B5- Dév. Mobile : Java

B-PAV-562

Epicture

A universal photo finder/browsing app

v1.2

EPITECH.
L'ECOLE DE L'INNOVATION ET DE L'EXPERTISE INFORMATIQUE

Epicture

A universal photo finder/browsing app

binary name:	epicture.apk
repository name:	Java_epicture_$YEAR
repository rights:	ramassage-tek
language:	Java
group size:	2-3
compilation:	via gradle

Your repository must contain the totality of your source files, but no useless files (binary, temp files, class files...)

The goal of this project is to use and implement online photo sharing API platforms.
You must create a photo finder and browsing application for the Flickr and Imgur platforms.

The different aspects of Android mobile development applications are to be taken into account during the creation of your project.

You will be evaluated on the management of build and your solution's test, on the developed functionalities and the user interface and experience your application has to offer.

Your application must be developed in Java (Webviews and NDK Android usage prohibited).
Your application must be compatible with the Android SDK API 19.
Your application must comply with the project's build constraints.
Your application must offer the project's functionalities that are detailed in the project description.
Your application must offer a high-quality, polished user interface and experience.
You're free to implement bonuses functionalities of your choosing.
You must set up a test strategy for your project.
You must provide documentation for your project.

It's up to you to choose the libraries you want in order to meet the project's needs (network, data etc.)

Build application

Your project must use `Gradle` as its build automation system.
The builds must be launched by using the `Gradle Wrapper` tool in the command line.

At the very least, the *assembleDebug* task could be called in order to build your application's APK with a debugging key.

Extra points will be given to projects that handle the *assembleRelease* task call to build your application's APK with your private key.

You can add extra tasks of your choosing (for the test coverage, for example).

Functionalities

You must create a photo finder and browsing application.

The project's functionalities are set out as follows:

- `Flickr` and `Imgur` API implementation
- *Connecting* to the `Flickr` and `Imgur` platforms
- The photo *display* put online by the user connected to `Flickr` and `Imgur`
- `Flickr` and `Imgur` photo *finder*
- *Uploading* photos to `Flickr` and `Imgur`
- Adding/deleting photos to/from your *favorites*
- Managing photo display *filters*

Don't limit yourselves to these suggestions. Feel free to add extra functionalities.

Read the API literature for Flickr and d'Imgur.

User interface and experience (UI/UX)

The user experience that your application offers is an aspect that is just as important as its functionalities.
You should consider some ideas for the experience that you would like to offer to your application's users, and how the interface you will create will connect with this experience.

Each mobile application is built according to its own set of rules and expectations. Navigation ergonomics, facility and simplicity are all non-exhaustive points that should be taken into account.

This reflection is an integral part of your application's design process.

Your application's interface will be evaluated, just like the quality of the experience it offers to the user.
Here are a few elements that we will take into consideration during this evaluation:

- the choice and coherence of style (colors, icons, typography, etc.)
- the usage of space (size, placement, density, etc.)
- the choice of elements (buttons, lists, menus, etc.)
- your project's identity and the cohesion of it all

Consult the guidelines provided by the Android development teams.

Application test

As you know, any software project must be tested in order to make sure that it functions correctly and complies with its scope statement.
Within the Android environment, the large variety of smartphones, different sizes and capabilities and the different versions of Android to support are extra reasons why tests are important in your development process.

You must set up a test strategy for your application. The completeness of your strategy will be taken into account during the evaluation of your project.

`Android Studio` is designed to make testing easier and it offers two different types of tests: the *local unit tests*, which run on your local JVM with `JUnit` and the *instrumented tests*, which run on a device.

You're free to incorporate any test framework you want (`Mockito`, `Espresso`, `UI Automator`, etc.) in order to make your test strategy more complete.

Documentation

We require you to provide us with documentation that will illustrate the design and development phases of your project.

You will document the project's technical and functional specifications.
You will document the application's principles of ergonomics and navigation.
You can incorporate a few conception diagrams.
You can document your project's code.

The goal is to have a document that we can use as a tool in order to easily understand the project so that we can make communication between groups and gaining new skills for new developers easier.

Bonus

You are free to implement the bonus functionalities of your choice, under the condition that you comply with the minimum expectations of your application for the build management project, the test strategy, the functionalities and the UI/UX.

Here are a few non-exhaustive ideas:

- your application is *available in the Play Store*
- integration of *additional API elements*
- implementing an *additional API*
- an *iOS* version