Cahier des charges

À fin de faire les appels api nécessaires depuis le dossier admin (Currency_Front), il convient de modifier les valeurs de ces trois variables : **urlCurrency** , **urlPairs** , **urlConverts** et de les remplacer par les bonnes url.
Pour cela, il faut se rendre dans **Currency_Front/src/App.vue** ligne 9, 10 et 11

Avant de commancer, il faut lancer 2 commandes sur terminal :
npm i et **npm i axios** dans le dossier **Currency_Front**

Analyse Client :

De ce que j'ai compris, MoneyValue est une startup qui veut développer une plateforme de conversion monétaire (convertir des devises en d'autres devises).
Mon client veut une API réutilisable et un front pour la gestion de l'administration de l'API.

Choix technologiques

*Utilisation de **SANCTUM** pour l'authentification entre le back (laravel/MySQL) et le front (vueJS).
SANCTUM va créer un tokens chaque connexion et l'enregistrer dans la table personnal_access_token et renvoyer un token avec plainTextToken. Et on va utiliser cette token plainTextToken pour déconnexion.
C'est le plus adapaté pour la récupération du **token** d'authentification côté front.
*Utilisation de **AXIOS** pour les appels api depuis le front (vueJS).

Evaluation du temps de travail

Poste de développement	Nombre de jours de travail
Initialisation du projet laravel(api), création de la bdd, reflexion autour du nombre de tables et des relations entre elles, migration, seeders, factories, models	1 jour
Création des routes(api.php) et des controllers(currencies, pairs et converts) et développement des fonctions de chaque controller	1,5 jours
Authentification avec sanctum	1,5 jours

Lier vueJS avec Laravel, faire les appels api pour chaque bloc Front Admin (CRUD, style, routes, authentification)	2 jours

Liste fonctionnelle

API :

- **Affichage de la liste des devise** en accédant à la route **api/currencies** sous forme de JSON
- **Affichage de la lise des paires** en accédant à la route **api/pairs** sous forme de tableau :

["id de la paire, nom devise initiale => nom devise destination => taux de change"]
À la base je renvoie un json sous la forme suivante : [{"id de la paire, id devise initiale => id devise destination => taux de change"}]
j'ai donc cherché une solution pour renvoyer les noms au lieu des id et la solution était de renvoyer un tableau !

- **Calcul de la conversion et incrémentation du nombre de requêtes effectué sur cette conversion** avec la fonction convert(somme à changer, devise initiale, devise de destination) dans AdminConvertController.php.
  J'y accède grâce à l'url :
  api/converts/somme à changer/devise initiale/devise de destination
  Si la paire n'existe pas, un message d'erreur est affiché .

Administration :

- **Page d'authentifiaction :** /login
  Adresse de connexion : admin@admin.fr
  Mdp : admin
- **Liste des devises avec le CRUD :** /admin/currencies
  Consultation de la liste des devises , création, modification et suppression d'une devise
  On ne peut pas ajouter une devise qui existe déjà
- **Liste des paires avec le CRUD :** /admin/pairs
  Consultation de la liste des paires.
  La création d'une paire engendre la création de son reverse
  Modification et suppression d'une paire
  La suppression d'une paire engendre la suppression de son reverse
  On ne peut pas ajouter une paire qui existe déjà
  Si on veut créer une paire avec une devise qui n'existe pas encore, j'ai mis à disposition un lien qui renvoie vers la création d'une nouvelle devise.

Recettage

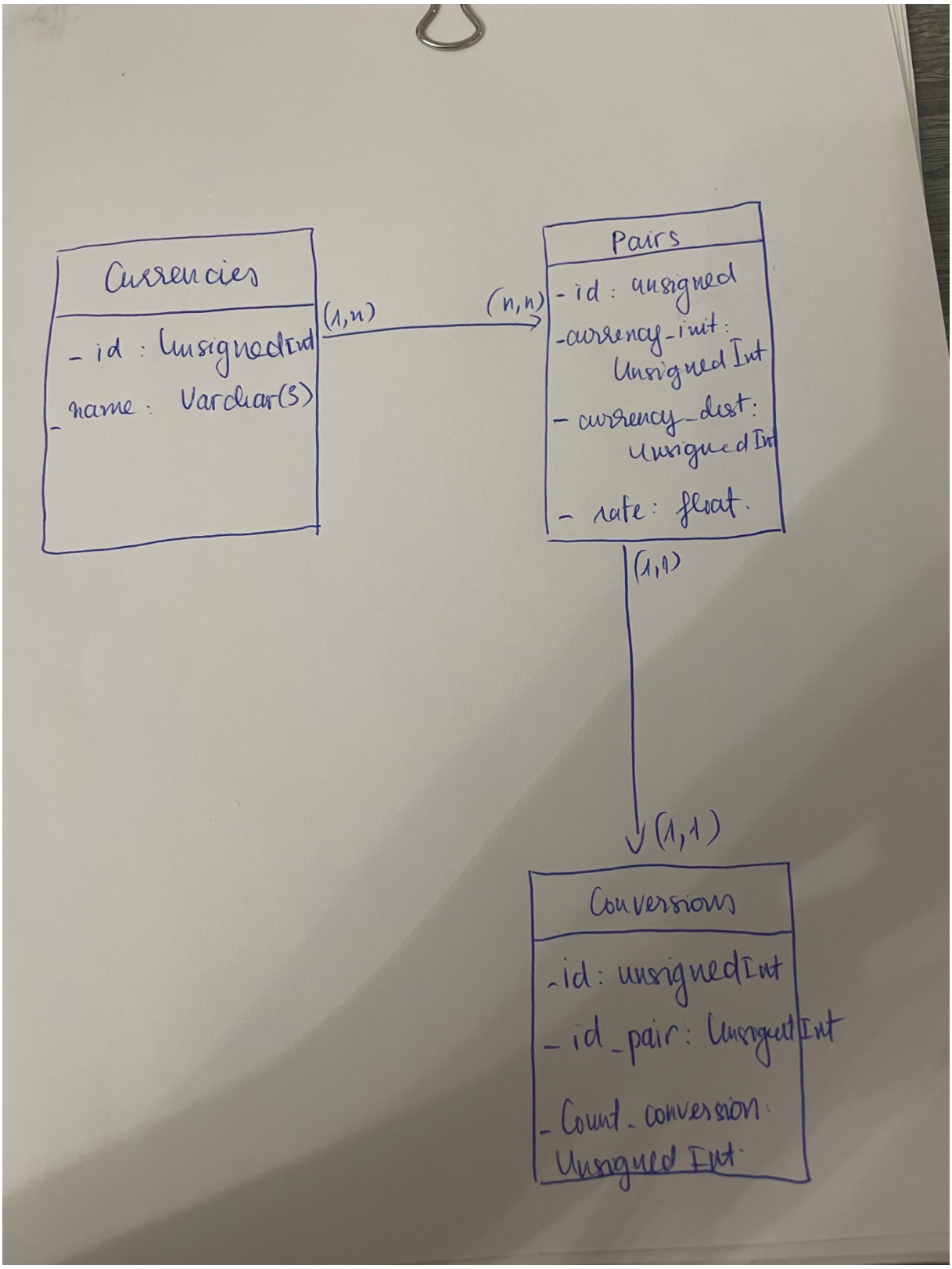
Liste fonctionnelle	Etat de la fonctionnalité
Affichage de la liste des devises côté api	Opérationnelle

Affichage de la liste des paires côté api	Opérationnelle
Calcul de la conversion et incrémentation du nombre de requêtes effectué sur cette conversion	Opérationnelle
Page d'authentification (administration)	Opérationnelle
Liste des devises avec le CRUD (administration)	Opérationnelle
Liste des paires avec le CRUD (administration)	Opérationnelle

Documentation de l'API

- **Affichage de la liste des devise** en accédant à la route **api/currencies** sous forme de JSON
- **Affichage de la lise des paires** en accédant à la route **api/pairs** sous forme de tableau :
  ["id de la paire, nom devise initiale => nom devise destination => taux de change"]
  À la base je renvoie un json sous la forme suivante : [{"id de la paire, id devise initiale => id devise destination => taux de change"}]
  j'ai donc cherché une solution pour renvoyer les noms au lieu des id et la solution était de renvoyer un tableau !
- **Calcul de la conversion et incrémentation du nombre de requêtes effectué sur cette conversion** avec la fonction convert(somme à changer, devise initiale, devise de destination) dans AdminConvertController.php.
  J'y accède grâce à l'url :
  api/converts/somme à changer/devise initiale/devise de destination
  Si la paire n'existe pas, un message d'erreur est affiché .

Diagramme de la base de données



Adresse Github:

https://github.com/nnota/currency_laravel_api_vueJs

Wireframe de la partie front de l'administration