

Polystage

Rapport Final du projet

|  |  |  |
| --- | --- | --- |
| Référence Fournisseur | : | PRJ/QA/TST-04120102-0A |
| Date | : | 09 Janvier 2021 |
| Version/Édition | : | 0A |
| État | : | Préliminaire |
|  |  |  |
| Type de diffusion | : | Diffusion restreinte |
| Autre référence | : |  |

Table des matières

1 Introduction...........................................................................................................................5

1.1 Description du projet........................................................................................................5

1.2 Les exigences fonctionnelles...........................................................................................5

1.3 La solution retenue..........................................................................................................5

1.4 Décomposition du dossier................................................................................................5

1.5 Outils utilisés....................................................................................................................6

2 Terminologie..........................................................................................................................7

2.1 Abréviations.....................................................................................................................7

2.2 Définitions des termes employés.....................................................................................7

3 Description Générale............................................................................................................8

3.1 Architecture générale.......................................................................................................8

3.2 Structure des données échangées..................................................................................8

3.3 Structure des données globales......................................................................................8

4 Description détaillée.............................................................................................................9

4.1 Décomposition générale..................................................................................................9

4.2 Couverture fonctionnelle..................................................................................................9

5 Tests unitaires.....................................................................................................................14

5.1 Description des tests unitaires.......................................................................................14

5.2 Mise en oeuvre des tests unitaires.................................................................................14

5.3 Couverture des tests unitaires.......................................................................................14

# Introduction

Dans le cadre d’un projet de fin de semestre, l’école Polytech confie à ses étudiants de 5A la réalisation d’une application de gestion de stages.

L’objectif est de créer un dossier AMUBOX partagé entre les enseignants et rassemblant toutes les données concernant les stages effectués sous tutorat par les étudiants de Polytech.

Nous verrons dans ce document la conception ainsi que les points importants que nous pouvons retrouver dans notre projet.

## Livrables

Différents livrables nous ont été demandés pour ce projet :

* Une présentation.
* Une documentation.
* Une vidéo de démonstration.
* Le code de notre projet.
* Un rapport final.

Ces derniers pourront être trouvés sur le GitHub dédié à notre projet.

**1.2 Présentation du projet PolyStages**

**1.2.1 Contexte et enjeu du projet**

Le projet consiste à concevoir et développer une application de gestion de stages d’élèves ingénieurs Polytech dans le but de constituer un dossier AMUBOX partagé entre les enseignants.

**1.2.2 Objectifs du projet**

Les élèves ingénieurs pourront uploader leurs rapports de stage et leurs présentations pour les soutenances, et remplir leurs grilles de compétences.

Les tuteurs en entreprise pourront remplir en ligne les fiches d’évaluation des stagiaires dont ils ont la charge ainsi que les grilles de compétences.

Les enseignants auront la possibilité de consulter aisément les différents documents et ils peuvent aussi enregistrer les stages effectués par les étudiants à partir d’un fichier CSV.

## Les services assurés

Pour les étudiants :

* L’upload des rapports de stage et des présentations pour les soutenances

Pour les enseignants :

* La consultation des documents, rapports de stage, présentations
* La possibilité de parcourir tous les stages effectués par les étudiants et enregistrés dans la base de données
* Ils peuvent effectués une recherche pour retrouver un ancien stage effectué par un étudiant et télécharger soit l’ensemble de stages soit quelques stages trouvés en effectuant une recherche par (Entreprise, Année, Promo, Titre du stages)
* La visualisation des statistiques des stages effectués par rapport à l’année, l’emplacement du stage, les entreprises.

Pour les tuteurs en entreprise :

* L’accès en ligne pour remplir les fiches d’évaluation ainsi que les grilles de compétences de leurs stagiaires

Envoi de mails à chaque tuteur

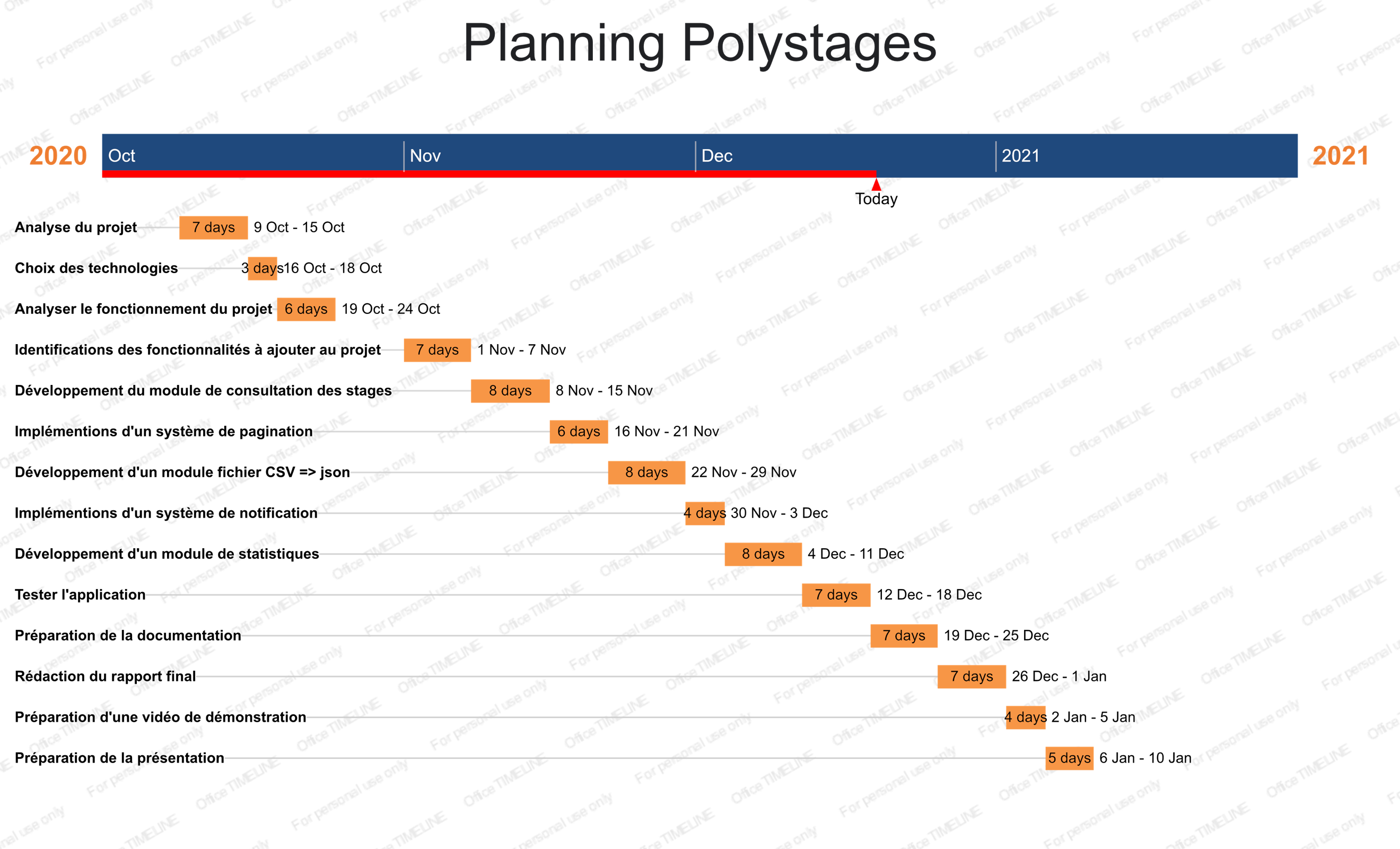
* Contenant un lien permettant de compléter la fiche d’évaluation
* Contenant un lien permettant de compléter les grilles d’évaluation

## Ordonnancement du projet

### **Planning du projet**

Le diagramme de Gantt ci-dessous représente les différentes tâches et leurs durées.

La durée totale du projet est de 87 jours.



### **Méthode de gestion du projet**

La méthode Agile est un ensemble de pratiques de gestion de projet. Cette méthode implique au maximum le client dans tout le processus de réalisation du produit afin de mieux comprendre les besoins et les fonctionnalités demandées.

Plus précisément on a choisi de travailler dans le cadre « Scrum » qui est basé sur un ensemble de réunions clairement définies et strictement limitées dans le temps (timeboxing) où on exécute les tâches suivantes :

* **Planification du Sprint** (Sprint = itération) : au cours de cette réunion, l'équipe sélectionne les éléments prioritaires qu'elle pense pouvoir réaliser au cours du sprint.
* **Revue de Sprint** : au cours de cette réunion qui a lieu à la fin du sprint, l'équipe présente les fonctionnalités terminées au cours du sprint et recueille les ‘feedbacks’ du client.
* **Mêlée quotidienne** : il s'agit d'une réunion de synchronisation de l'équipe de développement.

Nous avons décidé de diviser le projet en trois sprints.

A la fin de chaque sprint une réunion sera planifiée avec notre professeur encadrant qui est dans notre cas le client.

**Sprint 1 :**

Dans ce sprint, nous avons analysé l’intégralité du projet ça veut dire Comprendre le fonctionnement du projet.

Nous avons choisi les technologies pour le développement de notre application.

Après nous nous sommes concentrés sur les défauts du projet.

**Sprint 2 :**

Dans le deuxième sprint nous avons identifié les fonctionnalités à ajouter au projet et nous avons commencé le développement.

**Sprint 3 :**

Nous avons effectué des tests sur les fonctionnalités ajoutées et nous avons discuté leurs efficacités avec notre professeur.

## Les exigences fonctionnelles

L’application doit permettre de constituer une application permettant la gestion des stages concernant les étudiants de l’école Polytech.

**Partie utilisateur**

Upload / Download

Les fichiers seront renommés et classés automatiquement dans des dossiers

Utilisation de formulaires

* Pour les rapports de stage : contrainte date limite pour l’upload
* Pour les présentations
* Pour les fiches d’évaluation
* Pour les grilles de compétence

**Partie administrateur**

Alerte

Envoi automatique d’une alerte par mail à l’administrateur si le rapport de stage n’est pas rendu dans les temps par un élève.

Idem si un étudiant n’a pas eu d’évaluation.

Module de recherche

La possibilité d’afficher tous les stages sauvegardés dans la base de données et y effectués des recherches suivant l’année du stage, la promo de l’étudiant, le nom de l’entreprise, le titre du stage ou même selon la description de la mission du stage.

On peut avoir plus d’information sur un stage en affichant les données correspondantes à ce stage et on peut aussi mettre à jour ses données et les sauvegarder.

On peut aussi supprimer un stage de notre base de données.

Les administrateurs ont la possibilité de télécharger soit l’ensemble des stages soit les stages résultant à une recherche sous forme d’un fichier csv

Module ajout des stages à partir d’un fichier CSV

Cette fonctionnalité permet un ajout rapide d’un ensemble de stages contenus dans un fichier CSV à notre application.

Avant de sauvegarder les stages extraits du fichier csv on peut les consulter et mettre à jours leurs informations à travers l’interface de l’application dans le cas d’une erreur.

Module statistique

Nous pouvons consulter ce module pour avoir des statistiques sur les stages effectués par les étudiants.

Par exemple nous pouvons avoir le pourcentage des stages effectués à l’étranger par nos étudiants.

Sécurité

Utilisation de SSL/TLS (HTTPS) pour la confidentialité des communications.

L’application devra se prémunir d’injection de commande et d’injection SQL.

Protection des données sensibles de la base de données.

Les mots de passe doivent être stockés et hachés.

## La solution retenue

La problématique consiste à créer une application Web écrite en JavaScript s’exécutant naturellement dans l’écosystème du navigateur, construite de façon modulaire.

La solution conceptuelle retenue pour réaliser ce projet est d’utiliser la plateforme Node.js associée au framework Angular de Google.

Node.js est un environnement de très bas niveau, il est différent parce qu’il utilise le langage JavaScript basé sur les évènements.

Cette architecture est particulièrement adaptée aux applications induisant de fréquents transferts de données entre le client et le serveur. Node.js est très rapide, cela tient à deux choses :

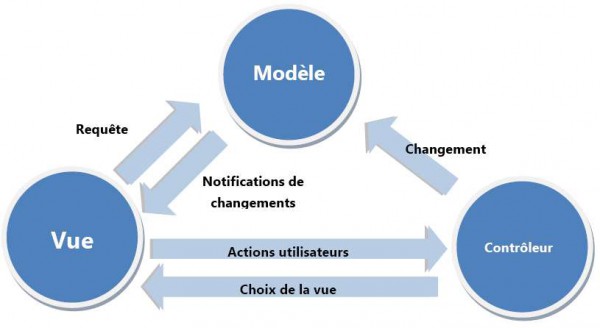
* + L’utilisation du moteur d’exécution rapide V8 de Google
  + Son fonctionnement non bloquant

Le framework Angular est idéalement conçu pour créer des applications intégrées dans une architecture client-serveur mettant en œuvre des services web avec la plateforme JavaScript Node.js.

*Le paradigme de conception modèle-vue-contrôleur*

L’application Angular est constituée de différents modules implémentant les grandes fonctionnalités de l’application, les modules regroupant eux-mêmes des composants implémentés par des classes.

Angular se projette dans un modèle de conception **MVC**.



* Le modèle regroupe les données (qu’elles soient générées par le serveur ou le client) et les traitements métier effectués sur celles-ci.
* La vue correspond à l’interfaçage de l’application (la vue est composite et construite sur une agrégation de templates).
* Le contrôleur désigne un mécanisme permettant, à partir d’une action sur la vue, d’initier un traitement : ce traitement met à jour le modèle ; le contrôleur peut en retour modifier la vue.

Le choix de JavaScript

Le code JavaScript mis en œuvre du côté serveur est non seulement utilisés pour gérer des sockets mais surtout pour créer des serveurs http très réactifs. Le principal environnement JavaScript permettant de créer de tels serveurs est Node.js.

La syntaxe qui permet de formater ces objets est appelée JSON (*JavaScript Object Notation*).

La plateforme Node.js

Node.js est un environnement permettant d’exécuter du code JavaScript hors d’un navigateur.

Son architecture est modulaire et événementielle.

Ses deux qualités principales :

* Sa légèreté (en corollaire de sa modularité)
* Son efficacité induite par son architecture monothread

Node.js permet donc, pour les applications Web, de créer des serveurs extrêmement réactifs

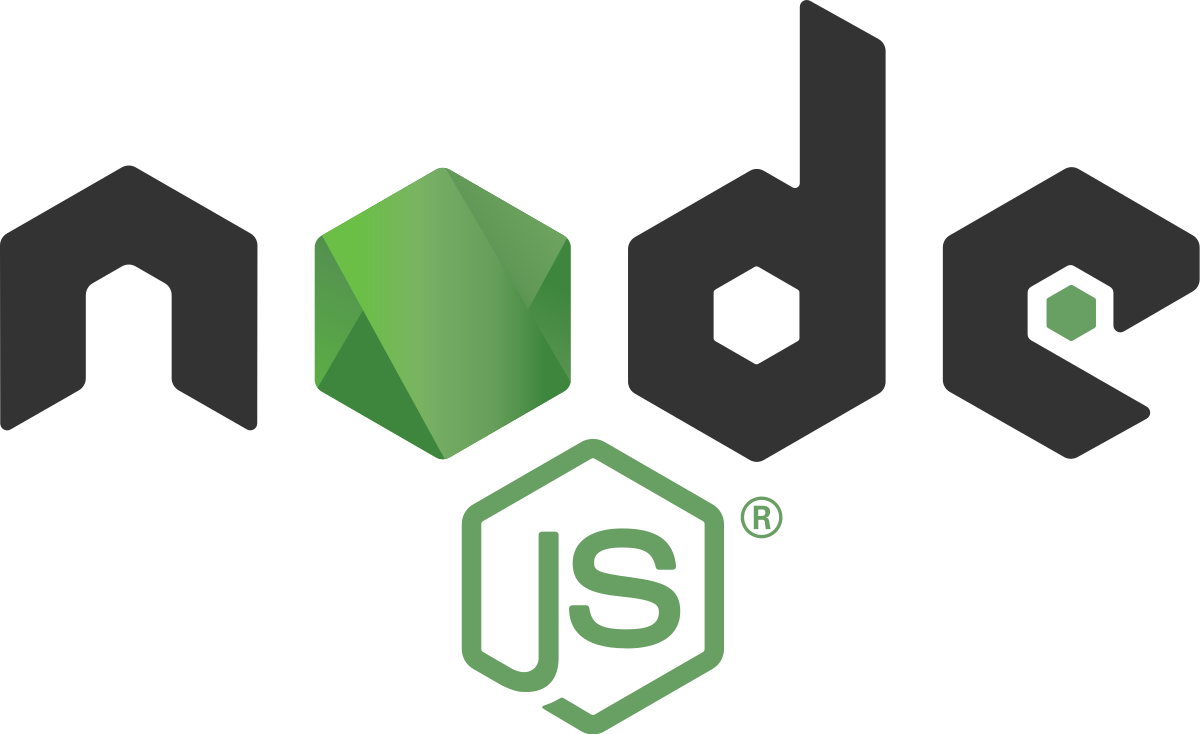
NPM, le gestionnaire de modules de Node.js

Les modules sont les briques conceptuelles d’une application Node.js.

Le module express

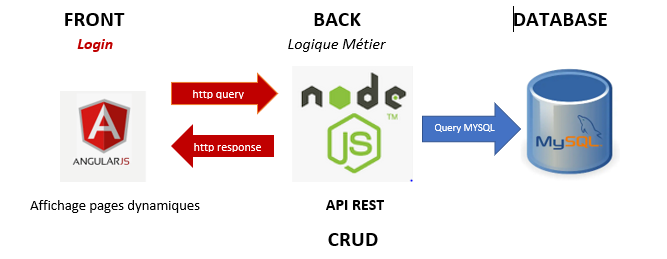
Le module express permet d’ajouter de puissantes fonctionnalités à un serveur Node.js, notamment la gestion de routes REST *(Representational State Transfer)* et la gestion des templates.

La gestion de routes **REST** permet d’associer des requêtes http à une action déterminée par un contrôleur



## Description Générale

**2.1 Architecture générale**



Création d’un serveur renvoyant les données

Deux modules de Node.js seront nécessaires :

* Le module express pour accéder à Node.js via des URL au format REST (il permet également de gérer des templates).
* Le module fs pour accéder à des fichiers stockés sur le serveur.

Le gestionnaire de modules NPM Node Package Manager permet d’installer des modules qui ne sont pas directement présents sur Node.js.

S’agissant d’un environnement JavaScript, ces données seront naturellement formatées en JSON.

Le module express permet très simplement d’ajouter de puissantes fonctionnalités à un serveur Node.js, notamment la gestion de routes REST ( Representational State Transfert)

# Les Interfaces De l’application

L’interface utilisateur, quant à elle, consiste à améliorer la présentation et l’interactivité du Web ou

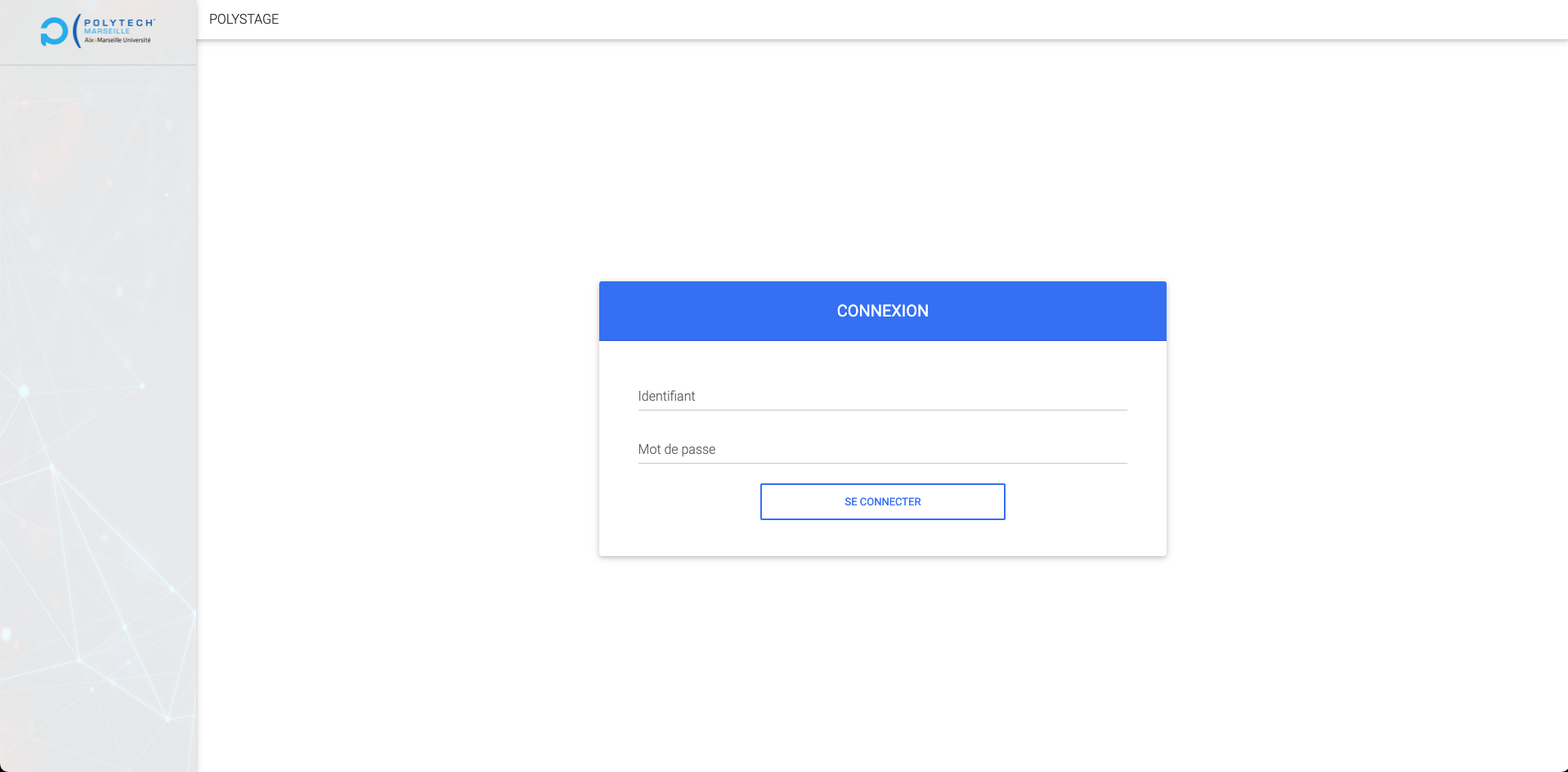
d’application. Il se concentre sur l’aspect de l’application et interagit avec les utilisateurs.

Nous avons alors opté pour une démarche qui permet de réduire le nombre de cliques demandé par l’utilisateur afin de réaliser une tâche dans la conception des interfaces utilisateurs.

## Page Login

C’est la première interface rencontrée par l’utilisateur lors du lancement de notre application

Elle contient deux champs ‘ Identifiant ’ et ‘ Mot de passe ’ afin d’accéder au compte utilisateur.



## Page d’accueil

Une fois nous sommes connectés à l’application, nous nous retrouvons sur la page d’accueil.

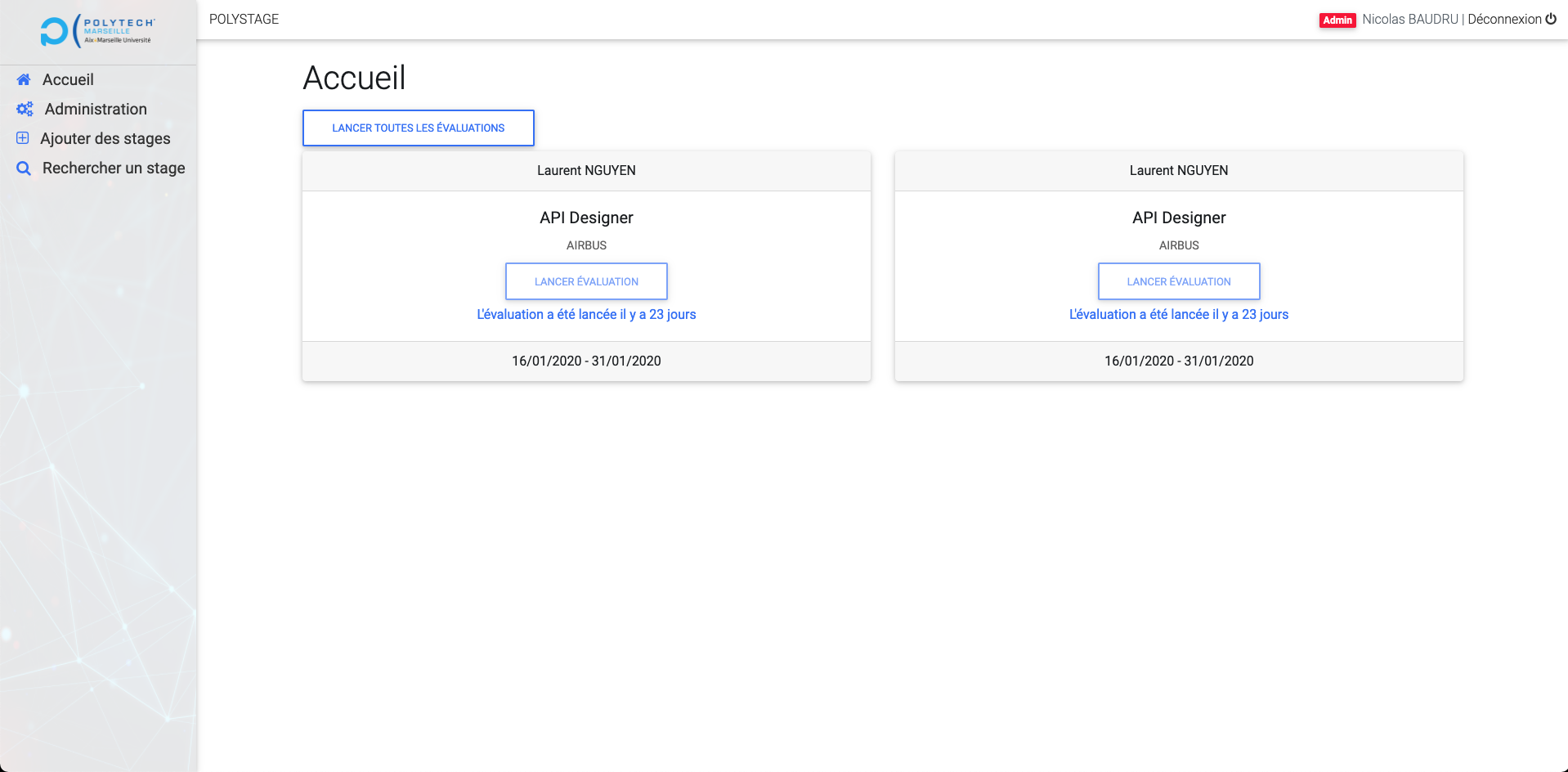
En haut à gauche nous avons un menu qui permet de se déplacer entre les différentes pages de

l’application.

En haut à droite nous avons le nom de l’utilisateur connecté afficher, son rôle et un bouton pour se déconnecter.

Dans la partie centrale de l’application nous avons une liste des étudiants qui que ce prof encadre.

Il a la possibilité de lancer l’évaluation individuel de chaque étudiant soit lancer l’évaluation pour tous les étudiants.



## Page Administration

Cette interface est déclenchée lorsque l’utilisateur clique sur le bouton « Jouer » de l’interface page d’accueil. Elle permet au joueur de choisir l’espèce qui va supporter dans le jeu en cliquant sur le bouton qui correspond.



## Interface 3 : Fenêtre Principal

Cette interface est la fenêtre principale où le jeu va se dérouler. Elle contient un menu (en haut-gauche) où l’utilisateur peut quitter le jeu, afficher la page d’aide ou même interagir avec le jeu.

Les interactions possibles sont :

* **Food** : Rajouter la quantité de la nourriture pour chaque espèce.
* **Battle** : Lancer le combat entre les deux espèces.
* **Reproduction** : Lancer la reproduction pour les deux espèces. Un chien va renter dans sa niche et pareil pour le chat.

Elle contient aussi des éléments (les personnages du jeu) : des chiens et des chats.

Nous pouvons visualiser sur cette fenêtre les différentes interactions entre les personnages pendant le déroulement du jeu.



## Interface 4 : Fenêtre d’aide

C’est la fenêtre d’aide où nous expliquons les règles de jeu. Elle est déclenchée lorsque l’utilisateur clique sur le bouton ‘Aide’ de l’interface 1 ou 3. Elle contient aussi un bouton ‘Retour’ pour revenir à la fenêtre précédente.



## Interface 5 : Fenêtre des résultats du jeu

Cette interface contient les résultats du jeu et s’affichera à la fin du combat.

Elle affiche les résultats du combat entre les deux espèces. À la fin on récupère le gagnant, soit les chiens soit les chats et on décide si l’utilisateur a gagné ou a perdu le combat suivant l'espèce qu’il a choisi de supporter.



# Conclusion

Nous avons eu le plaisir de travailler sur ce projet. À travers ce dernier nous avons découvert un aspect parmi plusieurs de travail d’ingénieur. En effet, nous avons appris à gérer les différentes contraintes de temps et d'organisation qu’il peut y avoir dans un projet d’ingénieur.

Ce projet, nous a aussi beaucoup apporté sur la maîtrise et la connaissance de la programmation orienté objet et surtout le langage Java. De plus, en travaillant en équipe, nous avons amélioré notre organisation ainsi que notre communication, en permettant d’effectuer un meilleur travail. Nous retenons aussi, qu’il ne faut pas hésiter à demander de l’aide lorsque nous sommes bloqués et qu'il faut écouter toutes les idées pour construire une meilleure. Cela, nous permet d’avancer plus vite et donc d’obtenir de meilleurs résultats.

Si nous avions plus de temps, nous aurions pu faire plus d'améliorations dans le jeu, plus d'interactions et d’animations.