University of Namur

INFOB318: personnal project

See U festival programmer guide

Project workers :

Gonzague Yernaux
Stéphane leblanc

Teacher:
Vincent Englebert

Assistant:
Maouaheb Belarbi

Contents

1 Introduction

This document is the programmer guide of the See U festival application. This app is an Android application that has been created to provide a line up and other functionalities for the participants of the See U festival.

2 Developpers

Stephane Leblanc worked on the development of See U festival.

3 Development environment

This application has been developed on Android Studio with the Java language. Android Studio is the standard to develop Android applications and can be downloaded here : Android Studio

4 How to compile

To compile the project on Android Studio, you just have to click on the green arrow like in image below. Make sure you have an android smartphone connected to your computer or set up a virtual device.

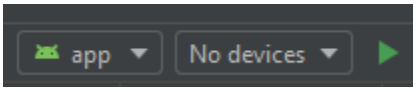

Figure 1: How to compile

5 How to generate APK

To make an executable version of your app, you have to generate an APK. To do that, go to "Build" and click on "generate signed APK"

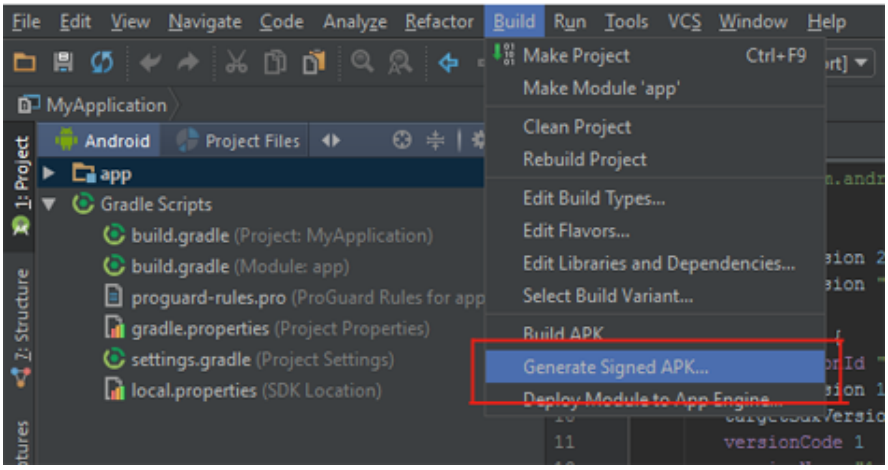

Figure 2: Generate APK

Then choose the APK option and click Next. If you don't have set up a key, click on "Create new" and fill in the different forms. Choose then this key to generate apk, click Next, choose release build and V2 (full APK signature) and click on finish. You executable version is now build in AndroidStudioProjects/MyApp/app/release.

6 Navigation

The app has been developed with a navigation drawer and tabs linked to the HomeFragment to provide a better navigation in the line up. Then, there's a main activity that gathers the fragments of the drawer and each tab of the home fragment is also a fragment. See the figure 3 to look at relations between activities and fragments.

7 Architecture

The architecture of the code follows the classics of the Android development. In the Java package, you can find the Activity, Fragments, ui (elements of the drawer) and utils packages. Some activities extends an Activity class that gathers the different methods common to these activities. See the figure 7 to look a the activities diagram class.

7.1 Activities package

In this package, you can find an activity class for each artist of the festival and a settings activity that extend the Activity class. The MainActivity is the activity loaded when the app starts. It gathers the different fragments of the drawer contained in the ui package. The MapActivity is an activity that has to be developed in the future.

7.2 Fragments and ui package

In the Fragments package, you can find the different fragments used for the tabs in the HomeFragment. Each fragment is named with the stage and the day. In the the ui package, you can find the fragment linked with the MainActivity that build the drawer.

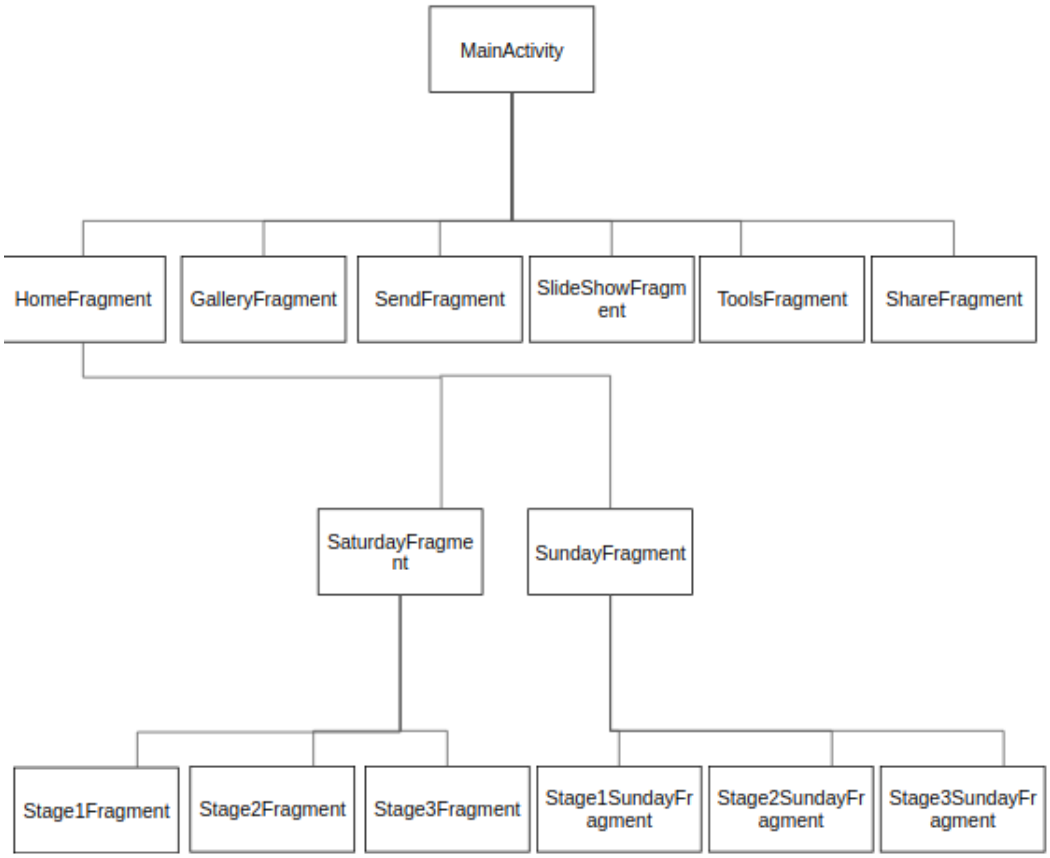

Figure 3: Relations between Activities and fragments

7.3 Utils package

This package contains the classes used to build the line up and to manage the settings of the app's locale with LocaleHelper.

7.3.1 CustomListView class

This class helps to create a list view with a list of artists. Then, the constructor builds a list view with an array for the image's id of the artists, an array for the schedules of the artists, an array for the names of the artists and an array to say if each artist is on air or not. The context is the Activity context. In the getView method, for each values of the different arrays, it takes a value at a common index (position parameter) to build an element of the list view. It can apply it thanks to the ViewHolder class. So, in each fragment that constitutes a tab of the HomeFragment, a CustomListView object is instantiated.

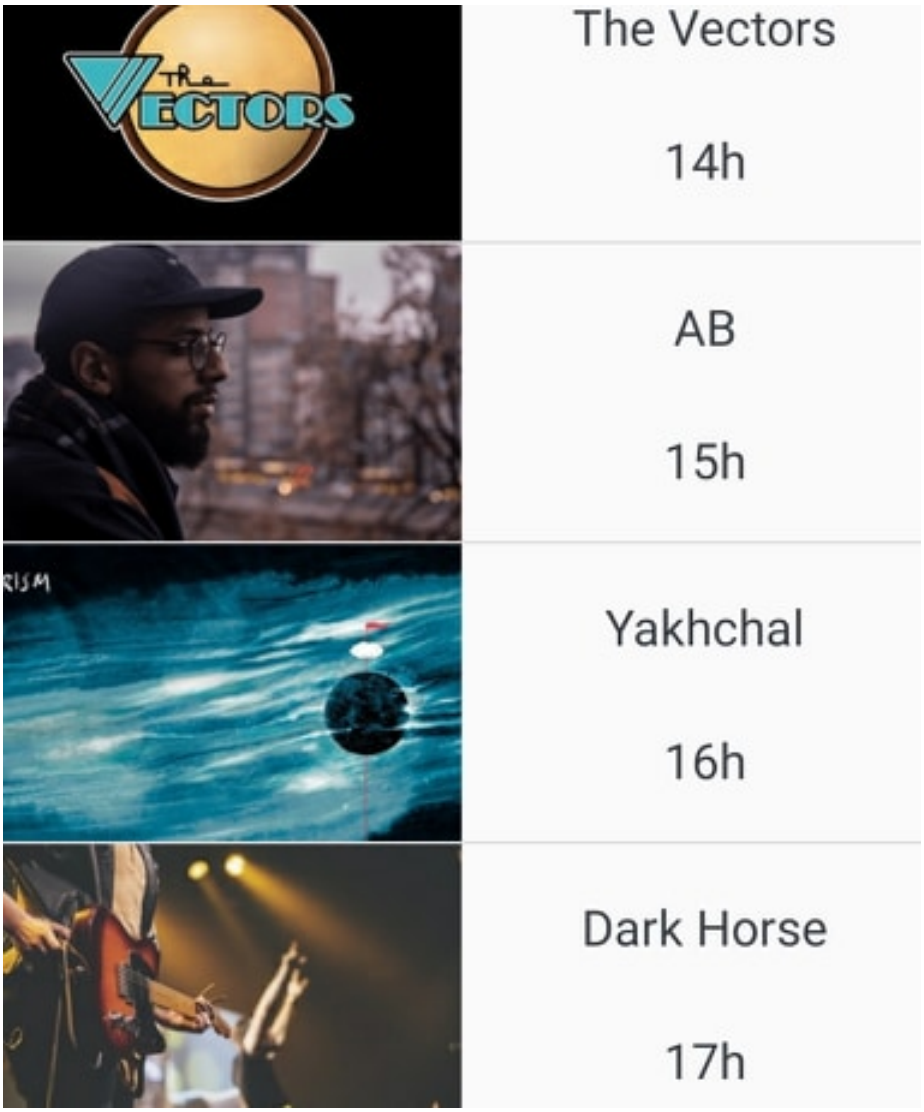

Figure 4: List view example

This layout is an example. Here, the layout of the list view is defined in mylist.xml.

7.3.2 LocalHelper class

The LocalHelper class is a class to manage the settings of the app's locale. It's then used by the PreferenceFragment to change the language of the app.

7.3.3 ViewPagerAdapter class

This class is used to build the tabbed navigation in the HomeFragment. A ViewPager is an Android pattern that provides an horizontal navigation. In a ViewPager, there are many fragments that each manage a page. It then extends the FragmentStatePagerAdapter that is an Android package that handles saving and restoring of fragment's state.
Then, in the HomeFragment, it instantiates a ViewPager and a TabLayout, the TabLayout is attached to the ViewPager. A ViewPagerAdapter is instantiated to handle the states of the fragments. Saturday and Sunday fragments are linked to the HomeFragment and it sets the adapter on the ViewPager. The same thing is done for the Sunday and Saturday fragments, it links each 3 fragments to make a tabbed navigation at many levels like in the image below.

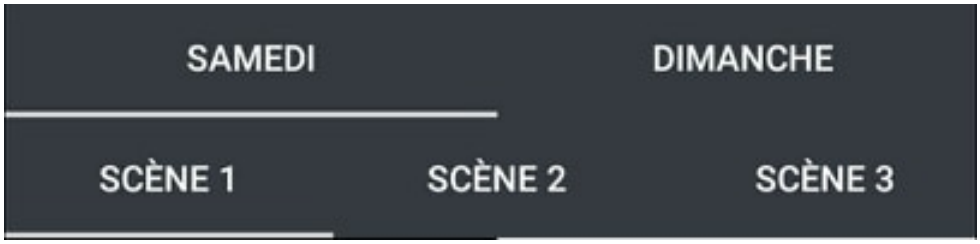

Figure 6: Tabbed navigation

7.4 XML part

To have more information about the XML part of the project, we invite you to check the programmer guide of Ticked-it. It's pretty the same thing except that here, as we are in a navigation drawer, there's an xml file to list the item in the drawer (activity_main_drawer.xml) and a xml file that lists the different fragments that can be loaded from the drawer (mobile_navigation.xml).

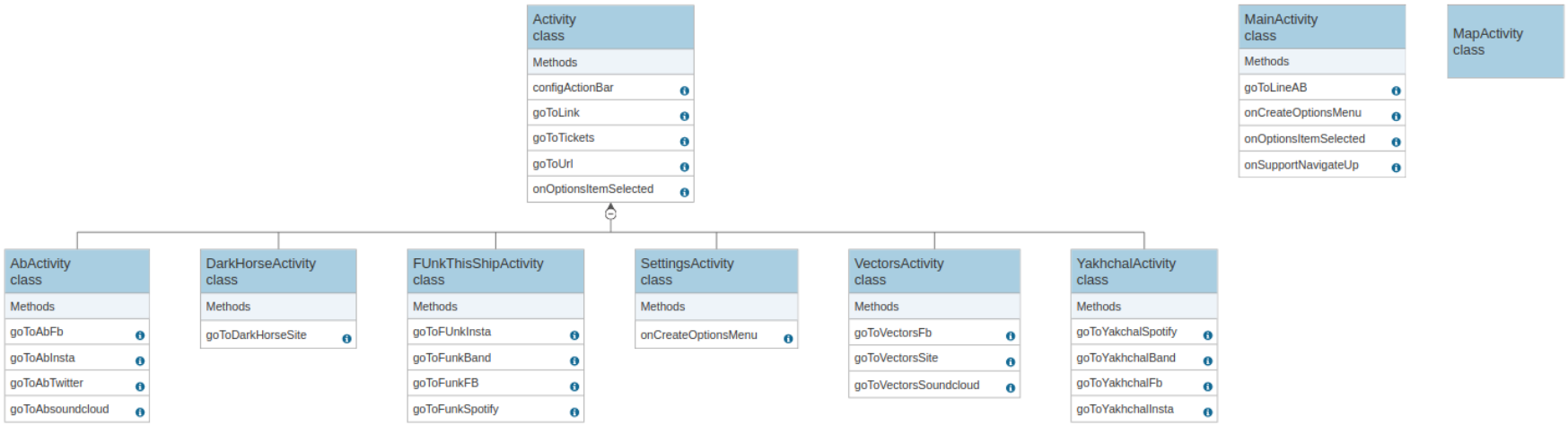

Figure 5: Activities class diagram