# Developer manual

## Change Log

| Ver. | Date | Change |
|---|---|---|
| 1.0 | 2012-10-01 | Document creation |
| 1.1 | 2012-10-14 | Added UML. Added info about layering and adjusted the manual because of the refactoring . Added instructions for ant builds. |
| 1.2 | 2012-10-15 | Added instructions for git cloning and submodules. |

## Getting the source code

1. Fetch the git repository from https://github.com/daubigne/Android-Budget-Project

```
git clone https://github.com/daubigne/Android-Budget-Project
```

2. Activate included library projects

```
cd Android-Budget-Project

git submodule update --init
```

3. Install the latest Android SDK into Eclipse. Download Android API level 8 and 16. Create a new Android (Virtual) Device. Make sure you are on a Java 6 SE development environment.

4. Import all included projects in the git folder into Eclipse. (Android_Budget_App , Android_Budget_App_Tests, android-viewflow)
5. Right click the viewflow project and choose “Build project”

6. Dig in!

## Structure

The application uses a Model-View-Controller structure. The classes are mainly divided into three packages; activity, controller and model.

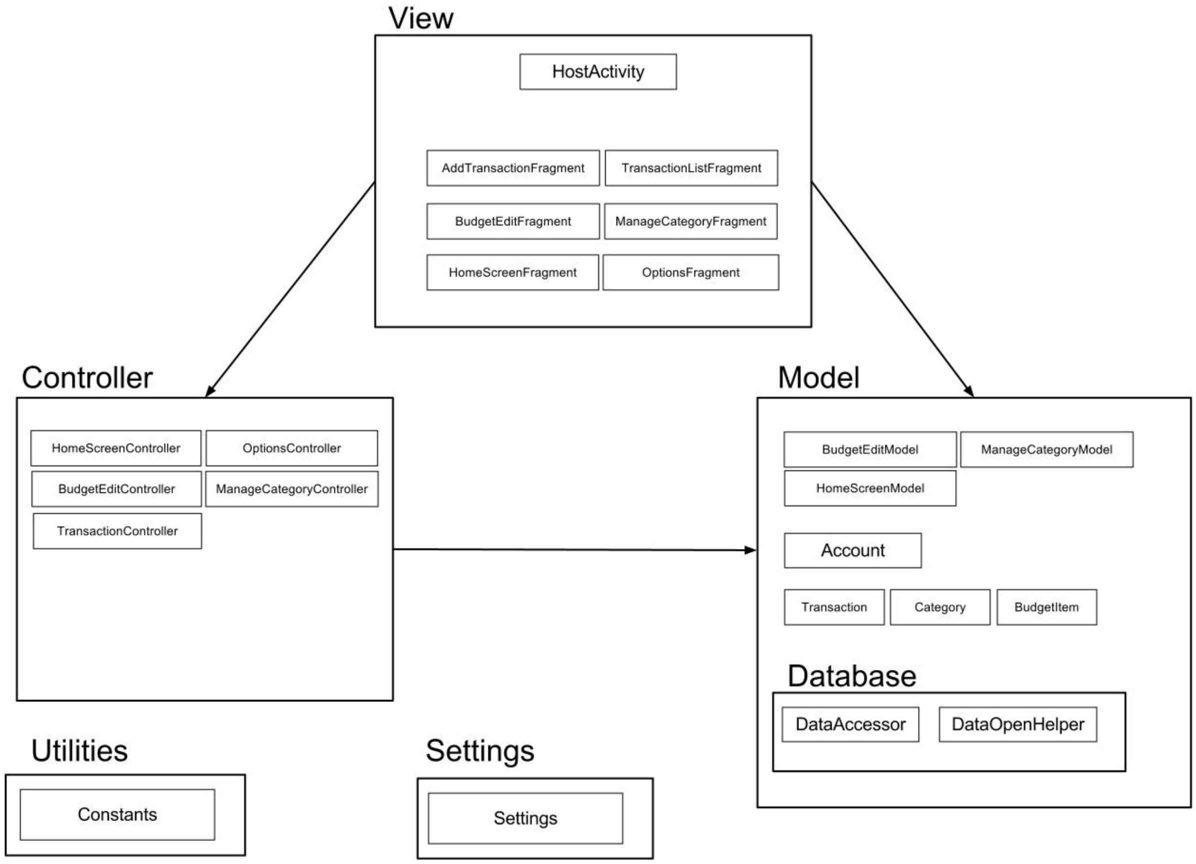
The activity package includes the host activity and fragment classes which represents the views and has a model which data it presents to the user, and a controller which it uses to change the state of the model . These activity classes responds to user input, as such they act as the gateway into the application.

The model classes represents the state of the application and always keeps the database updated.

```
         Activity
           |        \
     Controller---Model
                      |
                Database
```

The database package contains a SQLite database where all local data is stored and a DataAccessor class which the controllers uses to communicate with the database.

There is also a settings package containing a settings class which contains general settings for the application, some of which the user will be able to control, and a utilites package where things such as constants and other uncategorized classes are placed.



## The layers

The application consists of four layers; the view, the controller, the model and the database. The activities and fragments in the view layer has a layout.xml which it uses to present the graphics seen by the user. It responds to user input by sending the data provided by the user to the controller which processes the data and updates the model. The model then has DataAccessor class which uses queries to update the database. When the model is updated it fires a propertyChange to it’s listeners (the view) which then fetches the new data from the model and updates itself.

## Ant build

The source code comes bundled with a build.xml file for building the project from a command line. You will need to provide a local.properties file with the filepath to your android sdk. An example below:

```
sdk.dir=/home/user/android-sdk
```

Place this file in all Android project folders (Android_Budget_App, Android_Budget_App_Tests, android-library-projects/android-viewflow/viewflow) folder

To run an ant build with JUnit tests and coverage report use the following command while residing in the Android_Budget_App_Tests folder

```
ant clean emma debug install test
```