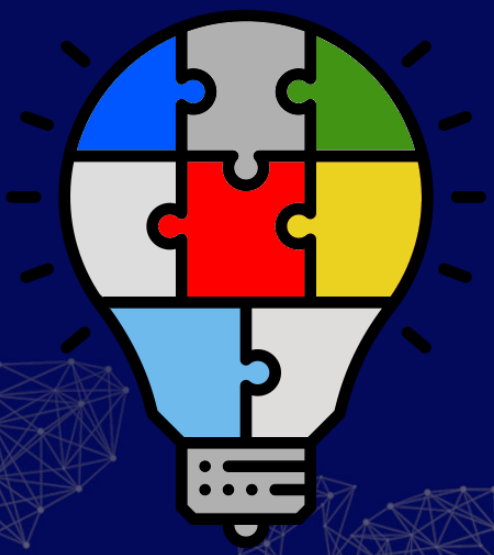
{EPITECH}
LEARN DIFFERENT*

TRELLTECH

PROJECT MANAGEMENT APP

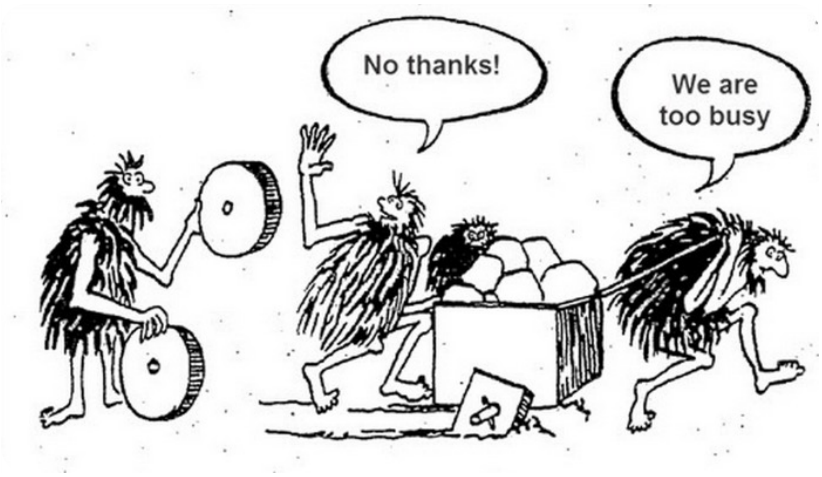


* apprendre autrement

TRELLTECH

The totality of your source files, except all useless files (binary, temp files, obj files,…), must be included in your delivery.
All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*

As a Software Architect, your main goal is to understand, select and integrate a wide range of existing libraries in order to perform various action without reinventing the wheel nor writing numerous lines of code.



Your code will be composed of two main parts:

- ✓ the first can be assimilated to *glue* in between of existing functional components ;
- ✓ the second part will be the so-called *business* logic.

These two things (and a bit more) should compose the requested project. However, before embarking on such a project, we suggest you take the time to:

- ✓ discover each framework and available libraries at your disposal (See the Development Environments section) ;
- ✓ understand the pros and cons of each framework ;
- ✓ analyze the API you'll have to implement ;
- ✓ choose the framework that suits you and the project best.

There are several codeless tools to request APIs and check the proper formatting of requests, responses status and data, … such as Postman, try it, it's worth it!

For this project, you are asked to create an application (mobile or desktop) using the Trello API.

Your application must:

- ✓ comply with the projet's delivery constraints ;
- ✓ offer the features detailed in the Mandatory Features section ;
- ✓ offer a high-quality, polished UX and UI ;
- ✓ respect the chosen framework best practices.

Your application should:

- ✓ come with a clear test strategy and be tested ;
- ✓ come with a technical documentation depending on your framework.

Your application could (AKA bonuses):

- ✓ come with a user documentation in the form of a file ;
- ✓ come with a user documentation in the form of an integrated tutorial ;
- ✓ have any other feature of your choosing not listed in the next section.

These are a few links concerning the Reddit API that you might find useful.

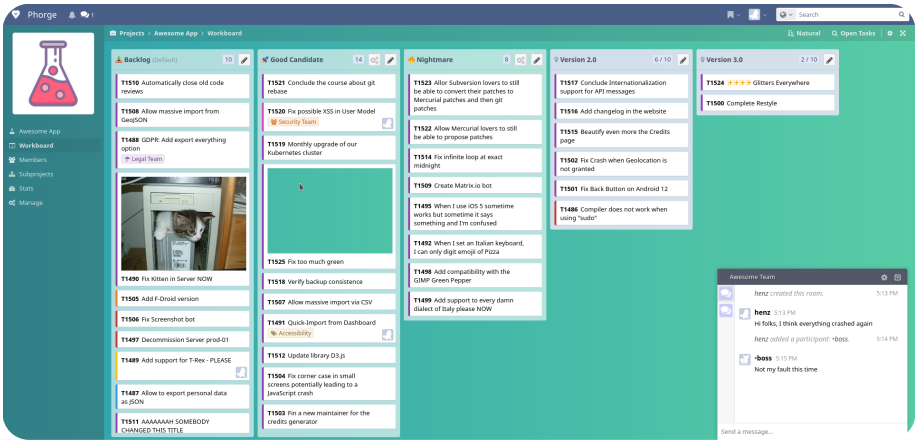
- ✓ API base url
- ✓ Endpoints documentation
- ✓ Token generation

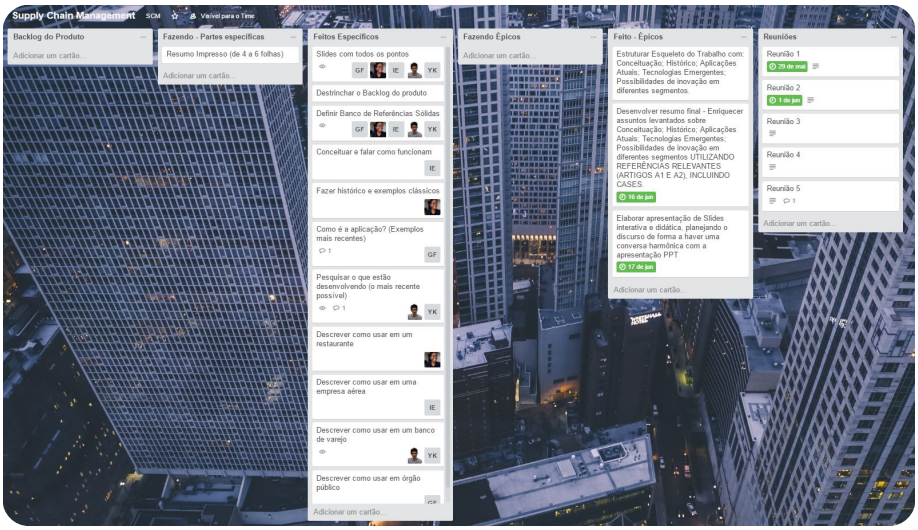
Mandatory Features

You must create a project management app with these features:

- ✓ create/update/delete/display workspaces ;
- ✓ create boards with template choice (ex : kanban), then update/delete/display ;
- ✓ create/update/delete/display lists ;
- ✓ create/update/delete/display cards on a list ;
- ✓ assign persons to a card ;

The order of the features listed above has its importance.





User interface and experience

The UX of your application is just as important as its functionalities.
You should consider some ideas for the experience that you would like to offer to your users, and how your - to be created - app interface will connect with this experience.

Each application is built according to its own set of rules and expectations.
Creation of a visual identity, navigation ergonomics, readability, intuitiveness are all non-exhaustive points that should be taken into account.

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

Consult the guidelines provided by the technology you chose. For instance, if you develop an Android app, we recommend you to look at Material Design/UI

Be attentive to details. For instance, you can easily create Dark Patterns by mistake, which isn't ideal

Your application's visual identity and user interface will be evaluated, just like the quality of the experience it offers to the user. As such, we will check if you respect the chosen platform's guidelines such as:

- ✓ 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.

Application test

As you know, any software must be tested to make sure that it works properly 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 - yet scalable - architecture without code duplication.

You should provide a clear and comprehensive documentation for your project, containing class diagrams, sequence diagrams, components Life Cycle, … depending of the framework and libraries you chose

The goal is to have a document that helps newcomers to easily understand the project architecture and more in order to reduce the project's learning curve and ease communication between team members.

Thus, there is no need to make diagrams/schemas of the entire project, but rather to choose the important parts that needs to be documented.

Delivery part

For the project to be corrected, it must have a README describing how to build the project in development and production.
Without this document, the project will not be evaluated.

v 2.0.1

* apprendre autrement