B5 - Application Development

B-DEV-501

Epicture

A photo finder / browsing app

2.0

{EPITECH.}

Epicture

repository name: DEV_epicture_*$ACADEMICYEAR*
repository rights: ramassage-tek
language: Java/Kotlin (Android), C#/.NET (UWP), React Native

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.

General consideratoins

As part of this project, you will take on the role of a Software Architect.
Your main goal is neither reinventing the wheel nor writing numerous lines of code. On the contrary, your main action is to understand, select and integrate a wide range of existing libraries.
The code you write will only implement the so-called *business* logic. In other words, your main job will be to write *glue* between selected software components to complete the requested project.
Before embarking on the carrying out of such a project, we suggest you take the time to analyze and understand the operation of each software brick. In the idea, we will talk about **state of the art** and **POC**:

- **State of the art**: Study the different possible solutions and choose the right component according to what is needed.
- **POC** (*Proof Of Concept*): Make a quick demo program that proves the proper functioning of a component or algorithm.

Your active participation in the preparation of *Workshops* and your attention during these activities will be **essential** for the success of this project.

The project

The goal of this project is to use and implement online photo sharing API platforms.
You must create a photo finder and browsing application the following platform: Imgur.
The different aspects of mobile (or universal) applications development 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.

Development environments

For this project, you have the choice to use one of three development environment.
The only constraint is that you are prohibited to use the same environment as the previous project.

The goal here is to discover differents languages and environment that you will be able to use for the AREA project.

If the members of your group used different techological environments, all of these are prohibited for you.

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, respecting the best practices of the choosen platform.
You must code using the best practices of the environment you chose.
You're free to implement bonuses functionalities of your choosing.
You should set up a test strategy for your project.
You can 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.)

+ JAVA/KOTLIN: SPECIFIC DETAILS

If you choose to develop the project in the JVM environment on Android, you must respect the following details:

- Your application must be developed in Java or Kotlin (Webviews and NDK Android usage prohibited).
- Your application must be targeted at the Android SDK API 28.

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.

That means that a `build.gradle` file should be in the root of your repository.

+ C#/.NET: SPECIFIC DETAILS

- The project must be developed in C#
- The project must be built using a Visual Studio 2017 solution (available here)

That means that an `epicture.sln` file should be in the root of your repository.

+ REACT NATIVE: SPECIFIC DETAILS

- The project must be, at least, able to build for Android

Your project must use *Gradle* as its build automation system (to be built for Android).
The builds must be launched by using the *Gradle Wrapper* tool in the command line.

That means that an `build.gradle` file should be in the `android` directory at the root of your repository.

Functionalities

You must create a photo finder and browsing application.
The project's functionalities are set out as follows:

- *Imgur* API implementation
- Authenticate to the *Imgur* platform
- Display the photos put online by the user connected
- Search for photos on the plateform
- Upload photos to the plateform
- Manage your favorites
- Filter the displayed photos

Read the API literature for 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 technology you chose.

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

You're free to incorporate any test framework you want in order to make your test strategy more complete.

Documentation

It is important to take the time to define a simple architecture without code duplication and that is also scalable.
You are asked to provide clear and simple documentation for your project. You can integrate some design schemes: class diagram, sequence diagram …

The goal is to have a document that serves as a working tool to easily understand the project in order to facilitate communication in the work teams and facilitate the development of skills of new developers.
Thus, there is no need to make class or sequence diagrams of the entire project, but rather to choose the important parts to understand that need to be documented.