

Carlo Fanciulli
Matricola 0000691815
Laboratorio di Applicazioni Mobili

Project MyBudget
C.d.L in Informatica per il Management
Università di Bologna

Relazione progetto LAM

Introduzione

Quest'app Android permette di tenere traccia delle entrate e uscite di carattere monetario dall'utente.

Questa applicazione permette di:

1. Aggiungere informazioni riguardante un entrata/uscita
2. Aggiungere portafogli per la gestione del budget
3. Aggiungere categorie per definire il tipo di entrata/uscita
4. Pianificare transazioni ricorrenti che si aggiungono ogni n°giorni
5. Visualizzare statistiche riguardo le transazioni per categoria
6. Visualizzare la cronologia della transazioni avvenute
7. Visualizzare la mappa con le posizioni nelle quali sono avvenute le transazioni
8. Esportare il pdf con le transazioni

Lo scopo dei capitoli successivi è descrivere i requisiti e le dipendenze utilizzate per questa applicazione, i dispositivi utilizzati per il testing e spiegare nei dettagli ogni punto scritto in precedenza.

Requisiti e Dipendenze

La versione minima supportata corrisponde alla API 19 (Android 4.4– KitKat).

L'ambiente di sviluppo utilizzato è Android Studio 2.3.

La CompileSdkVersion corrisponde alle API 24.

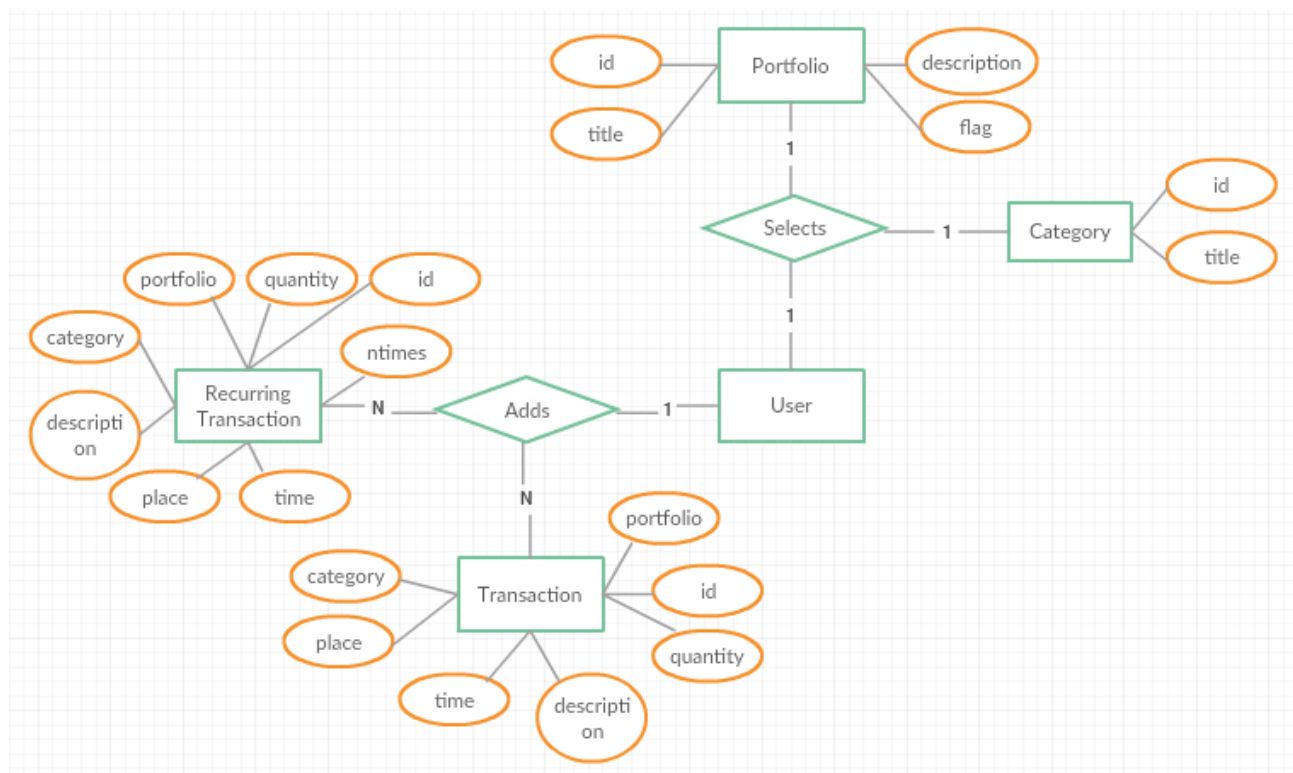
Le dipendenze sono:

- **com.android.support:appcompat-v7:24.2.1**
- **com.android.support:design:24.2.1**
- **com.android.support:cardview-v7:24.2.1**
- **com.android.support:gridlayout-v7:24.2.1**
- **com.google.android.gms:play-services-maps:10.0.1**
- **com.github.PhilJay:MPAndroidChart:v2.1.6**
- **com.itextpdf:itextg:5.5.10**

Dispositivi utilizzati

Il testing dell'applicazione è stato effettuato su: il mio dispositivo android personale Xiaomi Redmi Note 3 Pro, un Samsung Galaxy S6 e l'emulatore di Android Studio con il Nexus 5.

Modello ER / Database



I dati vengono salvati localmente attraverso l'utilizzo del database SQLite.
Il database utilizzato è composto da 4 tabelle descritte qui di seguito:

1. **PORTFOLIOS**: tabella contenente informazioni riguardanti i portafogli
I Dati inseriti sono id, titolo, descrizione e flag.

2. **CATEGORIES**: tabella contenente informazioni riguardanti le categorie
I Dati inseriti sono id, titolo.

3. **CASHFLOW**: tabella contenente informazioni riguardanti le transazioni
I Dati inseriti sono id, quantità, portafoglio corrispondente, categoria, descrizione, data, luogo.

4. **RECURRENT**: tabella contenente informazioni riguardanti le transazioni ricorrenti.

I Dati inseriti sono gli stessi della tabella PORTFOLIOS. In più è stato aggiunto il campo ntimes che tiene in considerazione ogni quanti giorni si deve aggiungere un record In PORTFOLIOS.

Per la gestione del database e le relative query sono state create due classi. DatabaseHelper per la creazione delle tabelle e i record di default e la classe DBManager in cui vengono create tutte le query che vengono utilizzare durante tutto l'utilizzo dell'applicazione.

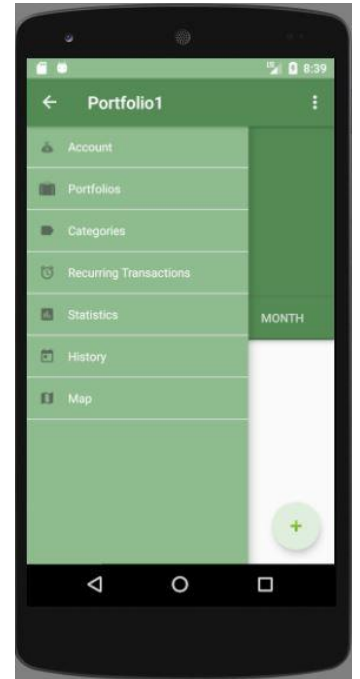
Architettura Applicazione

1. Menu

Il menu principale permette di spostarsi all'interno dell'applicazione.

In ogni activity è presente l'Hamburger Button che permette di visualizzare il menu.

Le scelte per implementare il menù in ogni activity erano due. O fare una Base Activity che fosse poi estesa ad ogni activity e usare poi un normale intent al click della scelta nel menù oppure non considerarle Activity e fare tutto tramite Fragment. E' stato scelto di provare con la prima opzione in quanto i Fragment erano stati già utilizzati, come vedremo nel prossimo capitolo, per visualizzare le ListView con le transazioni.



2. Budget Activity

Questa è l'activity principale che permette di visualizzare le principali informazioni dell'applicazione.

In particolare nella toolbar, come si può vedere dall'immagine, è possibile sapere la data corrispondente alla tab selezionata (TODAY – WEEK – MONTH) e i soldi totali sempre in base alla tab. In questo modo si può capire il budget acquisito dalle entrate e uscite importate selezionando la tab relativa al giorno, alla settimana o al mese.

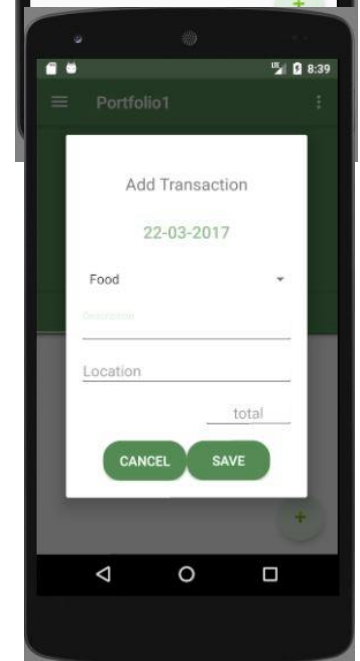
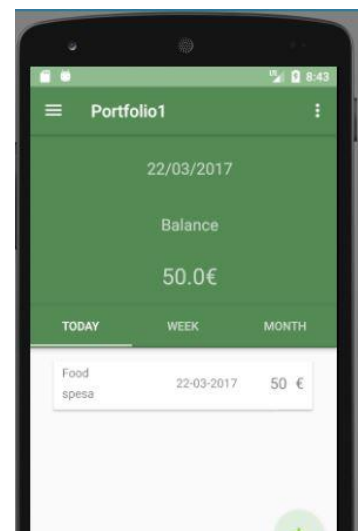
In alto si può vedere l'hamburger icon che permette di aprire il menu, il nome del portafoglio selezionato (la prima volta che viene aperta l'app crea di default un portafoglio vuoto) e sulla destra un menu che permette di estrarre, come vedremo in seguito, le transazioni effettuate in un pdf.

A questa activity ne sono collegate altre due.

Una che gestisce l'inserimento di una nuova transazione e una che gestisce la modifica e l'eliminazione.

Entrambi sono visibili attraverso Dialog.

Attraverso il f.a.b è possibile aprire il Dialog per la creazione di una nuova transazione che permette di scegliere in quale data far riferire quella transazione, la categoria corrispondente che pesca tra le varie categorie già nel database, la location che ha un sistema di autocompletamento del luogo che si sta per scrivere e il totale in euro.



Dalla ListView è possibile, con un long click, aprire la seconda dialog per modificare ed eliminare la transazione relativa.

3. Portfolios Activity

Questa Activity permette la gestione dei portafogli utilizzabili all'interno dell'applicazione.

Ogni transazione infatti è associata ad un portafoglio che è possibile selezionare da questa activity.

Oltre che selezionare quello da usare, è anche possibile crearlo uno nuovo oppure eliminare quello scelto.

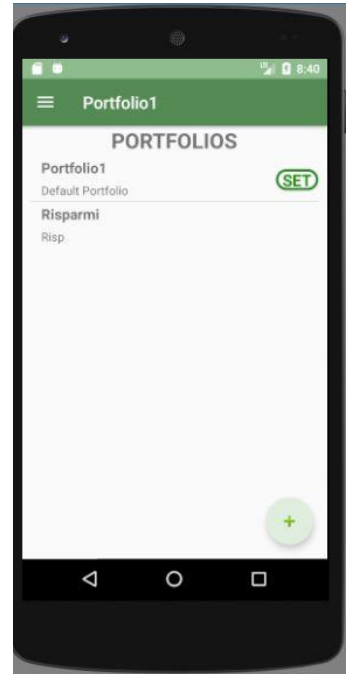
La ListView che prende sempre i dati dal database è popolato attraverso un Adapter customizzato che tramite il metodo getView permette di popolare la ListView personalizzando ogni elemento, in particolare rendere visibile o nascondere la ImageView "SET" in base ad un campo nella tabella PORTFOLIO del database.

Se è presente un solo portafoglio (quindi settato attivo di default) non è possibile eliminarlo.

Verrà avvertito l'utente di ciò attraverso una Snackbar.

Quando un portafoglio è settato attivo verrà automaticamente utilizzato in ogni altra activity, come ad esempio durante la creazione di una nuova transazione.

Ovviamente è possibile aggiungerne altre oppure eliminarle.



4. Categories Activity

Per questa activity è stato usato lo stesso meccanismo di creazione dell'activity precedente ma per popolare la ListView è stato utilizzato un SimpleCursorAdapter.

Di default, appena viene installata la prima volta l'applicazione, viene creato un insieme di categorie base che sono: Food, Rent, Salary, Savings e Bills.

Ovviamente è possibile aggiungerle altre oppure eliminarle.

Tutte queste categorie saranno visibili durante la creazione di una transazione attraverso uno spinner che permette di vederli attraverso un menu.

5. Recurring Activity

In questa activity è possibile gestire le transazioni ricorrenti, cioè che aggiungono una transazione ogni n giorni. (un record nella tabella CASHFLOW).

E' stato necessario creare una tabella apposita per gestire le transazioni ricorrenti in quanto era fondamentale tenere presente la data dell'ultima transazione che è runnata o deve runnare e ogni quanti giorni deve accadere.

Questi due campi sono rispettivamente time e ntimes.

E' stato deciso di eseguire la funzione che permette il meccanismo di aggiunta ogni volta che si aggiunge una transizione

ricorrente e ogni volta che viene aperta l'activity principale (cioè Budget Activity).

Ovviamente per evitare di fare ripetutamente la stessa funzione ogni volta che viene raggiunta la Budget Activity è stato implementato un controllo tra la data attuale e la data in cui è stata aperta l'ultima volta l'applicazione.(utilizzando le Shared Preferences)

Viene svolta la funzione solamente se le date son diverse.

La funzione di controllo esegue una verifica che la data presente nel database RECURRENT sommata al numero di giorni ntimes sia minore o uguale alla data corrente. Nel caso la data sia minore o uguale viene eseguita la transazione ricorrente sul database principale e la data presente in RECURRENT viene aggiornata alla data+ntimes.

6.Statistics Activity

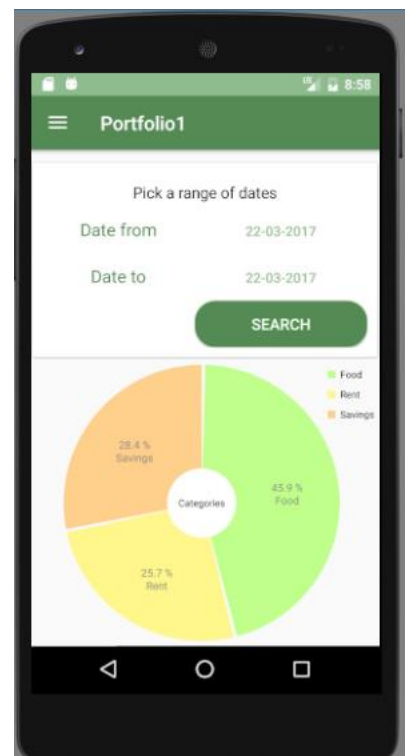
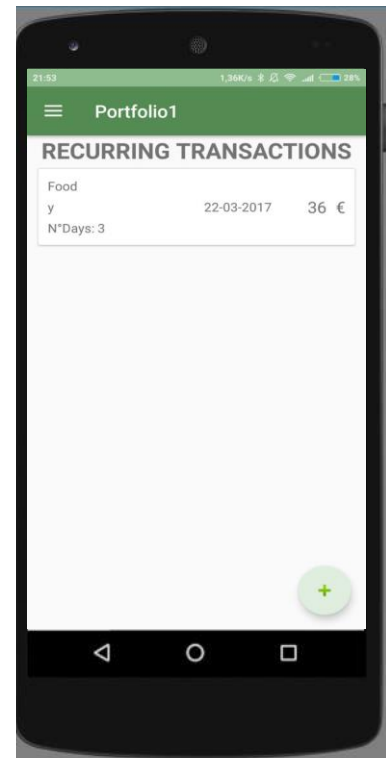
In questa activity lo scopo principale è visualizzare un grafico con le percentuali di distribuzione della spesa nelle varie categorie.

In input vengono inserite le date su cui vogliamo che venga effettuato il grafico così da filtrare i risultati ottenuti da una query al database.

La libreria utilizzata per generare il grafico è la MPAndroidChart.

La scelta era principalmente tra l'usare un grafico a barre oppure un grafico a torta.

Soprattutto per una scelta estetica è stato implementato il grafico. In particolare è stata inserita una legenda contenente le categorie, un animazione per muovere il grafico e l'aggiunta di colori ognuno diverso per categoria.



6. History Activity

Nella History è stato riutilizzato lo stesso layout per l'inserimento delle date per la ricerca.

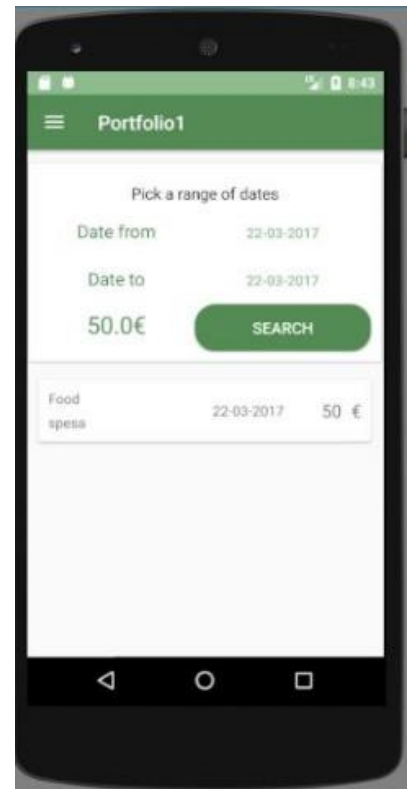
Il layout è composto da una CardView e un GridLayout per l'ordinamento degli elementi più le varie TextView e Button.

Il search permette di visualizzare la solita ListView contenente le transazioni avvenute durante i giorni cercati in input.

E' stato utilizzato sempre un SimpleCursorAdaptor per popolare la ListView ed è stato deciso di non poter più modificare o eliminare le transazioni.

Anche se all'inizio era stato omissso, è possibile anche visualizzare la data in cui è stata fatta la transazione, oltre a tutte le altre informazioni tranne la location.

Per questo è stata implementata l'ultima activity dell'applicazione: la Location Activity.

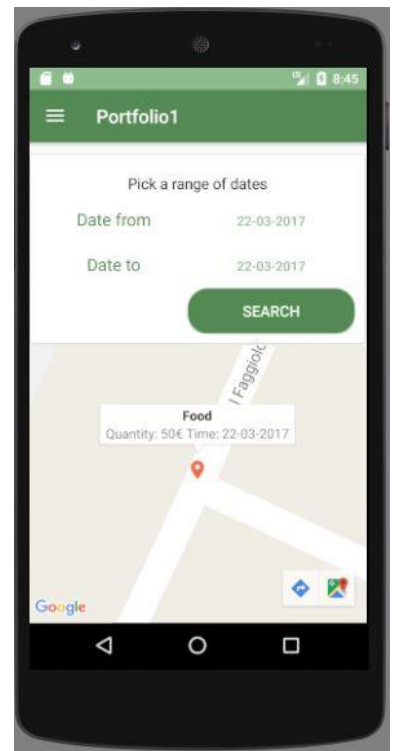


7. Location Activity

In questa activity viene visualizzata una mappa dopo aver inserito in input di quali giorni si vuole visualizzare la location delle transazioni.

Per la mappa sono state utilizzate le API di Google Maps e la classe Geocoder per la conversione di indirizzi testuali (presi e filtrati in base alla data dal database) in coordinate geografiche per poi posizionare i marcatori per identificare ogni transazione. Ovviamente ogni marcatore racchiude le informazioni base della transazione.

Per utilizzare le API è stato necessario abilitare i permessi di accesso ad internet android.permission.INTERNET.



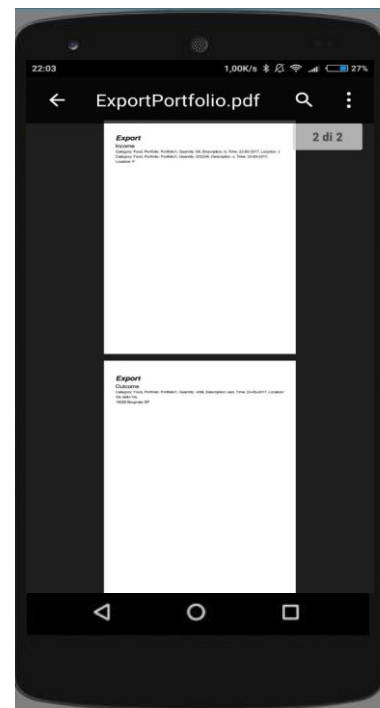
8. Export Activity

Questa activity creata in formato Dialog è accessibile dalla main Activity (Budget Activity) attraverso il menu in alto a destra.

Anche in questo caso viene utilizzato lo stesso layout usato precedentemente per la richiesta delle dati su cui vogliamo fare l'export delle transazioni che ci interessano.

Per poter esportare e poter scrivere all'interno del dispositivo sono stati richiesti due permessi: `READ_EXTERNAL_STORAGE` e `WRITE_EXTERNAL_STORAGE`.

Dopo il controllo per poter scrivere, viene utilizzata una libreria esterna chiamata ITEXTG per poter creare e modellare il pdf. Con questo intendiamo la creazione di due fogli con all'interno un titolo e un sottotitolo, uno per le entrate e uno per le uscite. Con questa divisione viene facilitata la visione delle informazioni delle transazioni, che consistono in elenchi con ogni dato presente nel database.



Fonti e Riferimenti

Studio Android:

developer.android.com/guide/index.html

Problemi in generale:

stackoverflow.com

www.google.it

Implementazione mappa:

www.vogella.com/tutorials/AndroidGoogleMaps/article.html

developers.google.com/maps/documentation/android-api/

Libreria per grafico:

github.com/PhilJay/MPAndroidChart

Export in pdf:

developers.itextpdf.com/itextg-android