Technical Report

Laboratorio di Applicazioni Mobili 2017/2018

Cristian Romanello – matr. 722443

Applicazione Android

"Money Control"

Introduzione

L'applicazione è stata sviluppata cercando di seguire, a monte, un'analisi degli strumenti più consoni da utilizzare, così da produrre un applicativo quanto più user-friendly possibile.

L'argomento trattato dall'applicazione consiste nella rivisitazione di una traccia proposta negli anni precedenti, ovvero l'"Android Budget Tracker". Lo scopo è quello di poter registrare e tener traccia nel tempo delle spese di tutti i giorni.

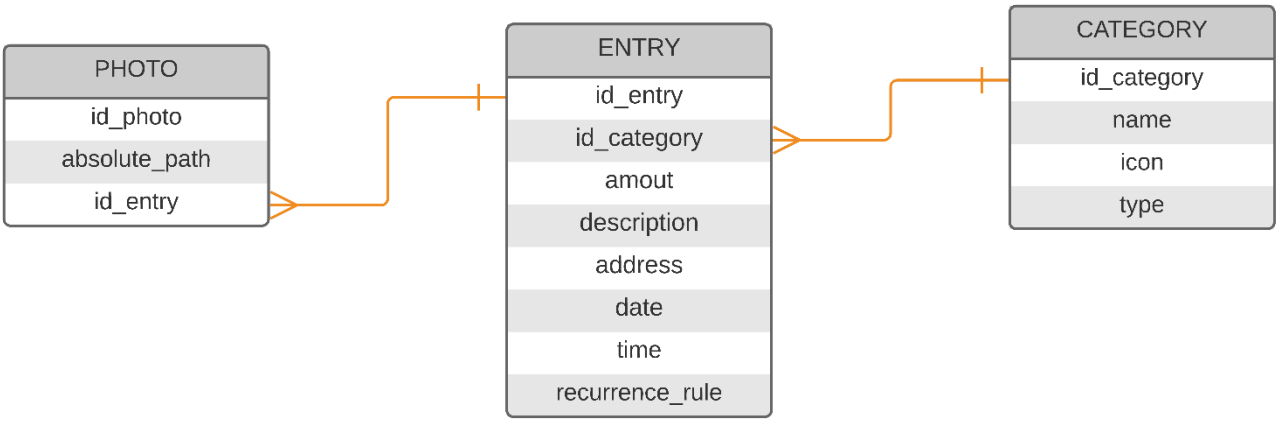
Analisi

L'analisi, prima di produrre il codice vero e proprio, è stata effettuata sulle applicazioni già esistenti che trattano questo argomento: il budget tracking. Alcune delle più scaricate da Google Play:

- Money Pro
- Gestore Spese
- Money Manager
- MoneyOne
- Wallet

Una panoramica dei principali spunti analizzati è disponibile a questo link: Presentazione.

Modello ER



Il modello è disponibile anche online al seguente link: Modello.

Progettazione Grafica

La fase successiva all'analisi è stata la progettazione grafica, dove ho utilizzato uno strumento disponibile online gratis (solo per studenti verificati): LucidChart. Il risultato ottenuto è una rappresentazione (graficamente semi-realistica) navigabile (comprensiva di collegamenti fra le varie schermate, opportunamente evidenziati).

Il Mockup è disponibile online al seguente link: Mockup.

Project Tracking

Una volta in possesso di tutti gli artefatti antecedenti lo sviluppo, ho adottato uno strumento finale per tenere traccia delle varie task durante le fasi della creazione del progetto: Freedcamp.

La Project Task List è disponibile online al seguente link: Task List.

Project Versioning

Dall'inizio alla fine della realizzazione del progetto è stato utilizzato lo strumento di versioning GitHub.

Il progetto è disponibile a questo indirizzo: GitHub Project.

Scopo Applicazione

Lo scopo dell'applicazione è quello di poter tracciare le spese (ed entrate) di ogni giorno e visualizzare report con diversa periodicità.

Funzionalità

Le funzionalità principali sono:

- Visualizzare le spese/entrate del giorno corrente ("oggi")
- Aggiungere una nuova spesa/entrata tramite la selezione di una categoria fra quelle proposte
- Inserimento manuale di un indirizzo fisico (agevolato dalla ricerca Google)
- Inserimento automatico di un indirizzo (tramite la posizione del dispositivo)
- Scattare foto relative ad una spesa/entrata
- Visualizzare le voci in una suddivisione di bilancio per giorno/mese/anno (grafici inclusi)
- Visualizzare le voci su una mappa interattiva (avendo poi la possibilità di riposizionare i luoghi delle spese/entrate)

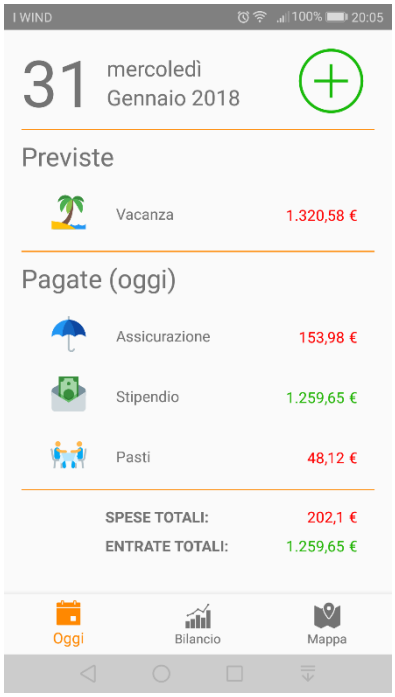
Requisiti

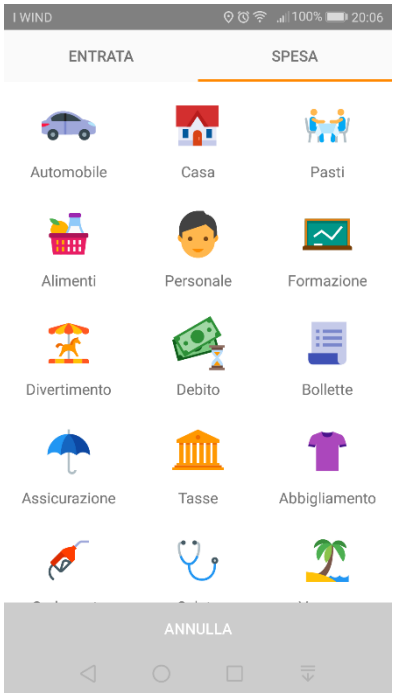
L'applicazione prevede i seguenti requisiti:

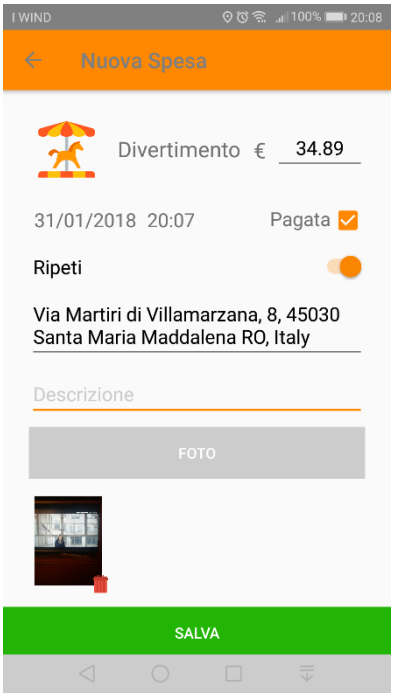
- Compile SDK Version: API 26: Android 8.0 (Oreo)
- Ambiente di sviluppo: Android Studio 3.0.1

Caratteristiche

L'applicazione, una volta avviata, presenta una prima schermata di caricamento; successivamente appare la schermata principale vera e propria:

1) 

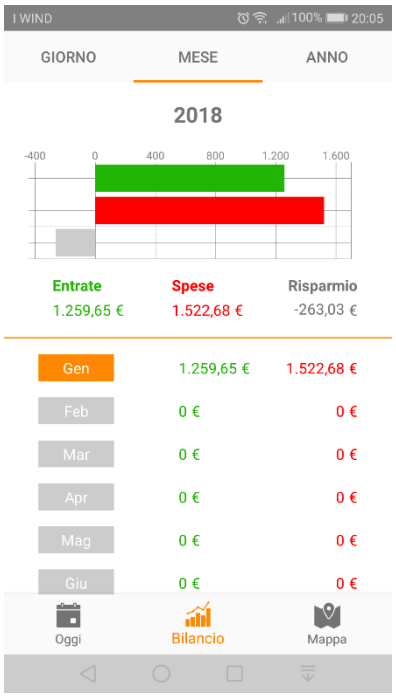
2) 

3) 

- Toccando il "+" verde in alto a destra si ha la possibilità di aggiungere una nuova spesa/entrata
- Toccando la categoria si passa alla schermata di inserimento dei dati ed eventuale posizione e foto

Selezionando la voce "Bilancio", presente nel menu nella parte inferiore dell'applicazione, si raggiunge la sezione dedicata al controllo delle spese 4).

Tramite la voce "Mappa", invece, si raggiunge la mappa interattiva 5).

4) 

5) 

In quest'ultima è possibile, inoltre, visualizzare le spese nei rispettivi luoghi, con i relativi dettagli. Le immagini dei markers all'interno della mappa sono le stesse icone utilizzate per raffigurare le diverse categorie. Tenendo premuto su un marker (icona categoria) è possibile riposizionarlo all'interno della mappa, salvando automaticamente la nuova posizione. Inoltre se si clicca all'interno del fumento contenente i dettagli delle spesa/entrata, si verrà reindirizzati alla pagina di modifica di quella specifica voce.

Progettazione e Scelte implementative

La progettazione ha seguito i principi e linee guida proposti dalla documentazione Google Android.

Struttura

La struttura del progetto prevede l'utilizzo di alcune activities principali:

- **StartActivity**: visualizza una semplice animazione di caricamento prima di invocare la MainActivity
- **MainActivity**: contiene il bottom navigation menu con i tre items "Oggi", "Bilancio" e "Mappa", ognuno dei quali invoca il corrispondente fragment
- **AddNewEntryActivity**: contiene i fragments necessary per visualizzare il tab menu ("Entrata"/"Spesa")
- **EntryDetailsActivity**: visualizza la schermata per l'inserimento (o modifica) di tutti i dati di una entrata/spesa
- **SingleDayActivity**: raggiungibile tramite la selezione di un determinato giorno dalla sezione "Bilancio", visualizza l'elenco delle entrate/spese per il giorno (mese e anno) selezionato

All'interno delle diverse activities vengono richiamati i seguenti fragments:

- **TodayFragment**: contiene tutte le varie sezioni della schermata principale (data, spese/entrate previste, quelle della giornata odierna e il totale)
    - Le spese/entrate previste sono riconosciute dal fatto di possedere almeno un'occorrenza di evento (rispetto alla serie) appartenente al futuro, oppure che la data stessa sia nel futuro
- **AddNewExpenseFragment**: visualizza l'elenco delle categorie per le spese
- **AddNewIncomeFragment**: visualizza l'elenco delle categorie per le entrate
- **BalanceFragment**: visualizza il tab layout (menu tab) navigabile nella parte superiore della schermata; al di sotto contiene il ViewPager, per poter scambiare i vari fragments (corrispondenti alle voci del tab menu: "Giorno", "Mese", "Anno")
- **DailyBalanceFragment**: visualizza il bilancio con l'elenco dei giorni (default: mese attuale)
- **MonthBalanceFragment**: visualizza il bilancio con l'elenco dei mesi (default: anno attuale)
- **YearlyBalanceFragment**: visualizza il bilancio per anno
- **BaseGoogleMapsFragment**: contiene il fragment all'interno del quale viene visualizzata la mappa

Vengono, inoltre, utilizzate le seguenti classi accessorie:

- **BalanceGrid**: estende la classe BaseAdapter per poter inserire all'interno della GridView (presente nei fragment Balance Giorno/Mese/Anno) l'elenco delle singole voci
- **CategoryGrid**: estende la classe BaseAdapter per poter inserire all'interno della GridView (presente nei fragment AddNewExpense e AddNewIncome) l'elenco delle singole categorie
- **EntryGrid**: estende la classe BaseAdapter per poter inserire all'interno della GridView (presente nel fragment Today e nell'activity SingleDay) l'elenco delle singole voci di entrata/spesa
- **CustomCalendar**: offre alcuni metodi fondamentali per l'utilizzo, conversione e confronto delle date
- **GPSLocator**: permette la localizzazione del dispositivo tramite qualunque delle tre tipologie ("GPS, Wi-Fi e reti mobili", "Wi-Fi e rete mobile" e "solo GPS"). Essa, inoltre, espone alcuni metodi per trasformare le coordinate latitudine-longitudine in indirizzi fisici e viceversa
- **Utils**: offre alcuni metodi di supporto, quali ad esempio la formattazione degli importi e la costruzione delle mappe Map<Key, Value> contenenti gli oggetti presenti nel database

Persistenza

Il mantenimento della persistenza è gestito tramite la libreria "Room".

In particolare, si è scelto di implementare l'accesso al database tramite l'utilizzo di una classe singleton "AppDatabase" la quale estende, per il necessario funzionamento, "RoomDatabase".

Per rappresentare la realtà di riferimento si sono poi create le singole classi (precedentemente graficate nell'ER):

- Category: contiene tutti i dati descrittivi di una categoria (id, name, icon, type [expense/income])
- Entry: contiene tutti i dati descrittivi di una singola voce di entrata/spesa (id, id_category, amount, description, address, dateTime, recurrenceRule)
- Photo: contiene principalmente i dati di una foto (id, absolute_path, id_entry)

Per poterle utilizzare è però necessaria la gestione delle interfacce Dao, le quali contengono il codice SQL vero e proprio che viene poi eseguito sul database.

Tutto quello che riguarda il database, ovvero la classe AppDatabase, le singole classi-oggetto e le classi Dao, sono contenute, per puro scopo organizzativo, in una cartella denominata "database".

Difficoltà e Soluzioni

Di seguito le principali difficoltà (poi risolte) riscontate durante lo sviluppo dell'applicazione:

Foto

L'implementazione della possibilità per l'utente di scattare foto e salvarle è stata particolarmente articolata (e qui risiede la difficoltà). Il punto cruciale è stato quello di richiedere i permessi a runtime (per l'accesso alla fotocamera) e salvare successivamente le foto sulla memoria interna del dispositivo. Il meccanismo implementato prevede la creazione di un file vuoto, ancor prima che la foto venga scattata. Una volta scattata, viene poi "inserita" nel file vuoto; per tenere traccia però della relativa spesa/entrata si è dovuto memorizzare i path assoluti nel database, poiché l'id della entry è noto solamente una volta inserita nel database, e non a priori. Dal punto di vista prettamente tecnico sarebbe anche stato possibile prevedere (in modo programmatico e imposto) l'id della nuova entry che si andrà a creare, ma non è stato ritenuto un approccio metodologicamente corretto, e quindi non implementato.

L'altro aspetto rilevante riguardante le foto è stato quello di creare una sezione dedicata alla visualizzazione, sotto forma di thumbnails e fullscreen, delle foto acquisite (nella pagina di ogni specifica entry).

Device Location

Questo punto è stato quello che, fra tutti, ha richiesto più tempo.

La "causa" è stata principalmente che l'applicazione è stata sviluppata testandola (continuamente) su un device reale, però di marca Huawei. Molteplici sono stati i tentativi (vani) di localizzare il device, anche semplicemente eseguendo applicazioni sorgenti fornite direttamente da Google (tramite GitHub). Il codice non forniva mai i valori di latitudine e longitudine. Successivamente è stato appurato, anche grazie ad un breve colloquio con il Professor. Bedogni, che una possibile causa risiede proprio nella marca stessa del telefono, o meglio, nella rivisitazione da parte di quest'ultima di Android.

Il problema è stato risolto solamente grazie al ritrovamento in rete di un codice effettivamente funzionante, il quale utilizza comunque un semplice LocationManager.

Aspetti Rilevanti

Fragments

Banale non è stata la gestione dei fragments, ma si è scelto comunque di operare con essi, soprattutto per la loro versatilità; ma anche per non lanciare, ogni volta, una nuova activity, cosa che comporterebbe uno sgradevole effetto visivo per l'utente, a causa della continua ristampa dei menu principali (superiore e inferiore).

Performances

L'applicazione è nel suo insieme veloce, ma in fase di sviluppo si era evidenziata una carenza in termini di velocità di caricamento delle sezioni "Bilancio" e "Mappa", testandola con un numero elevato di entry.

Questo è stato risolto introducendo, all'inizio delle sole classi che ne necessitano, un meccanismo di salvataggio pre-utilizzo dei record (interessati) presenti nel database. Dal punto di vista pratico si è trattato di creare, nella classe Utils, un metodo che estrae tutte le entry (o categorie) dal database e le mappa in una struttura Map<Integer, Entry>. Quest'ultima viene poi ritornata alla classe che ha invocato il metodo e così può accedere rapidamente alla entry di interesse, semplicemente specificandone l'id.

Estensioni

L'applicazione potrebbe risultare estremamente utile, seppur molto simile a tantissime altre già in commercio, per tenere conto delle spese ed entrate della vita quotidiana. Per renderla ancora più fruibile e con maggiori interessanti funzionalità si potrebbe:

- Implementare un sistema di notifiche per le spese non ancora contrassegnate come pagate e con eventi ricorrenti (parzialmente già presente)
- Introdurre un sistema di sotto-categorie
- Possibilità di creare nuove categorie personalizzate (con tanto di icona da scegliere) e modificarle
- Implementare una ricerca per categoria ed altri parametri rilevanti
- Visualizzare statistiche raggruppate anche per categorie principali

Librerie Utilizzate

- Auto-complete Address (Google) (https://github.com/seatgeek/android-PlacesAutocompleteTextView)
- BetterPickers (Recurrence Events) (https://github.com/code-troopers/android-betterpickers)
- Recurrence Rule Parser (https://github.com/dmfs/lib-recur)
- Icons8 (Premium) (https://icons8.it/icon/new-icons/all)
- MPAndroidChart (https://github.com/PhilJay/MPAndroidChart)
- GPS Location (https://www.anddev.it/index.php?topic=13374.0)
- Google Maps (https://developers.google.com/maps/documentation/android-api)