

# Money Tracking

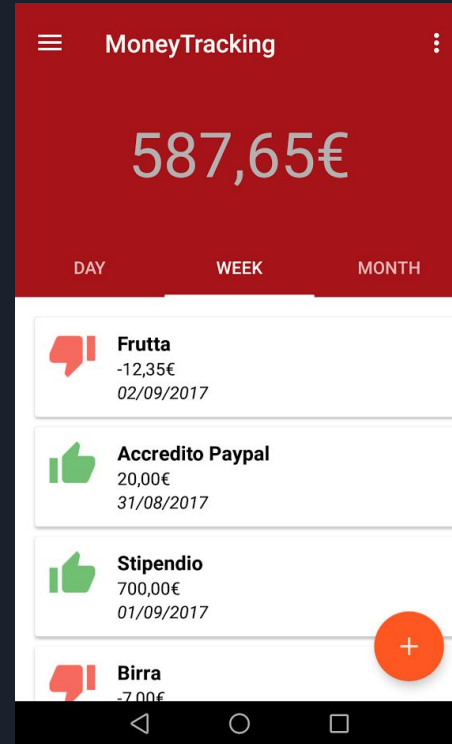
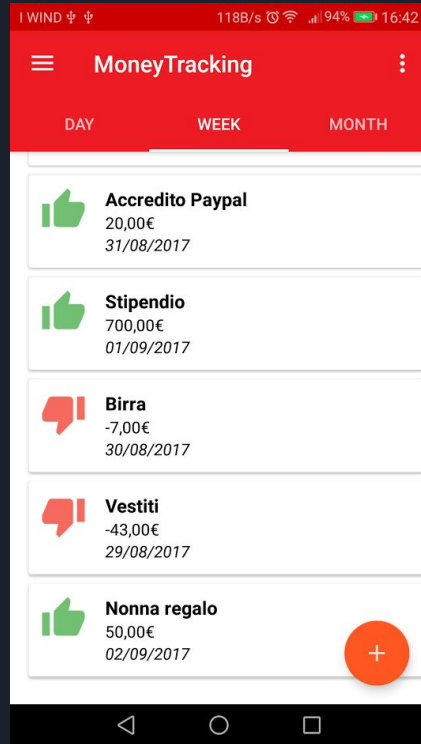
Alessio Koci 0000731157  
LAM 2016/2017



# MONEYTRACKING - FUNZIONALITÀ

- Aggiungere modificare ed editare transazioni ( spesa, profitto )
- Divisione delle transazioni per categorie personalizzabili
- Inserire transizioni pianificare e ricorrenti
- Generare report giornalieri, settimanali, mensili e annuali
- Visualizzare grafici
- Ricevere notifiche sulle transazioni pianificate o ricorrenti
- Visualizzare su mappa il luogo in cui è stata effettuata la spesa o guadagno
- Visualizzare la lista di tutte le transazioni effettuate

# HOME



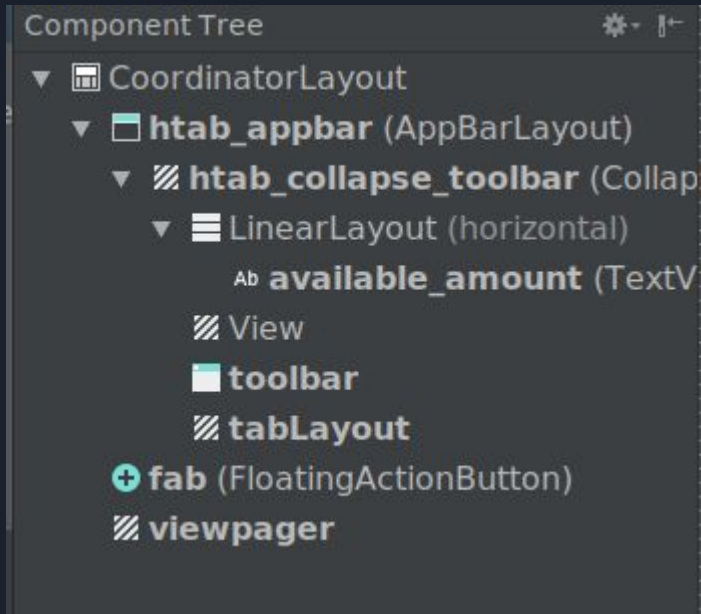


# HOME - LE FUNZIONALITÀ

- La pagina iniziale mostra il budget totale calcolato prendendo in considerazione tutte le transazioni effettuate
- E' possibile visualizzare le transizioni raggruppate in 3 diverse modalità :  
Giornaliera, Settimanale, Mensile
- Cliccando sul tasto “+” sarà possibile aggiungere una nuova transizione

# HOME - LAYOUT

Il layout è composto principalmente dalla CollapsingToolbar e il ViewPager che mostra le TAB con le transazioni



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/htab_appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:fitsSystemWindows="true"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

        <android.support.design.widget.CollapsingToolbarLayout...>

    </android.support.design.widget.AppBarLayout>

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        app:srcCompat="@drawable/ic_add" />

    <android.support.v4.view.ViewPager
        android:id="@+id/viewpager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="android.support.design.widget.AppBarLayout"

</android.support.design.widget.CoordinatorLayout>
```



# HOME - CODICE

- **MainActivity:** (inizializzazione della view)
  - **ViewPagerAdapter:** Adapter la la View principale della pagina contenenti le tab
  - **TabFragment:** Visualizza le transazioni richiedendo al database le transizioni in base alla tab selezionata ( day, week, month)
  - **MoneyItemAdapter:** Adapter per RecyclerView
  - **DBHelper:** facilita la connessione, e le query con il database SQLite

```
private void init_tabview() {  
    viewPager = (ViewPager) findViewById(R.id.viewpager);  
    vpage_adapter = new ViewPagerAdapter(getSupportFragmentManager());  
    TabFragment one = new TabFragment().newInstance(1), two = new TabFrage  
    vpage_adapter.addFragment(one, "Day");  
    vpage_adapter.addFragment(two, "Week");  
    vpage_adapter.addFragment(three, "Month");  
    viewPager.setAdapter(vpage_adapter);  
    final TabLayout tabLayout = (TabLayout) findViewById(R.id.tabLayout);  
    tabLayout.setupWithViewPager(viewPager);  
}
```

# NUOVO ELEMENTO

I WIND 72,9K/s 92% 19:23

Name  
T

Description

Location

Date

\$ Amount

Food

**Planned**

Daily

Repeat

ADD

I WIND 67,8K/s 92% 19:23

< New Item

Name  
T

Description

Location

Date

\$ Amount

Food

ADD



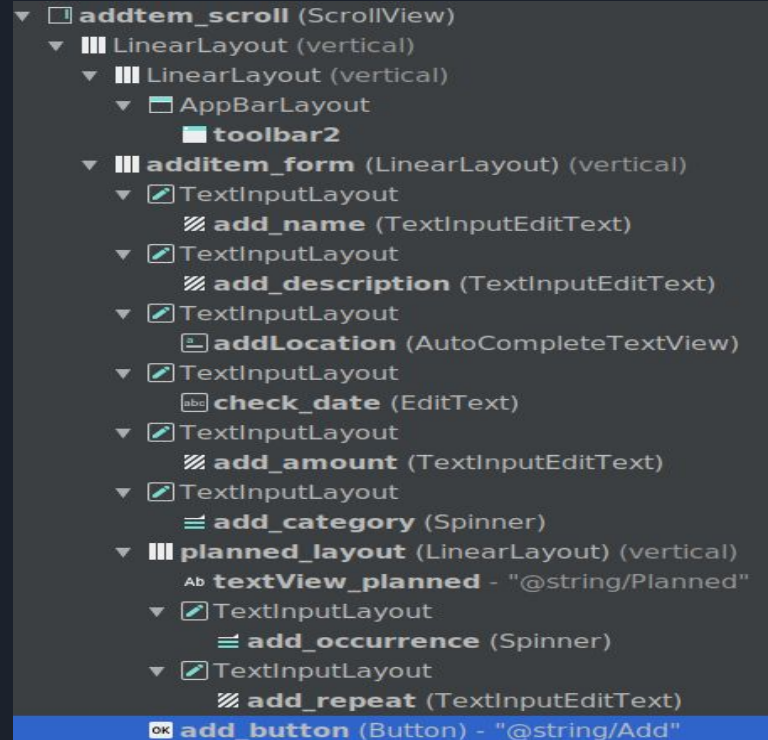
# NUOVO ELEMENTO - LE FUNZIONALITÀ

- Attraverso questa pagina sarà possibile aggiungere una nuova transizione indicando: nome, descrizione, posizione, data, spesa/guadagno, categoria, ( repeat, occurrence )
- Cliccando sulla sezione “Date” si aprirà una finestra che permetterà di selezionare in maniera interattiva e grafica la data della transizione
- Iniziando a scrivere sulla casella di testo “Location”, se si dispone di una connessione internet, verranno mostrati dei suggerimenti richiesti usando le API di Google Maps
- Cliccando sul tasto “ADD” la spesa o il guadagno verrà memorizzato nel database e sarà visibile nella home, nell’archivio, o nella sezione “Planned” se pianificato



# NUOVO ELEMENTO - LAYOUT

- Il layout per una “nuova transizione” e per una “transazione pianificata” o ricorrente è la stessa!
- Il layout è utilizzata anche per la modifica di una transizione
- Attraverso la modifica della visibilità del **layout\_planner** sarà possibile mostrare o nascondere determinati input usati per le transazioni pianificate

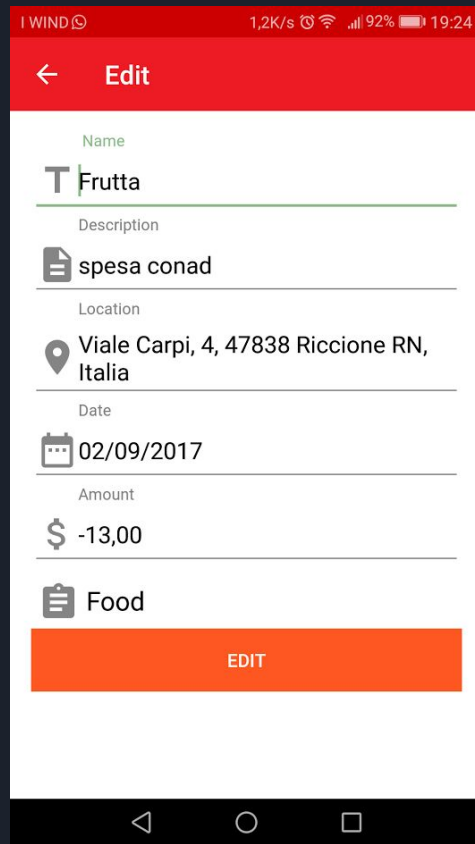
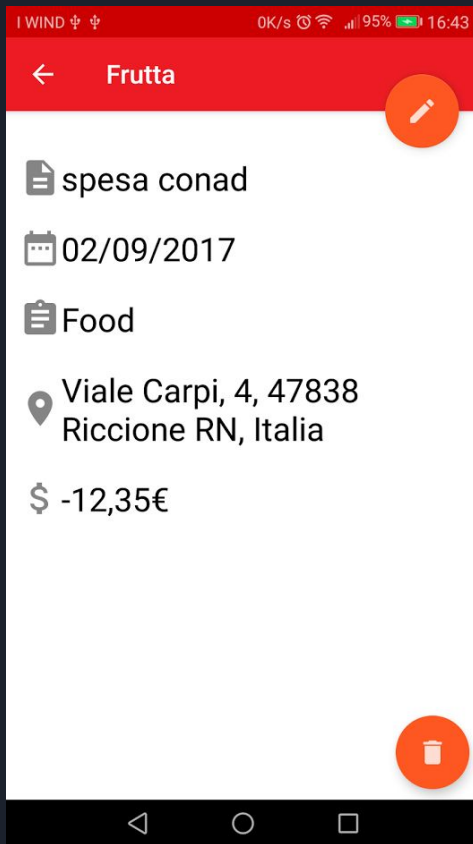
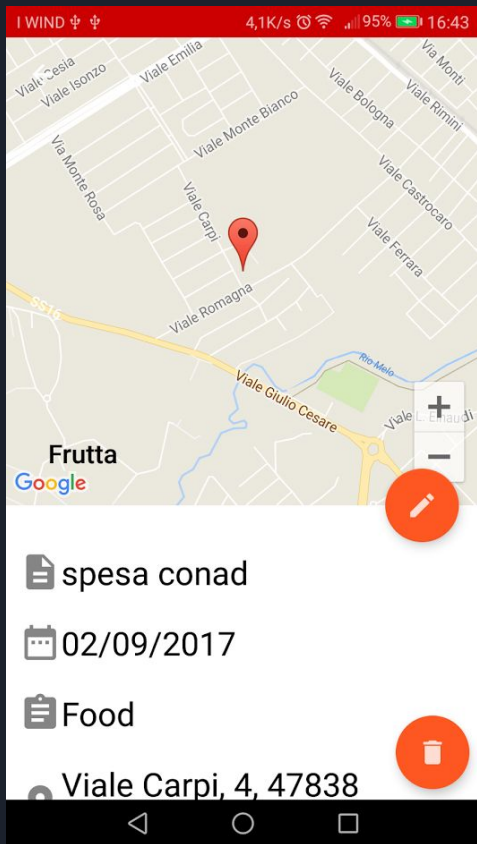


# NUOVO ELEMENTO - CODICE

Vengono effettuati i controlli sugli input, e se l'elemento è una transizione normale viene aggiunto in una tabella a parte ( **MoneyItem** ) altrimenti se è un evento pianificato avrà più dati e verrà aggiunto nella tabella **PlannedItem**

```
if (OK) {
    locid = dbHelper.getDaoSession().insert(loc);
    if (isPlanned) {
        PlannedItem pi = new PlannedItem(null, name, description, date, amount, catid, locid, occurrence_type, repeat);
        dbHelper.getDaoSession().insert(pi);
    } else {
        MoneyItem mi = new MoneyItem(null, name, description, date, amount, catid, locid);
        dbHelper.getDaoSession().insert(mi);
    }
    Toast.makeText(NewItemActivity.this, "Added", Toast.LENGTH_LONG).show();
    Intent intent = new Intent(NewItemActivity.this, MainActivity.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(intent);
} else {
    Toast.makeText(NewItemActivity.this, "Please fill all input", Toast.LENGTH_LONG).show();
}
```

# DETTAGLI E MODIFICA















# DETTAGLI E MODIFICA - LE FUNZIONALITÀ

- Attraverso questa pagina sarà possibile visualizzare tutti i dettagli di una transizione
- Visualizzare la locazione della transizione in una mappa interattiva (Google maps fragment)
- Cliccando sull'icona Edit si aprirà una finestra che permetterà di modificare tutti i campi della transizione
- Cliccando sull'icona Delete si potrà rimuovere dal database la transizione, e la locazione associata ad essa

# DETTAGLI E MODIFICA - LAYOUT

La pagina Dettagli è formata da una **collapsingToolbar** che contiene la mappa, e sottostante ad essa sono presenti le **TextView** per mostrati i vari dettagli della transizione

Nota: Il LAYOUT per una transizione e per una transizione periodica pianificata è lo stessa! basta nascondere e visualizzare il `LinearLayout` "planned\_layout"

```
▼  main_content (CoordinatorLayout)
  ▼  appbar (AppBarLayout)
    ▼  collapsing_toolbar (CollapsingToolbarLayout)
       map (<fragment>) - com.google.android.gms.maps.SupportMapFragment
       toolbar_detail
    ▼  NestedScrollView
      ▼  LinearLayout (vertical)
        Ab detail_description (TextView)
        Ab detail_date (TextView)
        Ab detail_category (TextView)
        Ab detail_position (TextView)
        Ab detail_amount (TextView)
      ▼  planned_layout_detail (LinearLayout) (vertical)
        Ab textView_planned_detail - "@string/planned_textview"
        Ab detail_planned_occurence (TextView)
        Ab detail_planned_repeat (TextView)
        Ab detail_planned_nextDate (TextView)
      +  fab_delete (FloatingActionButton)
      +  fab_edit (FloatingActionButton)
```



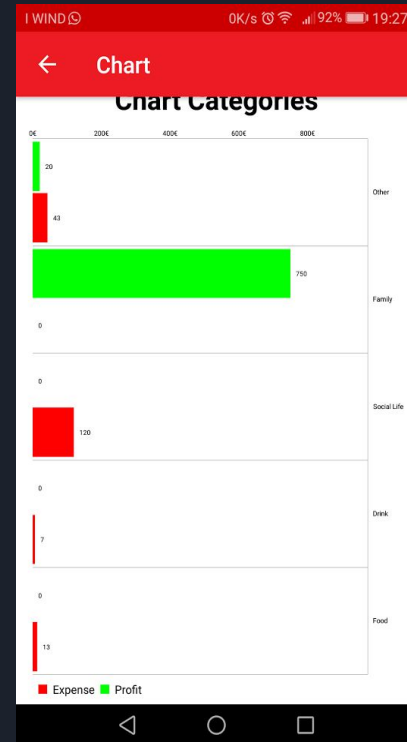
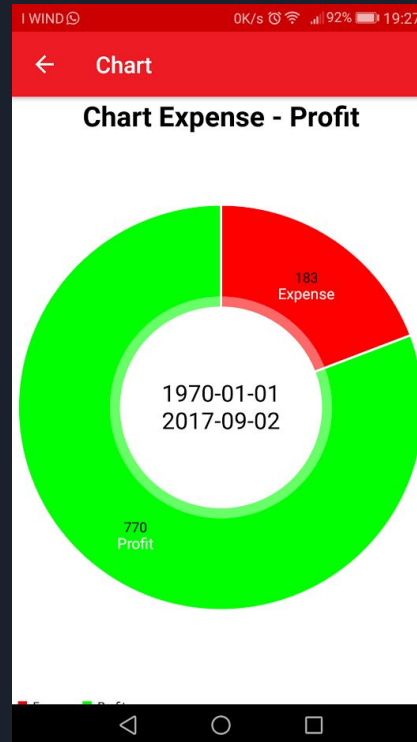
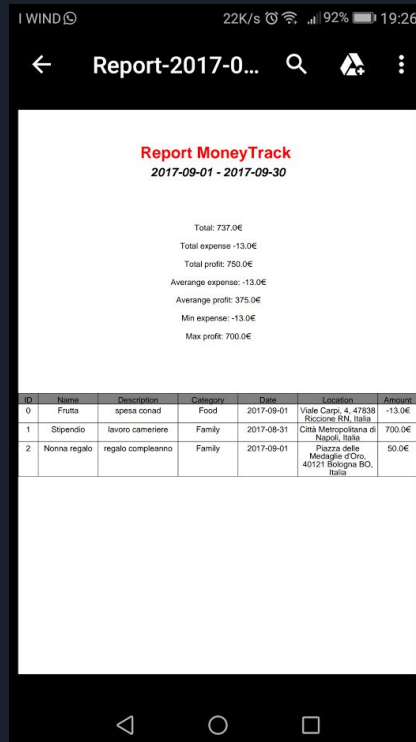
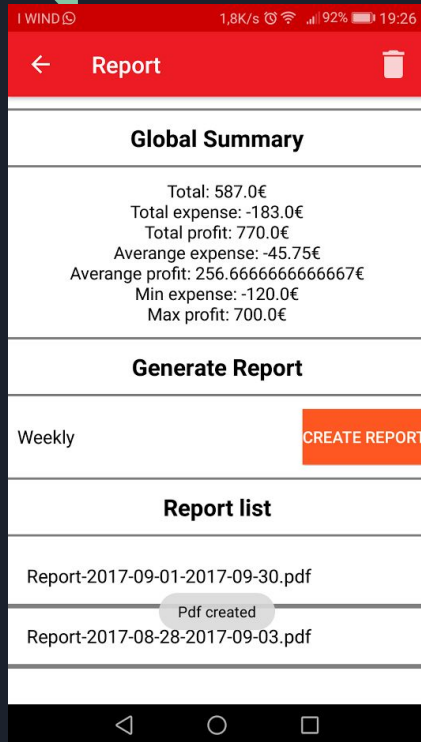
## DETTAGLI E MODIFICHE - CODICE

```
dbHelper = new DBHelper(this);
isPlanned = (Boolean) getIntent().getExtras().getSerializable("planned");
layout_planned = (LinearLayout) findViewById(R.id.planned_layout_detail);

if (isPlanned) {
    planned_item = (PlannedItem) (getIntent().getSerializableExtra("planned_item"));
    layout_planned.setVisibility(View.VISIBLE);
} else {
    money_item = (MoneyItem) (getIntent().getSerializableExtra("money_item"));
    layout_planned.setVisibility(View.GONE);
}
```

All'interno della funzione **onCreate()** viene fatta distinzione se l'elemento da visualizzare è pianificato oppure no!

# REPORT E GRAFICI





# REPORT E GRAFICI - LE FUNZIONALITÀ'

- Visualizzare le entrate/uscite globali e altre informazioni utili
- Generare e rimuovere **report** PDF giornalieri, settimanali, mensili, annuali, contenenti una serie di informazioni utili sulle transizioni e una tabella con la lista di tutte le transizioni
- Generare e visualizzare un **grafico** a torta sul totale dei profitti e delle perdite perdite, il grafico che mostra il totale dei profitti e spese per ogni categoria



# REPORT E GRAFICI - LAYOUT

## Report Layout

Per la visualizzazione dei grafici è stata utilizzata la libreria **MPAndroidChart**

```
▼ III LinearLayout (vertical)
  ▼ □ AppBarLayout
    ■ toolbar_report
  ▼ □ scrollView1
    ▼ III LinearLayout (vertical)
      🔗 horizontal_line (<include>) - @layout/horizontal_line
      Ab textViewSummary - "@string/global_summary"
      🔗 horizontal_line (<include>) - @layout/horizontal_line
      Ab textView_report
      🔗 horizontal_line (<include>) - @layout/horizontal_line
      Ab textView2 - "@string/generate_report"
      🔗 horizontal_line (<include>) - @layout/horizontal_line
    ▼ III LinearLayout (horizontal)
      ≡ report_spinner (Spinner)
      OK report_button (Button) - "@string/create_report"
      🔗 horizontal_line (<include>) - @layout/horizontal_line
      Ab textView_report_list - "@string/report_list"
      🔗 horizontal_line (<include>) - @layout/horizontal_line
      ≡ pdf_report_list (ListView)
```

```
<com.github.mikephil.charting.charts.PieChart
    android:id="@+id/show_chart_profitexpense"
    android:layout_width="match_parent"
    android:layout_height="500dp"
    android:layout_marginBottom="3dp"
    android:layout_marginTop="3dp" />

<TextView
    android:id="@+id/textView_chart_detail_category"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/chart_categories"
    android:textAlignment="center"
    android:textSize="24sp"
    android:textStyle="bold" />

<com.github.mikephil.charting.charts.HorizontalBarChart
    android:id="@+id/show_chart_category"
    android:layout_width="match_parent"
    android:layout_height="500dp"
    android:layout_marginBottom="3dp"
    android:layout_marginTop="3dp" />
```



# REPORT E GRAFICI - CODICE

## REPORT

Per creare report PDF vengono eseguiti i seguenti passaggi:

- Richiesta al database per avere la lista delle transazioni
- Creazione della cartella contenente i report
- Creazione del report
- Aggiunta dei paragrafi e tabelle necessarie a mostrare le informazioni

Libreria utilizzata: **itextpdf**

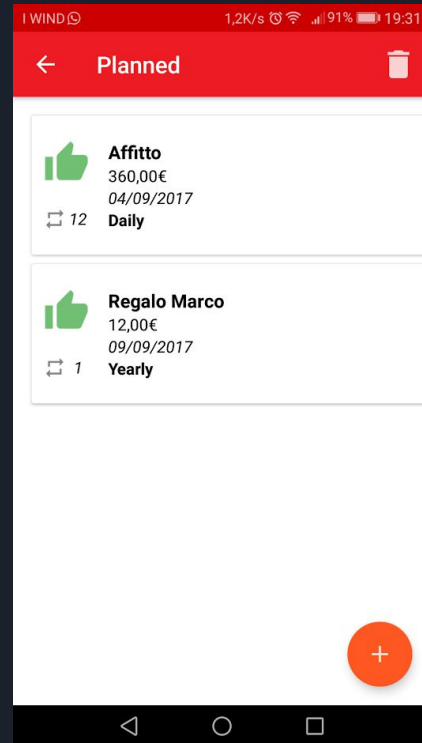
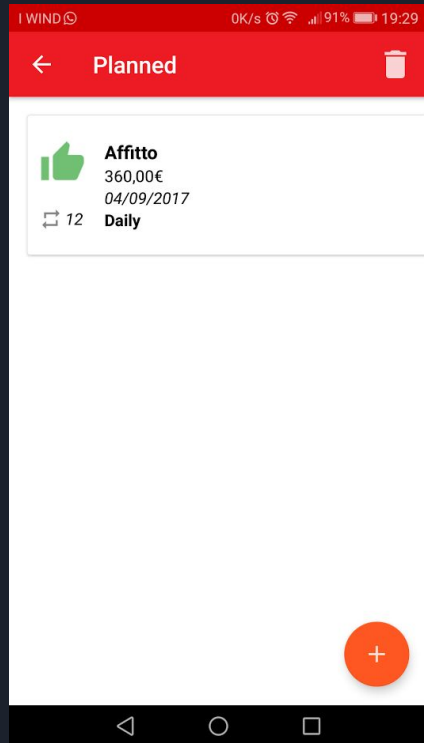
## GRAFICI

Per creare i grafici vengono eseguiti i seguenti passaggi:

- Richiesta al database per avere il totale dei profitti-spese totali, e dei profitti e spese divisi per categoria
- Inizializzazione del grafico a torta, e del grafico a barre
- Aggiunta dei dati al grafico

Libreria utilizzata: **MPandroidchart**

# PIANIFICAZIONI E RICORRENZE





# PIANIFICAZIONI E RICORRENZE

## - LE FUNZIONALITÀ

- Aggiungere una transizione **ricorrente**, specificando il numero di occorrenze ( repeat ) e il tipo di ricorrenza ( giornaliera, settimanale, mensile, annuale ), quando la data della transizione corrisponderà alla data del dispositivo, la transizione verrà aggiunta nella home, aggiornando il conto
- Aggiungere una transazione **pianificata**, semplicemente impostando la ricorrenza ( repeat ) ad 1
- Ricevere una notifica qualche giorno prima riguardo un evento pianificato o ricorrente

# PIANIFICAZIONI E RICORRENZE LAYOUT

Il layout è composto da una **Toolbar**, e una **RecyclerView** contenente tutte le card degli elementi pianificati o ricorrenti, ordinati per data crescente

```
▼ [Icon] CoordinatorLayout
  ▼ [Icon] LinearLayout (vertical)
    ▼ [Icon] AppBarLayout
      [Icon] toolbar_planned
    ▼ [Icon] planned_scroll (ScrollView)
      [Icon] recyclerview_planned
    [Icon] fab_add_planned (FloatingActionButton)
```

# PIANIFICAZIONI E RICORRENZE

## CODICE

- **PlannedItemAdapter** ( Adapter per RecyclerView )
- **DBHelper** ( facilita la connessione, e le query con il database SQLite )
- **MoneyReminder** ( **BroadcastReceiver** per gestire e controllare la data delle transazioni pianificate e gestire le notifiche )
- **ReminderService** ( servizio background che interagisce con MoneyReminder )

```
dbHelper = new DBHelper(context);
PlannedItem p = dbHelper.popPlanned();

if (p != null) {
    prefs = PreferenceManager.getDefaultSharedPreferences(context);
    int reminder_times = Integer.parseInt(prefs.getString("notification_reminder", "1"));
    LocalDate current_date = new LocalDate();

    if (p.getDate().getTime() <= current_date.toDate().getTime()) {

        // convert planneditem to ==> moneyitem
        MoneyItem mi = new MoneyItem(null, p.getName(), p.getDescription(), p.getDate(), p.getAmount());
        dbHelper.getDaoSession().getMoneyItemDao().insert(mi);
        PlannedNotifyUser(context, "MoneyTrack Transiction added", p.getName() + " planned item, was

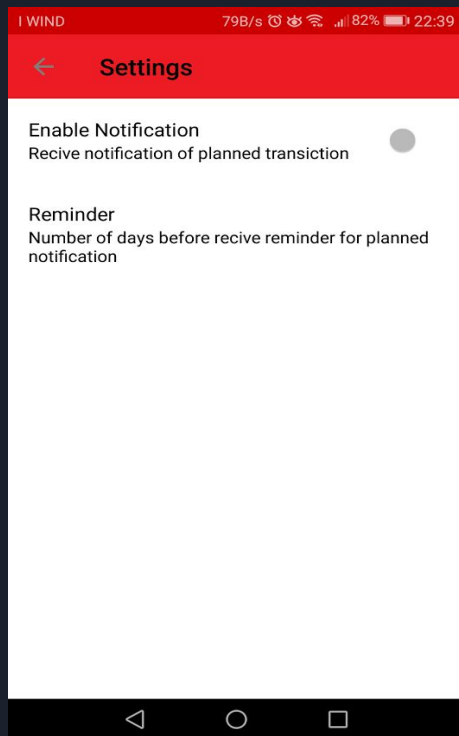
        // decrease repeat from planneditem
        int repeat = p.getRepeat();
        p.setRepeat(repeat - 1);
        dbHelper.getDaoSession().update(p);

        // if planned item repeat is 0 delete it
        repeat = p.getRepeat();
        if (repeat == 0) {
            dbHelper.getDaoSession().delete(p);
        } else {
            //update planneditem with new planned_date
            p.setDate(createPlannedDate(p.getOccurrence(), p.getDate()));
            dbHelper.getDaoSession().update(p);
        }

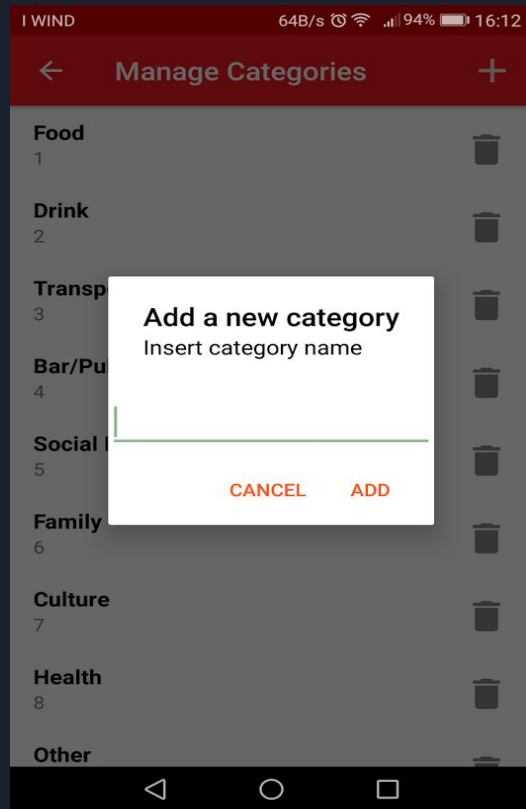
        checkPlanned(context);
    } else if (p.getDate().getTime() <= current_date.plusDays(reminder_times).toDate().getTime()) {
        PlannedNotifyUser(context, "MoneyTrack Reminder", p.getName() + "\n" + p.getDate());
    }
}
```

# ALTRE PAGINE

## IMPOSTAZIONI



## CATEGORIE



## ARCHIVIO





# ALTRE INFORMAZIONI UTILI

## DIPENDENZE

- com.android.support:appcompat-v7:25.3.1'
- com.android.support:support-v4:25.3.1
- com.android.support:palette-v7:25.3.1"
- com.android.support:design:25.3.1
- com.android.support.constraint:constraint-layout:1.0.2
- com.android.support:recyclerview-v7:25.3.1
- com.android.support:cardview-v7:25.3.1
- com.google.android.gms:play-services-maps:11.0.4
- com.google.android.gms:play-services-location:11.0.4
- Com.github.PhilJay:MPAndroidChart:v3.0.1
- org.greenrobot:greendao:3.2.2
- net.danlew:android.joda:2.9.9
- com.itextpdf:itextpdf:5.5.10

## DATABASE

- Category ( categoryID, name)
- Location ( locationID, name, longitude, latitude)
- MoneyItem ( id, name, description, date, amount, categoryID, locationID)
- PlannedItem ( id, name, description, date, amount, categoryID, locationID, occurrence, repeat )

## REPO GITHUB

<https://github.com/alewin/moneyGtracking>



# LIBRERIE ESTERNE

- [MPAndroidChart](#)
  - A powerful Android chart view / graph view library, supporting line- bar- pie- radar- bubble- and candlestick charts as well as scaling, dragging and animations.
- [greenDAO](#)
  - greenDAO is a light & fast ORM solution for Android that maps objects to SQLite databases.
- [itextpdf](#)
  - Core Java Library + PDF/A, xtra and XML Worker
- [JodaTime](#)
  - Joda-Time is the widely used replacement for the Java date and time classes prior to Java SE 8
- [GooglePlayService](#)
  - app can take advantage of the latest, Google-powered features such as Maps, Google+, and more

FINE