

{E}

## C2 - End Training Project

C-COD-260

ETP

My own project

1.44



# ETP

repository name: codac-etp



- 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*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

## INTRODUCTION

EPITECH pedagogy practices active teaching, it is project-based and built on experimentation. The project is at the heart of the learning process. It is by seeking solutions to new problems that you have acquired new knowledge and additional experiences. In this context, you have been able to provide important personal work and today you are able to acquire technical skills on your own and create a project from scratch. Agility and openness to the professional world are at the heart of the **Training Project (ETP)**. You will have to be pro-active in the choice of the project to be made according to the technologies proposed by the subject and its imposed constraints.

You will have to work in groups with 2 to 4 of your classmates and your group (say group **owner** or **client**) will be associated with another group (called group **developer** or **provider**).

## RUNDOWN OF THE PROJECT

If you are the group owner, you will have to write the technical specifications of the project you have chosen, and position yourself as a product owner for group 2 who will carry out your project as a service provider. And vice versa.

## PLANNING

- J1 morning : Functional specifications of the group owner and project validation by the pedagogical team
- J1 afternoon : Review of the functional specifications and choice of technologies of the developer group
- J2-J5 : Sprint 1
- J6-10 : Sprint 2



- J11 : Acceptance testing 1
- J11 : Re-evaluation of functional specifications
- J11-J14 : Sprint 3
- J14 morning : Deployment
- J14 afternoon : Acceptance testing 2
- J15 : Presentation of ETP

All these steps should appear on your group organization tool (for example where each step will have to be represented by a map of the comments added for the decisions made during the review).

You must respect the deadlines imposed by your client group and communicate with them according to the schedule provided by the subject. This project has very strong roots in the professional world and will serve you to highlight during your internship research your experience as a developer and project manager; but also to understand the needs and expectations of your future customers. This is why it is very important to fully invest in this professional and operational project.

As a reminder, a sprint is composed of:

- \* A sprint review at the start of the sprint in the presence of the product owner
- \* The election of a SCRUM master at the end of the sprint review for the upcoming sprint
- \* A sprint schedule after the election of the SCRUM master with the devteam
- \* A daily standup, every day starts with the devteam

## PROJECT VALIDATION

At the end of the morning of day 1 you must have your work validated by the pedagogical team and present:

- \* The functional specifications of your project
- \* A presentation of your dev-team with a team name
- \* A presentation of your project in 50 words maximum (French) it will be necessary to describe your project and to make the description of it synthetic. Your text should not be technical, it should be written in layman's terms to a person not familiar with the field of computing.

At the end of the first day, you must have the pedagogical team validate:

- \* The functional specifications of your project after discussion with your product owner
- \* Your justified technological choices



When drafting these documents, always put yourself in the place of those for whom they are intended.



You will pay particular attention to the quality of the documents submitted from both an ergonomic and an orthographic point of view.



All your documents must be submitted in PDF format and be present in your work file during the final presentation.



Only your specifications will be used to grade the finished work of your paired group.



The pedagogical team has the authority to modify your functional specifications until the end of day 1 if it deems necessary. Moreover, the pedagogical team is able to arbitrate your disagreements with your paired group.

## TECHNICAL CONSTRAINTS

- You must only choose technologies previously discussed during the training
- You must create a backend with an API

## FUNCTIONAL SPECIFICATIONS

The functional specification is the description of the functions of a software for its realization. The functional specification describes in detail how the requirements will be taken into account.

In addition to your functional specifications, you must provide:

- \* A product backlog
- \* A design schema (UML type, MySQL Workbench, etc ...)
- \* API documentation



## GROUP ORGANISATION

During this project, you will have to put in place and respect the agility as you approached it during your training.

Then you will need to set up and justify a gitflow that will reflect the agile organization of your group.

We expect at least:

- \* A KANBAN methodology (trello)
- \* A product-backlog with poker planning
- \* 3 sprints of a week

## DOCUMENTATION

During the project, you will have to create documentation for the end user, and technical documentation for future developers who would take over the project, so in this case, your customer group.

### USER DOCUMENTATION

The purpose of user documentation is to have a guide to your product so that users can operate it fully and in full understanding. The form of this document depends on the project and the type of user. Think of making it clear and accessible for the wider audience.



The manual of your microwave oven is a good example of user documentation even if we expect from you a greater reflection on the ergonomics of the interface to produce.

### TECHNICAL DOCUMENTATION

The purpose of the technical documentation is to describe the functional blocks of your code. The goal is to establish a developer's guide that would allow anyone joining the team to contribute to the project by understanding its status, structure, and how to contribute, without having to ask for the necessary information. This technical documentation must be produced as an annotation of your code and a complete README.md for each part of your project.



Have you heard of Doxygen ?



## CONTINUOUS INTEGRATION

During the development of the project you will have to choose and set up a continuous integration stack with at least:

- \* Unit tests
- \* Automated deployment
- \* Acceptance tests



For continuous integration, do you know Travis, Docker, Jenkins, and SonarQube ?



For integration tests, do you know how to use linters (type ESLint or Credo in CLI) and formatters (type PHPSniffer) ?

## DESIGN

The ergonomic of your website is important to think of. How many buttons you would put on your pages, the usage of white space, ...

You have different tools to be sure that the experience will be as great as expected.

For this project you will have to explain your design choices.



You can find some tools and methods to work with : <https://www.designkit.org/methods>.



## BONUS

---

- Application Models (Adobe XD, Photoshop, etc ...)
- Quality tests (Google Lighthouse, Dareboost, etc ...)
- Use an external API (facebook login, google, etc ...)
- Functional tests