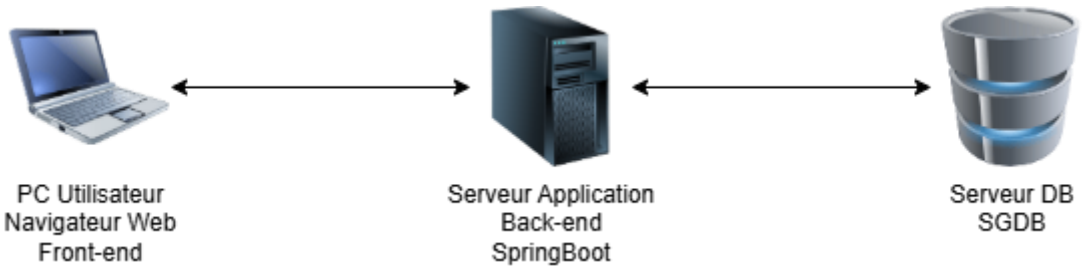
NFP121 : Projet FullStack

Cahier des charges - CDC

Objectif

Implémenter une application Web Full Stack (architecture 3 tiers) afin de gérer la notation des devoirs pour des classes d'étudiants.



Description

Une **classe**, caractérisée par une **dénomination**, est composée de **N étudiant.e.s**.

Un.e **étudiant.e** est caractérisé.e par un **nom**, un **prénom**, une **photo** (facultative). Un.e **étudiant.e** fait partie d'**une et une seule classe**.

Un **devoir**, caractérisé par une **description**, une **catégorie** (CC ou Examen), une **date de création** et un **coefficient**, concerne une **classe** et une **matière**.

Une **matière** est caractérisée par une **dénomination**.

L'application devra permettre de :

- Pour les **classes** :
  - **Créer** une nouvelle classe : **saisir** sa **dénomination** et éventuellement y **ajouter** des **étudiant.e.s disponibles**
  - **Modifier** une classe : **modifier** sa **dénomination**, y **ajouter** des **étudiant.e.s disponibles** (ne faisant partie d'aucune classe) et/ou y **enlever** des **étudiant.e.s** (qui deviennent disponibles)
  - **Supprimer** une classe : les **étudiant.e.s** de celle-ci deviennent disponibles et tous ses **devoirs** sont **supprimés**.
  - **Affiche** le **bulletin de notes** : **afficher** pour chaque **étudiant.e**, sa **moyenne par matière** et sa **moyenne générale**

- Pour les **étudiant.e.s** :
  - **Créer** un.e nouveau/nouvelle étudiante.e : **saisir** son **nom** et son **prénom** et sa **photo** et/ou sa **classe**
  - **Modifier** un.e étudiant.e :
    - si **aucune note saisie** : **modifier** son **nom**, son **prénom**, sa photo et/ou sa **classe**
    - si **au moins une note saisie** : **modifier** son **nom**, son **prénom** et/ou sa **photo**
  - **Supprimer** un.e étudiant.e : **supprimer** aussi toutes ses **notes**
  - **Afficher** son **relevé de notes** :

        - Pour chaque **matière**, la **note** de chaque **devoir** et la **moyenne** de la **matière**
        - La moyenne générale

- Pour les **matières** :
    - **Créer** une nouvelle matière : **saisir** sa **dénomination**
    - **Modifier** une matière : **modifier** sa **dénomination**
    - **Supprimer** une **matière** : si elle **n'est pas utilisée** dans un **devoir**

- Pour les **devoirs** :
    - **Créer** un nouveau devoir : **choisir** la **classe** et la **matière** concernées, et **saisir** sa **description**, sa **catégorie** et son **coefficient** (1.0 par défaut). La **date de création** est la date du jour
    - **Modifier** un devoir :
        - si **aucune notation** n'est saisie : **modifier** la **classe**, la **matière**, sa **description**, sa **catégorie** et/ou son **coefficient**.
        - si **au moins une notation** est saisie : **modifier** sa **description**, sa **catégorie** et/ou son **coefficient**.
    - **Supprimer** un devoir : **supprimer** aussi toutes ses **notations**.
    - **Saisir** / **modifier** la **notation** d'un devoir : **saisir** / **modifier** la **note** sur 20 des **étudiant.e.s** de la **classe** associée

Livrables attendus

Vous devrez fournir un **fichier zip** portant le NOM des 2 membres du binôme contenant :

1. le DDL de votre DB au format texte
2. le code source du back-end en Java / SpringBoot
3. le Powerpoint de présentation de votre projet avec des copies d'écrans de votre application

Recommandations

Généralités

- Bien lire le sujet pour concevoir votre projet et :
    - Identifier les objets et leur interaction / dépendance
    - Identifier les règles fonctionnelles
- Documenter le code du back-end
- Respecter les bonnes pratiques et optimiser votre code
- Tester unitairement chaque API REST du back-end pour être sur qu'elle fonctionne parfaitement dans le respect des règles fonctionnelles (voir CDC)
- Répartir les tâches entre les 2 membres du binôme afin de travailler un parallèle

Respecter l'ordre de réalisation

1) **DB :**
    a) Modéliser la DB correspondant aux objets à stocker et leurs relations (1:1, 1:n et n:m)
    b) Créer la DB et l'alimenter avec un jeu de données (une classe, 3 élèves, 1 matière)
2) **Back-end :**

a) Implémenter, en SpringBoot, chaque API REST des CRUD (C=Create/Créer, R=Read/Lire, U=Update/Modifier, D=Delete/Supprimer) de chaque objet en respectant les règles fonctionnelles
b) Tester chaque API REST avec des données de test, via swagger ou un outils tel que Postman, pour vérifier qu'elle fonctionne parfaitement et dans le respect du CDC

3) **Front-end :**
a) Implémenter chaque vue des CRUD de chaque objet en utilisant les API REST correspondantes
b) Tester chaque vue et vérifier que les données affichées et/ou stockées en DB sont correctes

Notation

Le barème de notation sur 40 points est :

- DB : DLL fournie pour créer la DB .....5 points
- Back-end :
  - API CRUD Classe .....8 points
  - API CRUD Etudiant .....8 points
  - API CRUD Matière.....4 points
  - API CRUD Devoir + notation .....10 points
- Powerpoint de présentation .....5 points