lista completa delle informazioni associate all'operazione salvata. Se sono presenti informazioni relative al luogo in cui l'operazione è stata effettuata, è possibile visualizzare la posizione su Google Maps (tramite Web o la relativa app, se installata sul dispositivo).



Cliccando su "Elimina" è possibile cancellare dal DataBase tutti i dati relativi all'operazione ed eventuali immagini ad essa associata; dopo ogni cancellazione viene aggiornato il saldo attuale del conto.

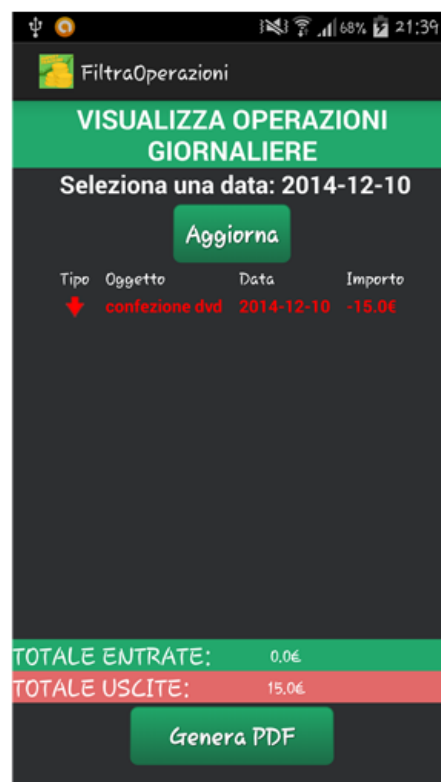
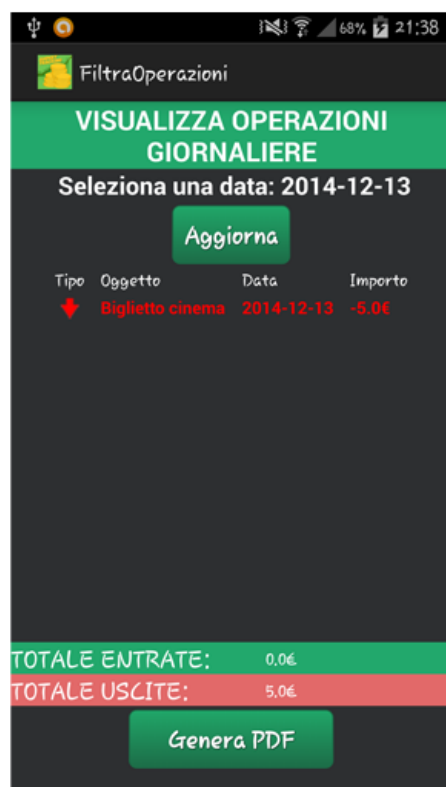
Cliccando sul pulsante "Genera Report PDF" viene creato un report delle operazioni visualizzate, con tutte le informazioni ad esse associate.

4. Operazioni Giornaliere

Permette all'utente di visualizzare l'elenco delle operazioni relative alla data selezionata

Per ciascuna operazione vengono mostrate le informazioni relative a:

- Tipo di operazione (Pagamento o Entrata)
- Oggetto
- Importo
- Data



Cliccando su una delle voci dell'elenco si apre una Dialog con la lista completa delle informazioni associate all'operazione salvata. Se sono presenti informazioni relative al luogo in cui l'operazione è stata effettuata, è possibile visualizzare la posizione su Google Maps (tramite Web o la relativa app, se installata sul dispositivo).

Cliccando su "Elimina" è possibile cancellare dal DataBase tutti i dati relativi all'operazione ed eventuali immagini ad essa associata; dopo ogni cancellazione viene aggiornato il saldo attuale del conto.

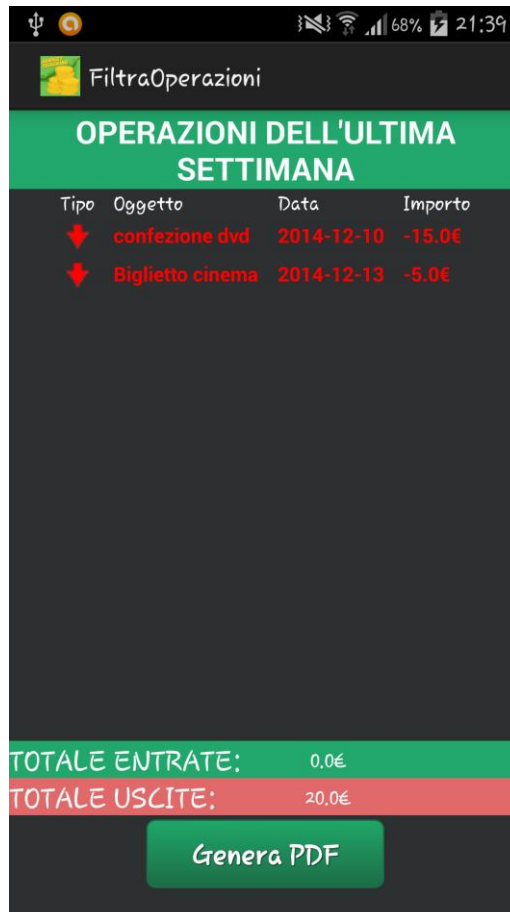
Cliccando sul pulsante "Genera Report PDF" viene creato un report delle operazioni visualizzate, con tutte le informazioni ad esse associate.

5. Operazioni Settimanali

Permette all'utente di visualizzare l'elenco delle operazioni effettuate nell'arco della settimana corrente (dal Lunedì al giorno attuale).

Per ciascuna operazione vengono mostrate le informazioni relative a:

- Tipo di operazione (Pagamento o Entrata)
- Oggetto
- Importo
- Data



Tipo	Oggetto	Data	Importo
↓	confezione dvd	2014-12-10	-15.0€
↓	Biglietto cinema	2014-12-13	-5.0€

TOTALE ENTRATE:	0,0€
TOTALE USCITE:	20,0€

Genera PDF

Cliccando su una delle voci dell'elenco si apre una Dialog con la lista completa delle informazioni associate all'operazione salvata. Se sono presenti informazioni relative al luogo in cui l'operazione è stata effettuata, è possibile visualizzare la posizione su Google Maps (tramite Web o la relativa app, se installata sul dispositivo).

Cliccando su "Elimina" è possibile cancellare dal DataBase tutti i dati relativi all'operazione ed eventuali immagini ad essa associata; dopo ogni cancellazione viene aggiornato il saldo attuale del conto. Cliccando sul pulsante "Genera Report PDF" viene creato un report delle operazioni visualizzate, con tutte le informazioni ad esse associate.

6. Operazioni Mensili

Permette all'utente di visualizzare l'elenco delle operazioni effettuate nell'arco del mese corrente (dal primo del mese, fino alla data corrente).

Per ciascuna operazione vengono mostrate le informazioni relative a:

- Tipo di operazione (Pagamento o Entrata)
- Oggetto
- Importo
- Data

Tipo	Oggetto	Data	Importo
↓	confezione dvd	2014-12-10	-15.0€
↑	Paghetta	2014-12-07	+10.0€
↓	Biglietto cinema	2014-12-13	-5.0€

TOTALE ENTRATE:	10.0€
TOTALE USCITE:	20.0€

Genera PDF

Cliccando su una delle voci dell'elenco si apre una Dialog con la lista completa delle informazioni associate all'operazione salvata. Se sono presenti informazioni relative al luogo in cui l'operazione è stata effettuata, è possibile visualizzare la posizione su Google Maps (tramite Web o la relativa app, se installata sul dispositivo).

Cliccando su "Elimina" è possibile cancellare dal DataBase tutti i dati relativi all'operazione ed eventuali immagini ad essa associate; dopo ogni cancellazione viene aggiornato il saldo attuale del conto.

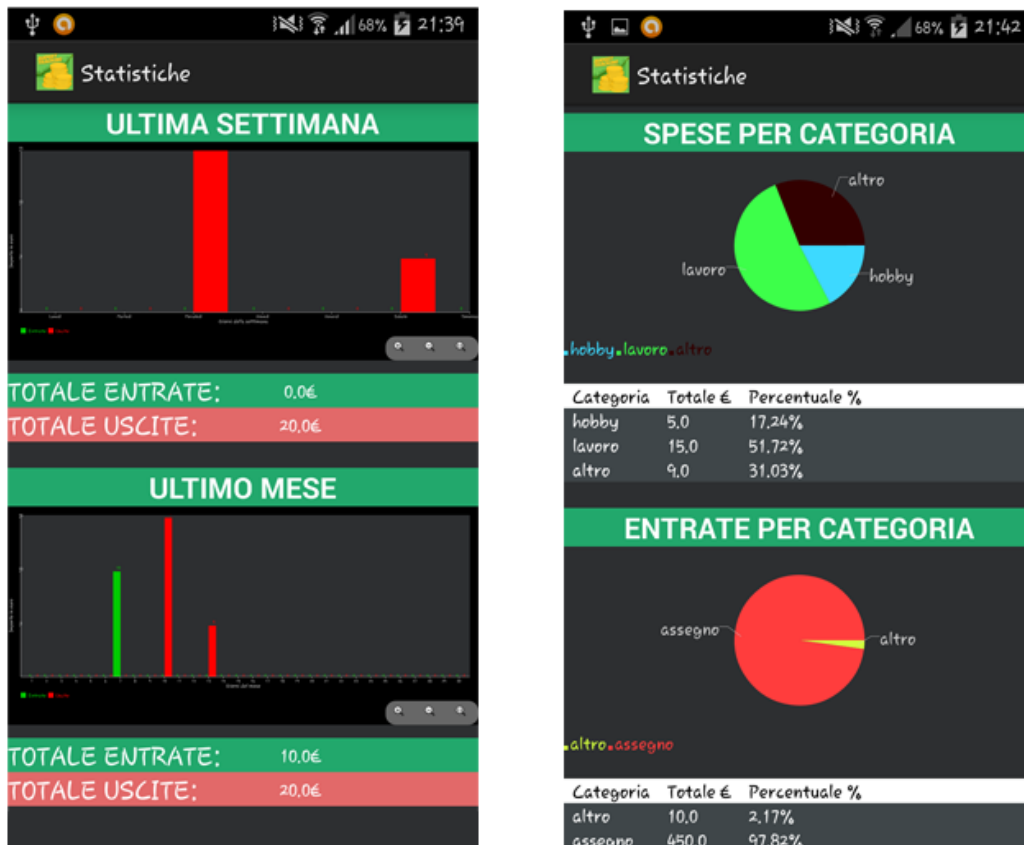
Cliccando sul pulsante "Genera Report PDF" viene creato un report delle operazioni visualizzate, con tutte le informazioni ad esse associate.

7. Statistiche

Permette all'utente di visualizzare utili grafici relativi alle transazioni effettuate. In particolare:

- BarChart relativo alle operazioni dell'ultima settimana
- BarChart relativo alle operazioni dell'ultimo mese
- PieChart con le percentuali delle spese effettuate per ciascuna categoria
- PieChart con le percentuali delle entrate ricevute per ciascuna categoria

Sotto ad ogni grafico sono visualizzate anche le informazioni relative ai totali delle spese e delle entrate.



Schermata di riepilogo

La schermata di riepilogo è quella che viene visualizzata ad ogni avvio dell'app. Racchiude le informazioni principali, in particolare:

- Saldo conto: lo stato attuale del conto. Cliccando sul pulsante "Mostra Dettagli" è possibile visualizzare tutte le informazioni associate all'account in uso, ovvero:
 - Nome Conto
 - Saldo Attuale
 - Saldo Iniziale
 - Descrizione
- Ultimi movimenti: elenco delle ultime 5 operazioni inserite. Cliccando sul pulsante "Mostra Tutti" è possibile visualizzare l'elenco completo delle operazioni inserite.
- Entrate vs Uscite: PieChart che mette in relazione le percentuali di Entrate e Uscite inserite, relative agli ultimi 30 giorni.



Progettazione e scelte implementative

Struttura del DataBase

La struttura del DataBase è la seguente:

Account (id, nome, descrizione, conto, conto_iniziale)

Operazioni (id, oggetto, importo, categoria, data, luogo_lat, luogo_lon, descrizione, foto, mezzo, tipo_operazione, pianificata, id_account)

Categoria (id, nome, tipo)

Op_periodica (id, id_operazione, periodo, data, next_data, tipo_operazione)

Per la persistenza dei dati è stato utilizzato il DataBase SQLite.

Per la gestione dei dati, si è scelto di ricorrere all'utilizzo di un DAO (Data Access Object) che si occupa di gestire la connessione al DataBase, accedere ai dati ed eventualmente modificarli; in questo modo qualunque modifica possa essere in futuro apportata alla struttura del DataBase non avrà il minimo impatto sul resto del codice, ma comporterà solo l'aggiornamento interno del DAO.

Tutte le activity manipolano e recuperano i dati utilizzando il DAO, senza preoccuparsi dei dettagli implementativi del DataBase.

Nello specifico, le classi che si occupano della gestione dei dati sono le seguenti:

- Database_string: fornisce le costanti statiche con i nomi delle tabelle, attributi e relazioni.
- Database: crea il database e popola la tabella delle categorie; estende SQLiteOpenHelper.
- DatabaseManager: il DAO; si occupa di gestire la connessione e contiene tutti i metodi per l'inserimento, la rimozione, la modifica e il recupero dei dati. I metodi, vengono utilizzati dalle altre classi, attraverso l'utilizzo di un oggetto di tipo DatabaseManager.

Gestione e rilevamento dei luoghi

Per l'implementazione della parte relativa alla ricerca di un luogo sono stati utilizzati i servizi di geolocalizzazione offerti dalle API Google Maps v.2 per Android.

Per poter fare qualsiasi richiesta alle API di Google Maps è stato necessario ottenere una API Key ed inserirla nell'applicazione, impostando i permessi corretti affinché potesse essere utilizzata.

La posizione corrente è rilevata grazie all'utilizzo dei Location Services e l'utilizzo della classe LocationClient.

Nelle activity che fanno uso del Location Services sono state implementate le due interfacce

- *GooglePlayServicesClient.ConnectionCallbacks*
- *GooglePlayServicesClient.OnConnectionFailedListener*

La posizione corrente viene recuperata chiamando il metodo `getLastLocation()`, che restituisce la posizione più recente sottoforma di un oggetto Location che contiene latitudine, longitudine ed altre informazioni utili relative al luogo.

Per la ricerca "manuale" di un luogo viene invece effettuata una richiesta HTTP all'url "<http://maps.google.com/maps/api/geocode/...>", passando come parametro l'input inserito dall'utente per la ricerca. La chiamata restituisce un oggetto di tipo JSON contenente i dati relativi ai vari risultati che corrispondono alla stringa ricercata. Questa chiamata HTTP viene effettuata da un thread secondario (come spiegato successivamente, nella sezione dedicata agli AsyncTask).

Gestione delle spese periodiche/programmate

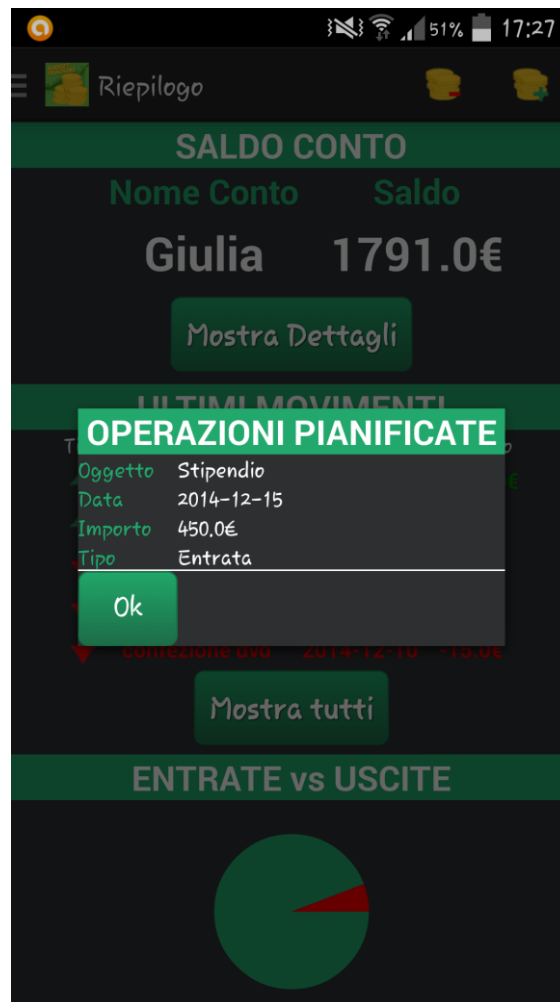
Quando l'utente inserisce un'operazione con una data futura, questa viene salvata sul DataBase nella tabella Operazioni con il campo pianificata = 1, in modo che non venga inserita immediatamente nell'elenco delle operazioni effettuate e non venga conteggiata nel saldo totale del conto.

Le operazioni periodiche vengono salvate sul DataBase nella tabella Operazioni e inoltre viene aggiunto un riferimento nella tabella Op_periodica nel seguente modo:

- Id_operazione: chiave esterna associata all'operazione principale (_id Operazione) ;
- Periodo: indica il tipo di ripetizione associata (giornaliera, settimanale, mensile)
- Data: data della prima ripetizione dell'operazione
- Data_next: data prevista per la successiva ripetizione dell'operazione

- Tipo_operazione: se si tratta di una Spesa o un'Entrata

All'avvio dell'applicazione vengono effettuati dei controlli su eventuali operazioni periodiche o programmate per la data odierna (o per i giorni precedenti, nel caso ancora non siano state inserite); se si trovano dei risultati, viene visualizzato una Dialog con la lista delle operazioni imminenti e viene aggiornato il saldo attuale del conto.



Generazione dei report .pdf

Per la creazione dei report .pdf è stata utilizzata la libreria iText, che consente appunto la creazione e la manipolazione di documenti in formato PDF.

La libreria è rilasciata con licenza AGPL dalla iText Software Corporation.

Generazione dei grafici

Per la creazione dei grafici è stata utilizzata la libreria AChartEngine.

La libreria, rilasciata con licenza Open Source, permette la generazione di vari tipi di grafici, per l'applicazione sono stati utilizzati PieChart e BarChart.

AsyncTask

Per evitare problemi, i task che potrebbero risultare lunghi o pesanti vengono eseguiti in background, grazie all'utilizzo di AsyncTask. In particolare AsyncTask viene utilizzato per le seguenti operazioni:

- La creazione dei report PDF: il thread in background si occupa di recuperare i dati relativi alle operazioni e successivamente di generare il file con l'elenco relativo. Al termine dell'operazione viene mostrato un messaggio di successo con il percorso di destinazione del file appena creato.
- Ricerca di un luogo: il thread in background effettua, tramite una richiesta HTTP, la ricerca delle coordinate (latitudine e longitudine) relative all'input inserito dall'utente. Al termine dell'operazione i risultati trovati vengono inseriti nella View, che viene aggiornata con i nuovi valori.

Per evitare crash improvvisi a causa di cambi di schermata, nel metodo onStop(...) delle activity che richiamano dei thread secondari, viene invocato il metodo cancel(...) sugli AsyncTask al momento attivi.

Difficoltà e soluzioni

Le principali problematiche e difficoltà sorte durante lo sviluppo dell'applicazione sono le seguenti:

1. Gestione e salvataggio delle date sul DataBase

La mancanza di un tipo di dato per la manipolazione delle date, in SQLite, mi ha causato alcuni problemi per quanto riguarda l'implementazione logica di alcune funzioni. Alla fine, dopo varie ricerche online, ho scelto di salvare le date come dei normali campi TEXT nel formato "yyyy-mm-dd" ed effettuare poi tramite Java la manipolazione di queste stringhe (con l'utilizzo della classe Calendar) in modo da poter effettuare controlli, confronti ed altre operazioni utili per il corretto funzionamento dell'applicazione.

2. Utilizzo dell'IDE Android Studio

Il problema più grosso che ho riscontrato nell'utilizzo dell'IDE Android Studio è sicuramente relativo all'emulatore; all'avvio impiegava diversi minuti prima di diventare operativo e la reattività del sistema risultava sempre molto scarsa, tanto da rendere snervante il suo utilizzo. Inoltre spesso si bloccava o non caricava le nuove istanze dell'applicazione, costringendomi ad un ri-avvio e successivi tempi di attesa di caricamento. Un'altra problematica riscontrata è relativa all'utilizzo del debugger, che spesso non si collegava all'emulatore; a volte bastava un riavvio del debugger da terminale, ma nella maggior parte dei casi si rendeva necessario un riavvio completo dell'IDE. Tutti questi problemi mi hanno successivamente spinto ad utilizzare un dispositivo esterno reale per effettuare il debug dell'applicazione, cosa che ha risolto per lo meno la questione relativa ai tempi di attesa e di esecuzione. Anche in questo caso però non sono mancati dei problemi, in quanto non sempre il device veniva rilevato e occorreva effettuare vari tentativi scollegando e ri-collegando il dispositivo.

Anche l'utilizzo dello stesso IDE Android Studio mi ha causato qualche problema, in quanto a volte segnalava errori nel codice anche dopo che questi erano stati risolti e i relativi file aggiornati, impedendomi di procedere con la compilazione del progetto; in questi casi occorreva chiudere e ri-aprire i file per risolvere il problema.

3. Gestione e salvataggio delle foto

All'inizio ho avuto qualche problema nella gestione dell'intent che richiama l'utilizzo della fotocamera, in quanto le immagini restituite erano sempre ruotate di 90° rispetto all'inclinazione reale. Sono servite numerose ricerche online prima di riuscire ad individuare la causa di questo problema e successivamente arrivare ad una soluzione (grazie soprattutto a stackoverflow.com).

Eventuali idee per estensioni future

Una possibile estensione dell'applicazione potrebbe consistere nel rendere disponibile la condivisione di un unico conto su più dispositivi e quindi tra account distinti: pensando ad una realtà familiare, potrebbe risultare utile avere un unico conto di riferimento in cui i vari componenti del nucleo familiare possono inserire, modificare e manipolare i dati.

Per ottenere un simile risultato occorrerebbe innanzitutto spostare il DataBase su un server online, in modo che possa essere accessibile in qualsiasi momento da più dispositivi; inoltre risulterebbe necessario pensare a delle politiche di sincronizzazione e sicurezza per l'accesso ai dati.

Inoltre potrebbe essere interessante, e molto utile, dare all'utente la possibilità di impostare delle scadenze periodiche (ad esempio ogni mese o ogni settimana..) in cui l'app generi automaticamente i report .pdf con l'elenco di tutte le operazioni effettuate.

Link utili e tutorial utilizzati

<http://www.achartengine.org/>

<http://itextpdf.com/>

<http://www.vogella.com/tutorials/AndroidSQLite/article.html>

<http://android.devapp.it/t014-eseguire-task-in-modo-asincrono-con-async-task>

<http://developer.android.com/google/play-services/setup.html>

<http://developer.android.com/training/location/index.html>

<http://developer.android.com/training/basics/data-storage/files.html>

<http://zacktutorials.blogspot.it/2014/07/android-read-and-write-pdf-file-using.html>

<http://stackoverflow.com/questions/6609414/howto-programatically-restart-android-app>

<http://developer.android.com/training/camera/photobasics.html>