

Rapport de Stage

**Concepteur Développeur d’Applications**



**AFPA**

407 Avenue de la Gironde

59640 DUNKERQUE

[**https://www.afpa.fr**](https://www.afpa.fr)

*Période en centre de formation :*

**18/10/2022** au **03/03/2023**

**IEMN**

Avenue Henri Poincaré

59650 VILLENEUVE D’ASCQ

**https://www.iemn.fr**

*Période en stage :*

**06/03/2023** au **02/06/2023**

Table des matières

[Remerciements - 3 -](#_Toc135321966)

[IEMN - 4 -](#_Toc135321967)

[Présentation de l’entreprise - 4 -](#_Toc135321968)

[Présentation de l’équipe - 5 -](#_Toc135321969)

[Environnement de travail - 6 -](#_Toc135321970)

[Outils Utilisés - 6 -](#_Toc135321971)

[Technologies Utilisées - 7 -](#_Toc135321972)

[Présentation du projet - 9 -](#_Toc135321973)

[Présentation du projet en français - 9 -](#_Toc135321974)

[Présentation du projet en anglais - 10 -](#_Toc135321975)

[Tâches effectuées - 11 -](#_Toc135321976)

[Duplication de projets et changement de termes pour certains boutons - 11 -](#_Toc135321977)

[Relance de projet par mail - 12 -](#_Toc135321978)

[Ajout de tests unitaires - 14 -](#_Toc135321979)

[Responsive de certaines parties de l’application - 15 -](#_Toc135321980)

[Système de logs - 16 -](#_Toc135321981)

[Conclusion - 17 -](#_Toc135321982)

# Remerciements

Je tiens tout d’abord à remercier ma formatrice Mme Martine POIX pour sa pédagogie, son soutient et sa disponibilité lors de la formation. Ainsi que mes collègues de formation pour leurs aides et les bons moments de camaraderies apportés lors de la période de formation.

Je remercie également Mr Billel GUERBOUKHA qui fut mon tuteur lors de mon stage ainsi que Mr Benjamin LECHA pour leurs aides et conseils lors de mon apprentissage des technologies utilisées dans l’entreprise ainsi que pour leurs disponibilités tout au long du stage.

Et je remercie pour finir Mr Dino EL HANI, Mr Nicolas GOGUILLON, Mlle Lucie CAUDRON, Mr Antoine LEVA et Mr Julien PRONIER pour leurs aides lors du stage mais aussi pour les bons moments apportés lors de celui-ci.

# IEMN

## Présentation de l’entreprise

L’IEMN (Institut d’Electronique, de microélectronique et de nanotechnologie) est une UMR (Unité Mixte de Recherche) du CNRS (Centre National de la Recherche Scientifique). L’IEMN est un établissement partenaire de plusieurs écoles et universités notamment l’université de Lille et l’université de Valenciennes avec plusieurs programmes de recherches et d’enseignements communs. Il fait également parti du réseau Renatech qui est le réseau académique Français des équipements de pointe dans le domaine de la micro et nanotechnologie piloté par le CNRS.

## Présentation de l’équipe

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquementUne image contenant texte, clipart

Description générée automatiquementUne image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquementUne image contenant texte, clipart

Description générée automatiquementUne image contenant texte, clipart

Description générée automatiquementUne image contenant texte, clipart

Description générée automatiquement

**Julien PRONIER**

**Antoine LEVA**

**Nicolas GOGUILLON**

*Développeur*

**Billel GUERBOUKHA**

*Développeur*

**Lucie CAUDRON**

**Yoan DECONINCK**

**Dino El HANI**

*Développeur*

**Benjamin LECHA**

*Chef de projet*

# Environnement de travail

## Outils Utilisés



Zulip nous a servi pour la communication, c’est un logiciel qui peut être téléchargé sur un poste de travail mais qui est également accessible sur un navigateur internet ce qui fait que l’on peut l’utiliser sur n’importe quel système d’exploitation. Sur Zulip on dispose de plusieurs canaux pour pouvoir communiquer entre nous sur différents sujets.

Gitlab nous sert pour le versioning du projet. Gitlab possède des outils tels qu’une liste et un tableau pour pouvoir créer et assigner des tâches comme des fix, features, etc… aux différents membres du projet.

Pour pouvoir coder nos différentes fonctionnalités, j’avais le choix entre VS Code et PyCharm (un IDE spécialisé dans le langage Python). J’ai eu l’occasion d’utiliser les 2 lors de mon stage.

## Technologies Utilisées



Docker est une plateforme de conteneurisation qui permet de créer des conteneurs pour nos applications. C’est une alternative aux machines virtuelles. Par rapport aux machines virtuelles, un conteneur est plus léger, plus simple et partage son kernel et son système d’exploitation avec d’autres conteneurs. Docker permet grâce aux conteneurs d’avoir le même environnement sur toutes les machines où l’application est déployée. Docker est aussi une très bonne solution pour créer des applications micro service.

Flask est un framework Python qui est classé comme microframework car il est très léger et a pour objectif de garder un noyau simple mais extensible. En effet il ne possède pas de système d’authentification, de validation de formulaire ou autre. Mais de nombreuses extensions nous permettent facilement d’ajouter ces fonctionnalités.

Python est un langage de programmation open-source dit interprété. Il favorise la programmation structurée, fonctionnelle et orienté objet.



AJAX de son nom complet **Asynchronous Javascript And XML** permet d’envoyer des requêtes JavaScript asynchrones ce qui permet une mise à jour de contenu sans rechargement de page.

JavaScript permet d’implémenter des mécaniques complexes à une page web.

Pour faire le responsive de l’application, j’ai utilisé Bootstrap 4.

HTML5 (**HyperText Markup Language**) est utilisé pour créer la structure et le contenu des pages web et CSS3 (**Cascading Style Sheets**) est utilisé pour soigner l’apparence de nos différentes pages.

# Présentation du projet

## Présentation du projet en français

J’ai été positionné sur le projet Repotech, c’est une application de gestion de projet. Cette application sert à déposer des projets de recherche dans différentes centrales du réseau Renatech. L’application est composée de 2 interfaces différentes : l’interface administrateur et l’interface utilisateur. Pour accéder à l’application il faut obligatoirement avoir un compte. Il y a 3 types d’utilisateurs pour permettre de séparer l’accès à ces 2 interfaces : l’administrateur national, les administrateurs locaux et les utilisateurs.

L’administrateur national et les administrateurs locaux ont accès aux 2 interfaces tandis que les utilisateurs ont uniquement accès à l’interface utilisateur.

Dans l’interface utilisateur on peut effectuer une demande de faisabilité de projet si on ne connait pas la centrale qui effectuera le projet. Cette demande pourra être confirmée ou infirmée par un administrateur. On peut également consulter nos demandes de faisabilités avec un historique si elles sont refusées, acceptées ou en attente de confirmation. Si à contrario on connait la centrale qui effectuera le projet, on peut déposer un projet. Dans ce projet on pourra affecter des utilisateurs, partenaires et créer des étapes.

Dans l’interface administrateur on a 2 types d’administrateurs : l’administrateur local qui peut uniquement interagir avec les projets de sa centrale et l’administrateur national qui lui peut interagir avec les projets de toutes les centrales du réseau Renatech. Les administrateurs peuvent valider au non les faisabilités, modifier les projets, les dupliquer, les supprimer et effectuer des relances par mail aux demandeurs et project leaders directement depuis l’application pour les différents projets. Ils peuvent également effectuer des migrations d’établissements et d’utilisateurs en cas de doublon. L’application est développée par l’IEMN un institut de recherche du CNRS pour Renatech et elle est développée en python Flask et Jinja pour faire des templates dans le front. Mon but a été d’ajouter des fonctionnalités dans l’application, d’effectuer des requêtes vers l’API REST de l’application et d’améliorer le design et l’ergonomie de l’application pour les utilisateurs finaux.

## Présentation du projet en anglaise / Project presentation in english

I was positionned on the Repotech Project, this is a project management application. This application is used to deposit research projects in differents Renatech network plants. The application is composed of 2 differents user interfaces: the admin interface and the user interface. To access the application, you need to have an account. There are 3 types of users to separate the access to those two interfaces: the national admin, the local admins and the users.

The national and local admins have access to both interfaces while the users only have access to the user interface.

In the user interface you can ask the network for a feasibility request if you do not know the plant that will take care of the project. This demand can be confirmed or refused by an admin. You can also consult your feasibilities demands with an history to see if they are refused, accepted or pending confirmation. But if you already know with plant will do your project you can directly do a project deposit. In this project you will be able to affect users, partners and create steps.

In the administrator interface we have two types of admins: the local admin who can only interact with projects from his plant and the national admin who can interact with every project existing from every plant of the Renatech network. The administrators can validate or not the feasibilities requests, modify the projects, delete them, send a reminder email about the projects to the applicant and the project leaders directly from the application. They can also migrate establishments and users in case of duplicates. The application is developped by the IEMN a research institute of the CNRS for Renatech and the app is developped in python Flask and jinja to make templates in the front-end. My role was to add multiple functionalities in the application, to make requests to the API REST of the app and to upgrade the design and ergonomics of the application for the end user.

# Tâches effectuées

## Duplication de projets et changement de termes pour certains boutons

Ma première tâche fut de changer les termes pour deux boutons dans la page d’accueil de la partie utilisateur de l’interface dans la page d’accueil. En effet 2 boutons avaient un terme très similaire alors qu’ils faisaient des actions très différentes, ce qui rendait les utilisateurs confus. J’ai donc modifié ce terme pour 2 autres termes bien explicite de ce que ces 2 boutons faisaient ces 2 termes sont : « Demander au réseau une étude de faisabilité » et « Déposer un projet ». J’ai également changé le terme d’autres boutons pour correspondre avec ceux changés précédemment.

J’ai ensuite dû rajouter une nouvelle feature qui est la possibilité de dupliquer un projet existant. En effet l’application étant une application de gestion de projet de nombreux projets sont créés sur cette application. Ce qui implique que parfois des projets se ressemblent en tous points (mis à part les personnes et partenaires affectés au projet). Donc pour simplifier le travail des utilisateurs de l’application pour qu’ils n’aient pas à re-rentrer toutes les informations. J’ai dû créer une fonction qui duplique toutes les informations d’un projet mis à part son numéro de projet (qui est un numéro unique), les dates de début et de fin, le statut du projet. La duplication de projet peut être effectuer à la fois dans l’interface utilisateur mais aussi dans l’interface administrateur.

L’utilisateur a le choix de repositionner les mêmes membre et partenaires du projet d’origine sur le projet dupliqué et il a également le choix de saisir une date de fin pour le projet. Cependant s’il ne saisit pas de date de fin pour le projet lors de la duplication, le projet se termine automatiquement au bout d’un an car un projet doit obligatoirement avoir une date de fin.

## Relance de projet par mail

Après cela j’ai dû m’occuper de la relance de projet par mail. C’est une fonctionnalité qui permet aux administrateurs d’envoyer des mails à propos d’un projet bien particulier. Comme c’est une fonctionnalité qui ne sert qu’aux administrateurs, le bouton de « relance de projet » n’apparaît que dans l’interface administrateur. Pour l’envoi de mail j’ai utilisé Flask-Mail qui est un package de Flask installable sur le projet qui été déjà présent avant mon arrivé qui nous permet de facilement envoyer des mails.



Quand on clique sur le bouton de relance une modal apparaît et permet à l’administrateur de saisir un message qui sera dans le corps du mail ainsi que le sujet du message qui sera envoyé aux demandeurs et aux project leaders du projet.

**Modal pour la saisie du sujet et du message du mail :**

**Modal pour la saisie du sujet et du message du mail (remplie) :**

Après avoir validé l’envoi du mail, la fonction recherche les e-mails des project leaders et du demandeur pour les mettre en destinataires et pour faciliter la discussion par mail on récupère l’email de l’expéditeur et on l’ajoute dans le champ « reply-to » de la fonction. De cette manière quand un mail de relance aura été envoyé, les destinataires pourront répondre à l’expéditeur même s’ils ne connaissaient pas l’e-mail de celui-ci.

Quand le mail a bien été envoyé on a un message qui apparaît dans la page du projet côté administrateur qui nous confirme le bon envoi de celui-ci.



Et voici le mail que les project leaders et le demandeur recevront :



On voit bien ici le nom et prénom de la personne qui reçoit le mail mais aussi le numéro du projet concerné comme un utilisateur peut être positionné sur plusieurs projets, il faut pouvoir savoir de quel projet on parle. On a également le message précédemment saisi dans la modal et un bouton qui permet de directement aller sur la page côté utilisateur du projet concerné.

## Ajout de tests unitaires

Ensuite je me suis occupé de faire des tests unitaires afin de valider l’intégrité des fonctionnalités du projet. Pour simplifier le travail des développeurs. J’ai donc dû faire des fonctions qui appellent les différentes fonctions du projet à tester et j’ai dû saisir le résultat attendu pour qu’à la fin de la batterie de tests on compare le résultat attendu avec celui obtenu. Pour les tests unitaires on peut soit tester les fonctions d’un seul fichier soit toutes les fonctions qui sont située dans un dossier. S’il y a une erreur lors des tests on a un message qui nous indique le nombre de tests qui ont fonctionnés et le nombre de tests échoués avec une erreur.



J’ai écrit les tests unitaires pour qu’ils couvrent la majeure partie des fonctions de l’application.

## Responsive de certaines parties de l’application

J’ai également dû m’occuper du responsive de l’application dans l’interface utilisateur car un retour des différents utilisateurs était que l’application était difficilement utilisable sur mobile, tablette mais également certaines résolutions d’écrans d’ordinateurs. J’ai donc utilisé Bootstrap 4 et retravaillé le CSS afin de faire en sorte que l’application supporte une plus grande variété de résolutions d’écrans allant du mobile au desktop en passant par les tablettes et les laptops. Mon travail s’est porté sur les 4 pages les moins responsives et donc les plus problématiques :

* La page de Connexion
* La page d’Accueil (index)
* La page de Dépôt de projet
* La page de Détails d’un projet

## Système de logs

Pour finir ma dernière tâche fût de créer un système de logs pour les actions effectuées par un utilisateur sur les projets. Après avoir consulté mon tuteur de stage nous avons convenus que la meilleure solution était de faire un fichier de logs pour avoir un historique local mais aussi d’avoir une table dans la base de données pour un historique « long ». Pour faire les logs de l´application j´ai utilisé le module Flask-Logging qui nous permet de faire des logs personnalisés pour l´application. J´ai également choisi Flask-logging car il nous permet aussi de pouvoir automatiquement envoyer un mail à un adresse mail de notre choix quand un log d´erreur est renvoyé par l´application, ce qui permet une plus grande réactivité en cas d’erreur dans celle-ci car l´utilisateur n´a pas besoin de contacter l´équipe de développement, l´application le fait elle-même.

# Conclusion

Ce stage m’a beaucoup apporté j’ai pu apprendre une nouvelle technologie que j’apprécie beaucoup et je suis également très heureux d’avoir pu effectuer mon stage dans un lieu comme l’IEMN où j’ai pu rencontrer des personnes très talentueuses et très attentionnées.

Ils m’ont chacun apportés quelque chose et j’espère avoir pu faire de même. J’ai beaucoup appris à leurs côtés.