Запуск проекту

Для роботи з проектом необхідно встановити середовище розробки Android Studio.

Детальна інструкція установки написана тут: <http://web-academy.com.ua/stati/14-stati/63-ustanovka-android-studio-instruktsiya>

Потім потрібно імпортувати проект з ресурсу GitHub в Android Studio. Детально процес описано за посиланням <https://maxrohde.com/2014/08/18/import-github-project-to-android-studio/> .

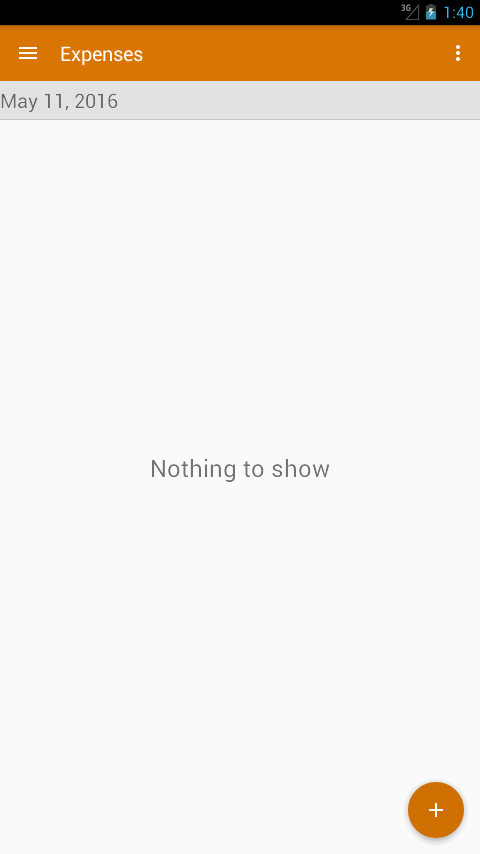
Посилання на Git репозиторій: <https://github.com/OleksandrPelevin/Expenser.git>

Якщо не виходить налаштувати Git, то потрібно завантажити zip-файл за посиланням <https://github.com/OleksandrPelevin/Expenser/archive/master.zip>, розархівувати та відкрити в Android Studio, вибравши в стартовому меню "Open an existing project" і папку з проектом. Після побудови Gradle потрібно вибрати в стрічці меню Run – Run 'app' або натиснути Shift-F10. Після цього потрібно обрати девайс, на якому буде запущено додаток. Проект працює на пристроях з Android 4.1.2 (API 16) та вище.

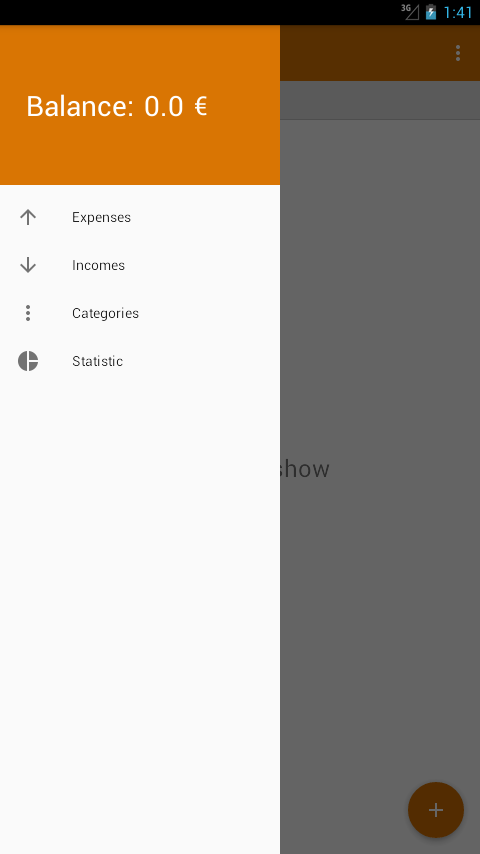
Інструкція користувача

При запуску додатку користувач потрапляє на екран зі списком витрат.

Якщо це перший запуск програми або користувач ще не додавав витрат, то на екрані буде повідомлення **"Nothing to show"** – немає що відображати.



Для навігації по додатку потрібно використовувати бокове меню, що висувається жестом "свайп" при проведенні пальцем з лівого краю екрану до правого.



На боковому меню відображається поточний баланс та пункти меню, доступні для користувача.

Нижче описані пункти меню:

**Expenses** (Витрати) – відображає список витрат за певну дату.

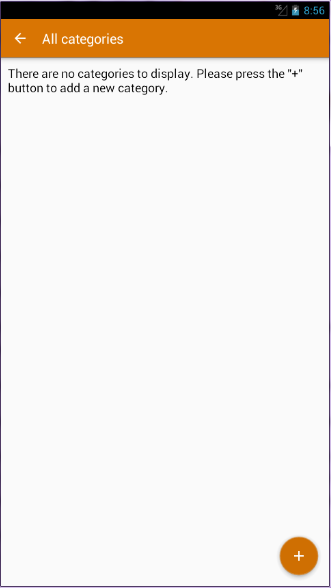
**Incomes** (Прибутки) – відображає список доходів за певну дату.

**Categories** (Категорії) – відображає список категорій транзакцій.

**Statistics** (Статистика) – статистика доходів та витрат користувача по категоріям з можливістю вибору часового проміжку, за який відображається статистика.

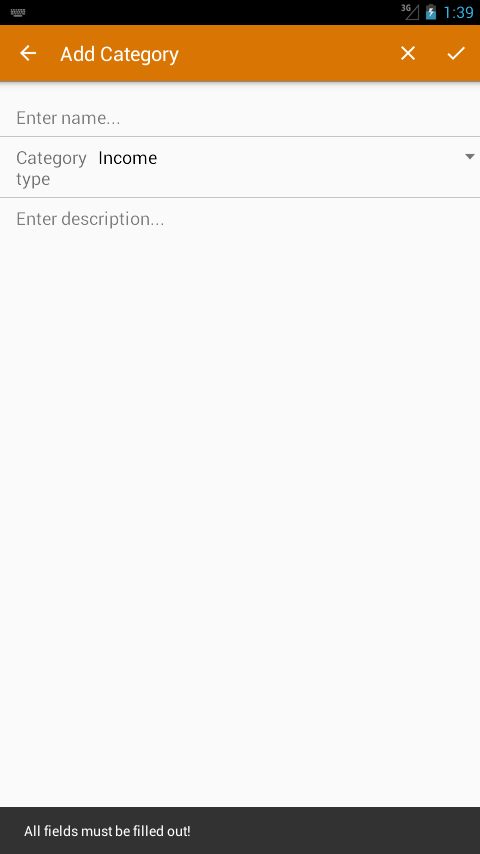
Для додавання нової транзакції потрібні заздалегідь створені категорії витрат чи доходів.

Саме тому потрібно вибрати пункт **Categories.** Відкриється зображений нижче екран.



Тепер потрібно додати нову категорію транзакцій. Для цього потрібно натиснути на кнопку .

Після натиску на кнопку відкриється нижче зображений екран.



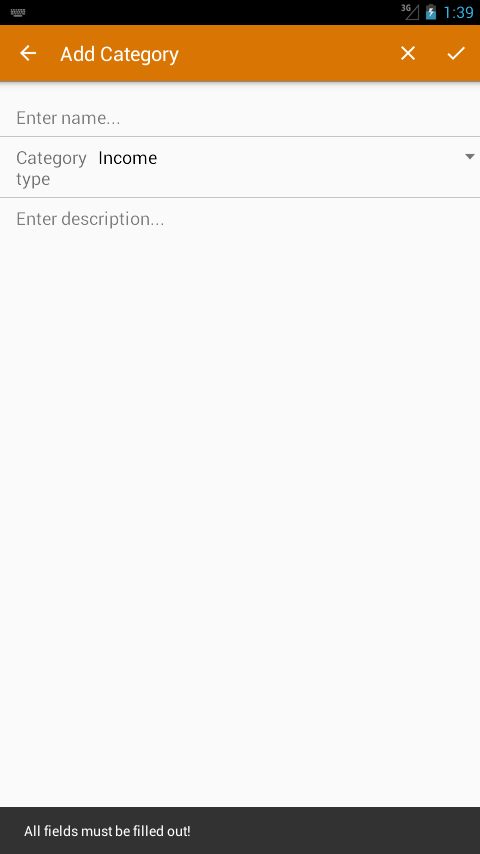
Для того щоб створити нову категорію, у цьому вікні необхідно заповнити всі поля. Із випадаючого списку можна вибрати тип категорії: **Income** (Прибуток), **Expense** (Витрата).

Для того, щоб очистити поля форми, необхідно натиснути на знак .

Якщо Ви не бажаєте додавати нову категорію, то потрібно натиснути на знак.

Для того, щоб зберегти категорію після заповнення всіх полів, потрібно натиснути на знак .

Якщо не всі поля будуть заповнені і Ви натиснете на знак , Вас буде повідомлено про помилку, як зображено нижче. Ви не зможете зберегти категорію, якщо не заповните всі поля форми.

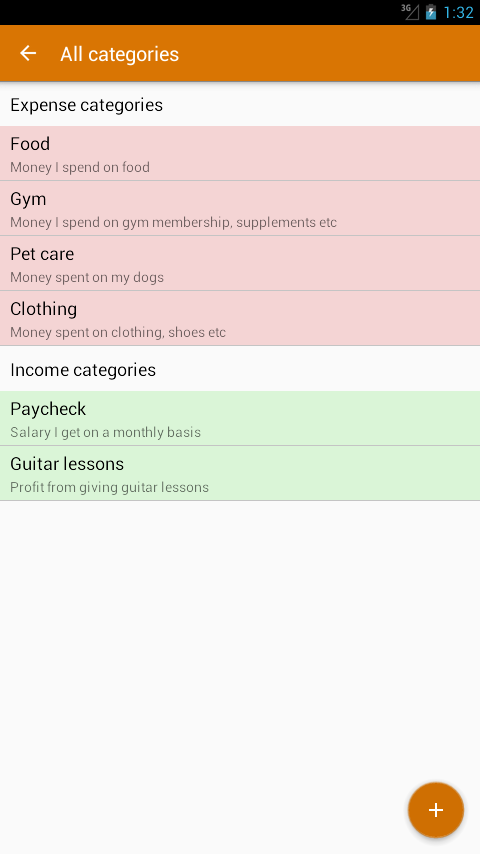


Повідомлення має такий зміст: "**All fields must be filled out**" – всі поля повинні бути заповненими.

Після заповнення всіх полів, потрібно натиснути на знак .

Якщо збереження категорії було успішним, Ви знову опинитесь на екрані зі списком категорій. Ви відразу ж зможете побачити новостворену категорію.

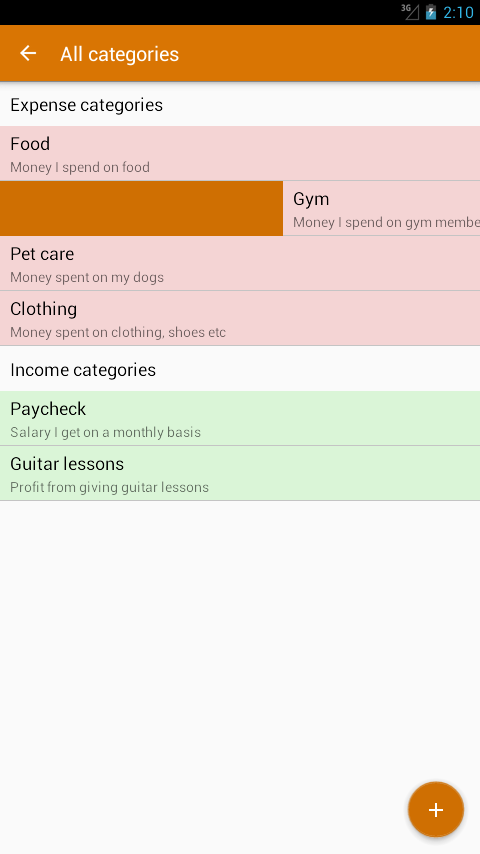
Категорії витрат в списку відображаються світлочервоним кольором, доходів – світлозеленим. Нижче наведено приклад заповненого списку категорій.



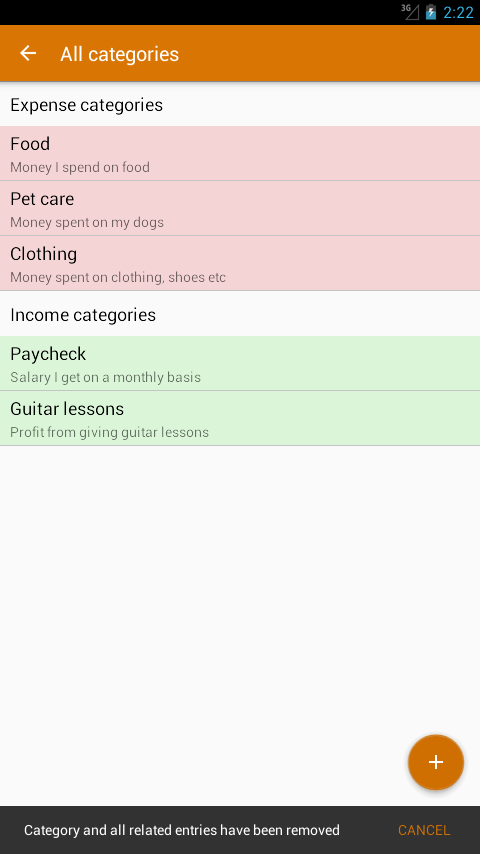
Для того, щоб редагувати категорію, потрібно клікнути на неї. Відкриється форма для редагування категорії, як зображено нижче. **Тип категорії змінити не можна.**



Для того, щоб видалити категорію, потрібно виконати жест "свайп" зліва направо над категорією, яку Ви бажаєте видалити. Цей процес зображено нижче.

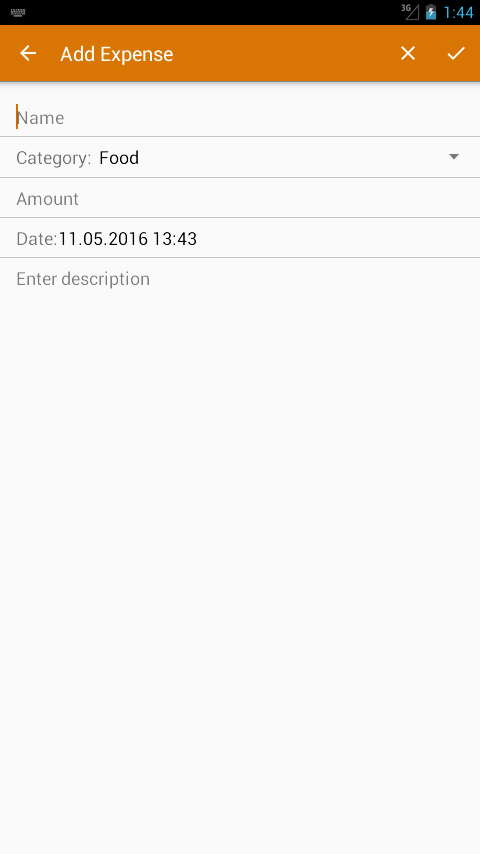


**При видаленні категорії видаляються всі транзакції в ній!** Після виконання "свайпу" буде показано повідомлення про те, що категорію видалено. Якщо Ви хочете відмінити видалення категорії, натисніть на надпис "**CANCEL"**.



Для того, щоб повернутися на головний екран, потрібно натиснути на стрілку .

Для додавання нової транзакції потрібно натиснути на кнопку  в нижньому правому кутку екрану, коли ви знаходитесь на екрані зі списком доходів або витрат. Відкриється форма для створення нової транзакції, як показано нижче.



В формі потрібно **обов'язково ввести** **назву транзакції**, вибрати **категорію**, до якої слід її віднести та **суму**.

Якщо Ви додаєте витрату, то в випадаючому списку категорій будуть доступні, відповідно, категорії витрат, якщо додаєте дохід – категорії прибутку.

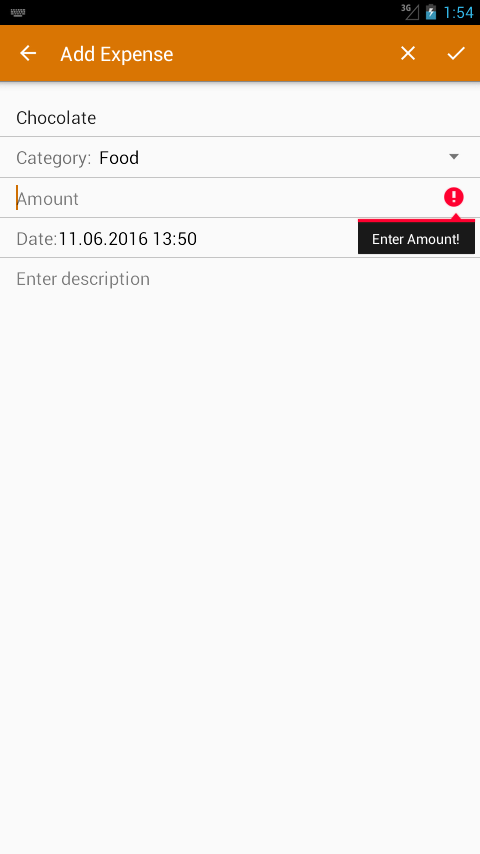
Сума не може бути нулем. Дата та час за замовчуванням встановлюються згідно з моментом натиснення на кнопку , але при бажанні можна вибрати іншу дату. **Опис транзакції може бути пустим.**

Для того, щоб очистити поля форми, необхідно натиснути на знак .

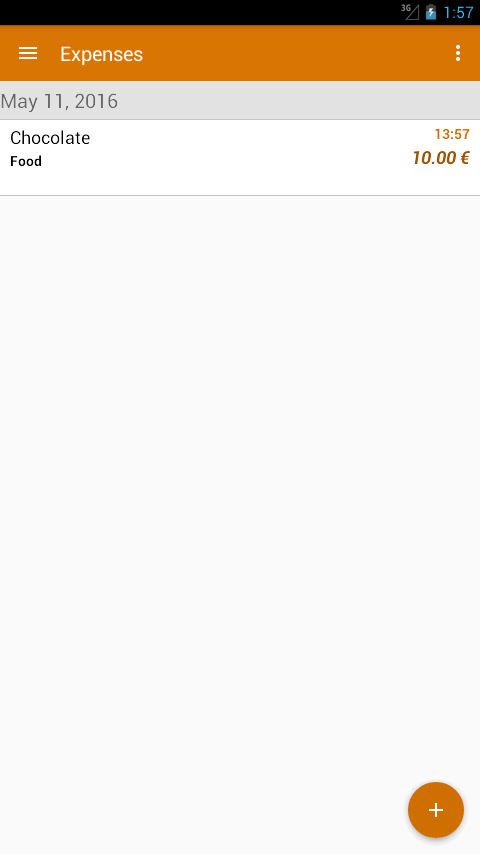
Якщо Ви не бажаєте додавати нову транзакцію, то потрібно натиснути на знак, щоб повернутися до списку транзакцій.

Для того, щоб зберегти транзакцію після заповнення всіх необхідних полів, потрібно натиснути на знак .

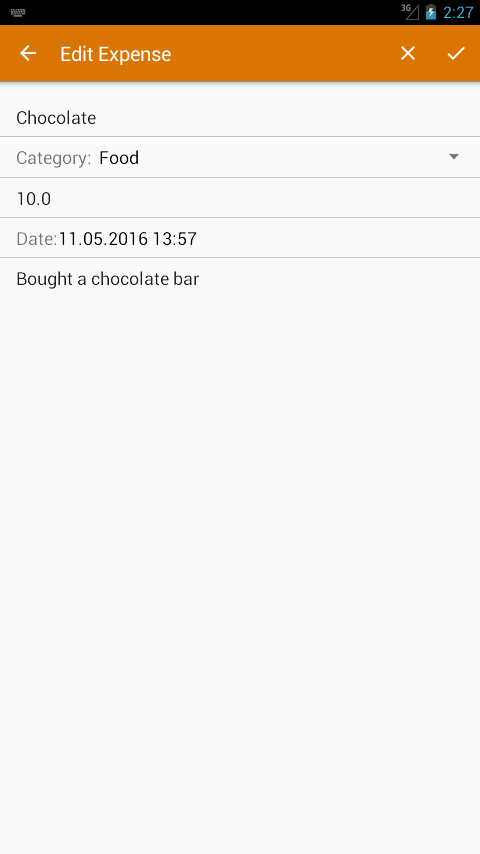
Якщо не всі необхідні поля будуть заповнені і Ви натиснете на знак , Вас буде повідомлено про помилку, як зображено нижче. Ви не зможете зберегти транзакцію, якщо не заповните всі необхідні поля форми.



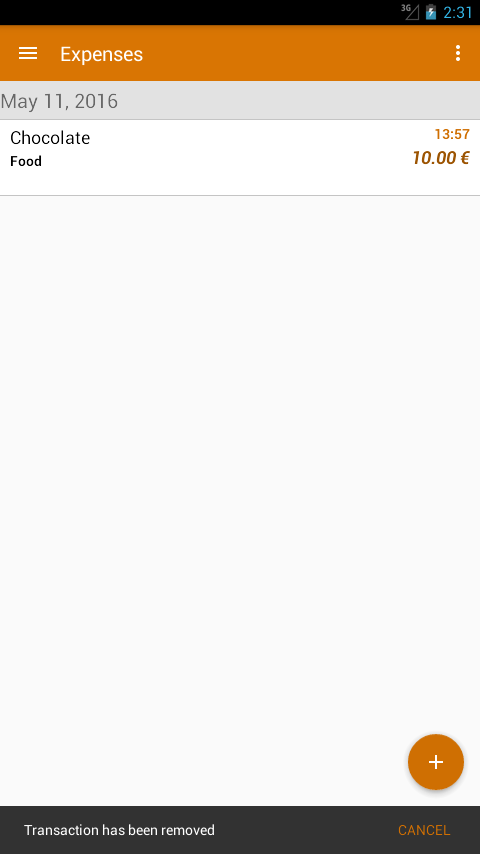
Якщо транзакція успішно збереглась, Ви знову опинитесь на екрані зі списком транзакцій. Ви відразу ж зможете побачити збережену транзакцію.



Для того, щоб редагувати транзакцію, потрібно натиснути на неї в списку. Відкриється форма для редагування транзакції, як зображено нижче. Ви можете змінити значення будь-якого поля.

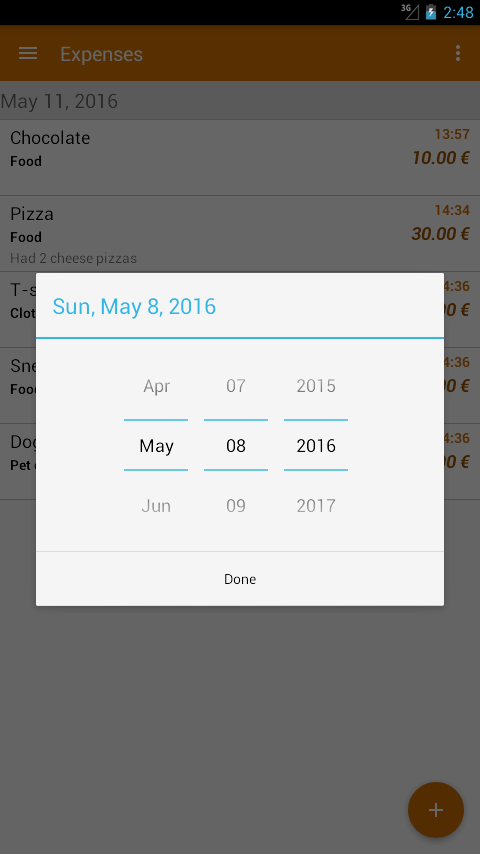


Для того, щоб видалити транзакцію, потрібно виконати жест "свайп" зліва направо над транзакцією, яку Ви бажаєте видалити. Після виконання "свайпу" буде показано повідомлення про те, що транзакцію видалено. Якщо Ви хочете відмінити видалення транзакції, натисніть на надпис "**CANCEL"**. Результат виконання "свайпу" зображено нижче.

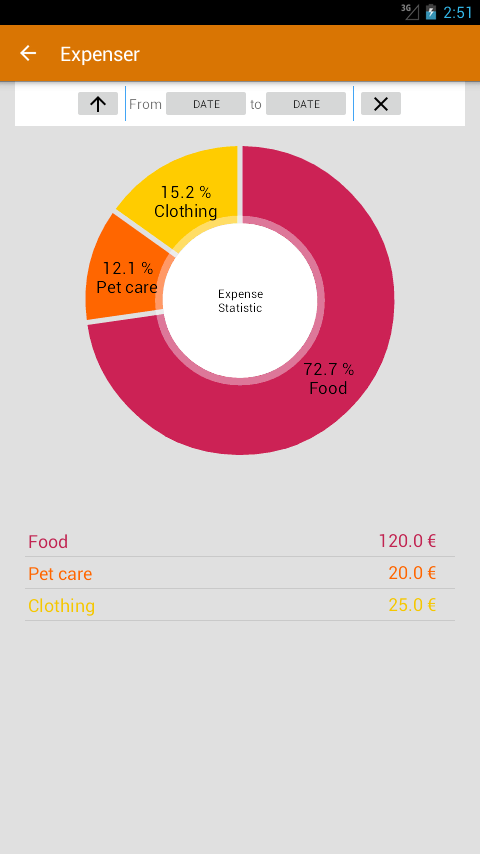


Процес додавання доходу аналогічний, тільки потрібно спочатку перейти на список доходів, вибравши пункт **Incomes** в боковому меню, та натиснути кнопку .

Ви можете переглядати список витрат/доходів за будь-який день, натиснувши на вкладку з датою. В діалоговому вікні Ви зможете обрати дату, за яку Ви хочете переглянути транзакції.



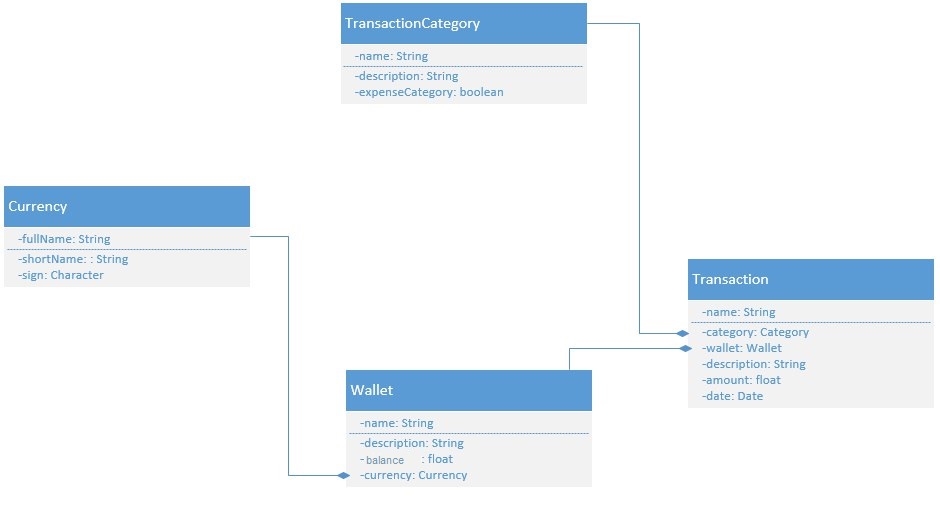
Для того, щоб переглянути статистику, потрібно в боковому меню вибрати пункт "Statistics", як показано нижче.



За замовчуванням показується статистика витрат. Для того, щоб переглянути статистику доходів, потрібно натиснути на знак . Він здійснює перемикання між статистиками доходів та витрат. В полях  Ви зможете встановити часовий проміжок, за який Ви хочете переклянути статистику. Для того, щоб повернутися на головний екран, потрібно натиснути на стрілку .

Опис класів

### com.github.sinapple.expenser.model

В цьому пакеті розташовані моделі даних. Нижче зображена схема моделі даних.

Кожен клас моделі наслідує клас SugarRecord для використання функціоналу Sugar ORM.

Кожен клас також має один пустий конструктор, що є вимогою Sugar ORM.

Моделі можуть мати декілька конструкторів, тому що об'єкти моделі можуть бути створені при різній кількості даних.

Кожен клас моделі має набір гетерів та сетерів для доступу до його полів та метод toString(); вони не будуть описані нижче, тому що не потребують детального опису.

## public class Wallet

extends SugarRecord

Клас-представлення гаманця.

Кожна транзакція повинна бути пов'язаною з певним гаманцем. У гаманця є назва, опис (необов'язковий), баланс та валюта, в якій зберігається баланс. За замовчуванням існує лише один гаманець.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **public static** Wallet getCurrentWallet() | Повертає об'єкт класу Wallet за замовчуванням, що існує лише в одному екземплярі. Якщо об'єкту не існує, створює його. |

## public class Currency

extends SugarRecord

Клас-представлення валюти, що використовується в транзакціях. Валютою за замовчуванням є євро.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **public static** Currency getDefaultCurrency() | Повертає об'єкт класу Currency за замовчуванням, що існує лише в одному екземплярі. Якщо об'єкту не існує, створює його. |

## public class TransactionCategory

extends SugarRecord

Клас-представлення категорії транзакцій.

Категорія може бути пов'язаною або тільки з витратами, або тільки з прибутком. Булеве поле **mExpenseCategory** дозволяє визначити тип категорії. Якщо поле має значення **true**, то категорія пов'язана лише з витратами, а **false** - з прибутком.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **public boolean** equals(Object obj) | Перевіряє рівність двох об'єктів категорій транзакцій. Якщо об'єкти рівні, повертає true, в іншому випадку – false. |

## public class MoneyTransaction

extends SugarRecord

Клас-представлення транзакції.

Обов'язковими параметрами для створення об'єкту транзакції є назва, об'єкт категорії транзакцій, до якої відноситься транзакція, об'єкт гаманця, з яким буде пов'язана транзакцію та сума.

Транзакція може представляти або прибуток, або витрату. Якщо транзакція є витратою, то поле **mAmount** має від'ємне значення, якщо прибутком - додатнє.

Для визначення типу транзакції використовується метод **isExpense(),** що повертає **true**, якщо транзакція є витратою, **false** - доходом.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **public boolean** isInDate(Calendar date) | Перевіряє, чи була транзакція здійснена певної дати. |
| **public boolean** isInDateInterval(Calendar begin, Calendar end) | Перевіряє, чи чи була транзакція здійснена в певному часовому інтервалі. |
| **public static** List<MoneyTransaction> findTransactionByDate(Calendar date, **boolean** lookForExpenses) | Повертає список транзакцій, що були здійснені певної дати. |
| **public static** List<MoneyTransaction> findTransactionByDate(Calendar first, Calendar second, **boolean** lookForExpenses) | Повертає список транзакцій, що були здійснені в визначеному інтервалі дат. |
| **public static** List<MoneyTransaction> findTransactionByTime(Calendar first, Calendar second, **boolean** lookForExpenses) | Повертає список транзакцій, що були здійснені в визначеному часовому інтервалі. |
| **private static void** setTimeToZero(Calendar date) | Обнуляє час для того, щоб додати весь день до інтервалу. Використовується в методі findTransactionByDate(Calendar date, **boolean** lookForExpenses). |

### com.github.sinapple.expenser

## public class NewTransactionActivity

extends AppCompatActivity

Клас активності для додавання та редагування транзакцій.

Ця активність може отримувати інтент з інформацією стосовно дії, яка буде виконуватись.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **private void** dataFillingDb() | Наповнює поля старими даними, якщо транзакція редагується. |
| **private void** editTransaction() | Зберігає змінену транзакцію. |
| **private void** showSpinner(List<TransactionCategory> categories) | Ініціалізує список категорій, до яких можна віднести транзакцію. |
| **private long** saveTransaction() | Зберігає нову транзакцію та повертає її ID. |
| **private int** getIndex(String category) | Встановлює в формі вибраною категорію, до якої належить транзакція, що редагується. Використовується в методі dataFillingDb(). |
| **private void** isEmpty() | Перевіряє, чи заповнені необхідні поля перед збереженням транзакції. |
| **public void** onDateSet(DatePicker view, **int** year, **int** monthOfYear, **int** dayOfMonth) | Викликається після того, як користувач обрав дату в діалоговому вікні. |

## public class CategoryActivity

extends AppCompatActivity

Клас активності для перегляду списків категорій витрат та категорій доходів.

В цій активності ініціалізуються 2 списки для категорій витрат та категорій доходів. До кожного списку прив'язується адаптер типу CustomCatListAdapter, "слухач" кліків користувача RecycleItemClickListener та об'єкт-обробник жестів користувача RecyclerViewItemCallback.

## public class CustomCatListAdapter

extends RecyclerView.Adapter<CustomCatListAdapter.ViewHolder>

implements RecyclerViewItemCallback.ItemTouchHelperAdapter, RecycleItemClickListener.OnItemClickListener

Клас, що відповідає за створення View-компонентів для списку RecyclerView, де відображається інформація про категорії транзакцій.

## public class CustomRListAdapter

extends RecyclerView.Adapter<CustomRListAdapter.ViewHolder>

implements RecyclerViewItemCallback.ItemTouchHelperAdapter, RecycleItemClickListener.OnItemClickListener

Клас, що відповідає за створення View-компонентів для списку RecyclerView, де відображається інформація про транзакції.

## public class MainActivity

extends AppCompatActivity

implements NavigationView.OnNavigationItemSelectedListener, DatePickerDialog.OnDateSetListener

Основна активність додатку, в якій відображається список транзакцій та забезпечується навігація по додатку за допомогою NavigationDrawer.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **private static void** initializeDB() | Ініціалізує базу даних за замовчуванням. |
| **private void** changeActivityAppearance(**int** flag) | Змінює зовнішній вигляд активності в залежності від того, що відображається – список витрат чи доходів. |
| **private void** notifyIfEmptyList() | Відображає текстове повідомлення для користувача, якщо відсутні транзакції для відображення. |
| **private void** addItemToList(MoneyTransaction transaction) | Метод для оновлення списку транзакцій у випадку додання нової. |
| **private void** addAllToList(List<MoneyTransaction> list) | Метод для оновлення списку транзакцій при виборі дати, за яку відображаються транзакції. |
| **private** MoneyTransaction removeItem(**int** index) | Метод для оновлення списку транзакцій при видаленні транзакції. |
| **public void** setDateFieldValue() | Встановлює поточну дату. |
| **public void** onDateBoxClick(View view) | Ініціалізує діалогове вікно для вибору дати при кліку на вкладку для вибору дат. |
| **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) | Оновлює список згідно з результатами роботи NewTransactionActivity. |
| **public void** onItemDismiss(View messageOutput, **final int** position) | Викликається в результаті виконання жесту "свайп" над елементом списку. |
| **public void** onItemClick(View view, **int** position) | Викликається при натисненні на елемент списку та запускає активність NewTransactionActivity для редагування вибраної транзакції. |
| **public void** onDateSet(DatePicker view, **int** year, **int** monthOfYear, **int** dayOfMonth) | Викликається після того, як користувач вибрав дату в діалоговому вікні, що було викликано методом onDateBoxClick(View view) та оновлює список транзакцій згідно з обраною датою. |
| **public boolean** onNavigationItemSelected(MenuItem item) | Оброблює натиснення на пункти бокового меню. |

## public class NewCategoryActivity

extends AppCompatActivity

Клас активності для додавання та редагування категорій транзакцій.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **private void** editCategory() | Зберігає зміни в категорії транзакцій. |
| **private void** saveCategory() | Створює нову категорію транзакцій. |
| **private boolean** isInputValid() | Перевіряє, чи всі обов'язкові поля для редагування/створення категорії заповнені. Інформує користувача про помилку за допомогою віджету SnackBar. |
| **private boolean** isEmpty(EditText t) | Перевіряє, чи текстове поле є пустим. |

## public class RecycleItemClickListener

extends Object

implements RecyclerView.OnItemTouchListener

Клас, що перехоплює сенсорний натиск на елемент RecyclerView, надає інтерфейс для обробки цієї події. Надає інтерфейс-адаптер OnItemClickListener для взаємодії з іншими класами.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **public boolean** onInterceptTouchEvent(RecyclerView rv, MotionEvent e) | Метод для визначення елементу списку, на який натиснув користувач. |

## public class RecyclerViewItemCallback

extends ItemTouchHelper.Callback

Клас, що калібрує жест "свайп" для кожного елемента списку RecyclerView та забезпечує належне оформлення цієї події. Також включає в себе колбек для виклику потрібних методів адаптеру.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **public void** onSwiped(RecyclerView.ViewHolder viewHolder, **int** direction) | Метод, що викликається після закінчення жесту "свайп". |
| **public int** getMovementFlags(RecyclerView recyclerView, RecyclerView.ViewHolder viewHolder) | Встановлює напрямок, в якому дозволений "свайп". |
| **public void** onChildDraw(Canvas c, RecyclerView recyclerView, RecyclerView.ViewHolder viewHolder, **float** dX, **float** dY, **int** actionState, **boolean** isCurrentlyActive) | Замальовує фон елементу списку під час свайпу. |

## com.github.sinapple.expenser.statistic

У цьому пакеті знаходяться класи для маніпуляцій з статистикою та її відображення у вигляді діаграм.

## public class StatisticActivity

extends AppCompatActivity

Клас, що ініціалізує статистичну діаграму та необхідні для її роботи елементи та дані.

## public class StatisticAdapter

extends ArrayAdapter

Клас-адаптер даних для списку з статистичними даними.

## public class StatisticItem

extends Object

Клас-сутність елементу списку статистики.

## public class SDataCreator

extends Object

Клас, що формує списки статистичних даних для побудови графіків.

## public class PieChartManager

extends Object

Клас-менеджер діаграм, побудованих на основі статистичних даних.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **public void** initPieChart() | Метод, що ініціалізує кругову діаграму. |
| **private void** initXVals() | Ініціалізує значення осі X в діаграмі, тобто назви категорій транзакцій. |
| **private void** initYVals() | Ініціалізує значення осі Y в діаграмі, тобто загальні суми категорій транзакцій. |
| **private void** generateCenterSpannableText(String text) | Ініціалізує надпис в центрі діаграми, тобто "Expense statistic" якщо відображаються витрати та "Income statistic" якщо выдображаються прибутки. |

## public class ColorsListCreator

extends Object

Клас, що створює список кольорів для використання в діаграмі статистики.

|  |  |
| --- | --- |
| Метод | Опис методу |
| **public** List<Integer> getColors() | Метод, що повертає палітру кольорів, створену в конструкторі класу. |