

Documentation projet architecture logicielle

1. Cas d'utilisation

Cas d'utilisation simples :

Se connecter

Consulter ses comptes.

Connaitre son solde.

Faire un virement vers un compte interne.

Faire un virement vers un compte externe (simuler le bon fonctionnement du virement).

Cas d'utilisation spécifiques :

Gérer le cas où le solde serait insuffisant pour effectuer un virement.

Gérer l'erreur d'identifiants.

2. Front-end

Pour la partie front-end nous avons choisi d'utiliser la technologie React.

La library 'axios' est utilisée pour envoyer les requêtes à l'API.

La library '@testing-library/react' est utilisée pour faire des tests unitaire.

La library 'fontawesome' a été utilisée pour récupérer des icones.

La library 'tailwind css' a été utilisée pour récupérer des éléments de design.

La library '@reach/router' a été utilisée pour gérer les autorisations d'accès aux routes.

La library 'msw' a servi a mocker des données pour les tests.

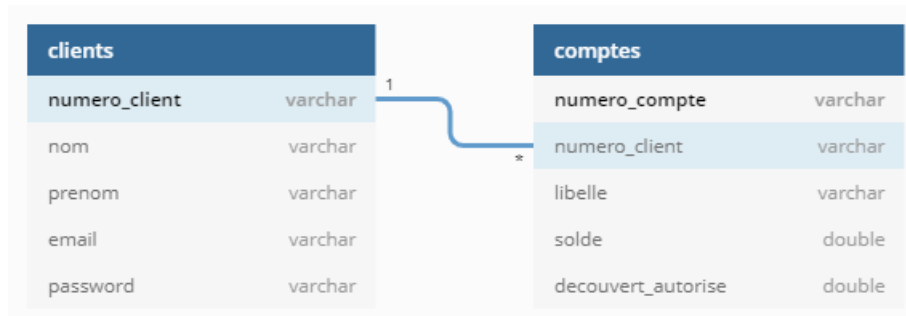
Voici la forme que la requête de login envoyée à l'API doit avoir :

```
1 {  
2   ... "email": "test@test.fr",  
3   ... "password": "test"  
4 }
```

3. Back-end et API

Pour la partie back-end et API, nous avons choisi d'utiliser la technologie Springboot.

4. Base de données



5. Guide d'installation

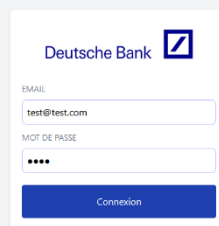
- Commencez par cloner le repository git avec la commande suivante :


```
git clone https://github.com/Ustalioth/Projet_AL_MNS.git
```

- Importez la base de données dans PostgreSQL depuis le fichier suivant : « export.sql »
- Installez le JDK 17 et assurez-vous d'être en Maven version 17
- Placez vous à la racine de la partie back-end du projet et lancez la commande :
« mvn spring-boot:run »
- Placez-vous à la racine de la partie front-end du projet et lancez la commande :
« npm i » ou « yarn install » (selon votre gestionnaire de packages)
- Modifier si besoin dans le fichier "/src/services/axios.js" l'url de l'api.
- Lancer l'application avec la commande « yarn start », l'application se lance sur l'adresse suivante : « http://localhost:3000 »

6. Guide d'utilisation

Tout d'abord, entrez vos identifiants et cliquez sur le bouton « Connexion »



Deutsche Bank 

EMAIL

MOT DE PASSE

[Connexion](#)

Vous pouvez ensuite consulter vos comptes, le solde de chacun d'entre eux est affiché sur la droite.

En cliquant sur « Virement » il vous est possible d'effectuer des virements depuis votre compte vers d'autres comptes.

En cliquant sur « Envoyer de l'argent », vous arriverez également sur la page de virement mais le compte correspondant sera sélectionné en tant que source d'approvisionnement.

Deutsche Bank

Mes comptesVirements

MES COMPTES

LIVRET A

N° 2

Envoyer de l'argent

SOLDE

890 €

COMPTE COURANT

N° 1

Envoyer de l'argent

SOLDE

3438 €

COMPTE AUX ILES CAIMAN

N° 3

Envoyer de l'argent

SOLDE

50 €

Sur la page « Virements », il est possible de sélectionner le compte source, le compte cible ainsi que le montant à transférer.

Dans le cas d'un virement interne (après avoir coché la case « Virement interne »), la liste des comptes apparaîtra. Pour un virement externe il faudra entrer l'id du compte cible.

Deutsche Bank

Mes comptesVirements

VIREMENTS

Compte source

Livret A - 890 €

Montant

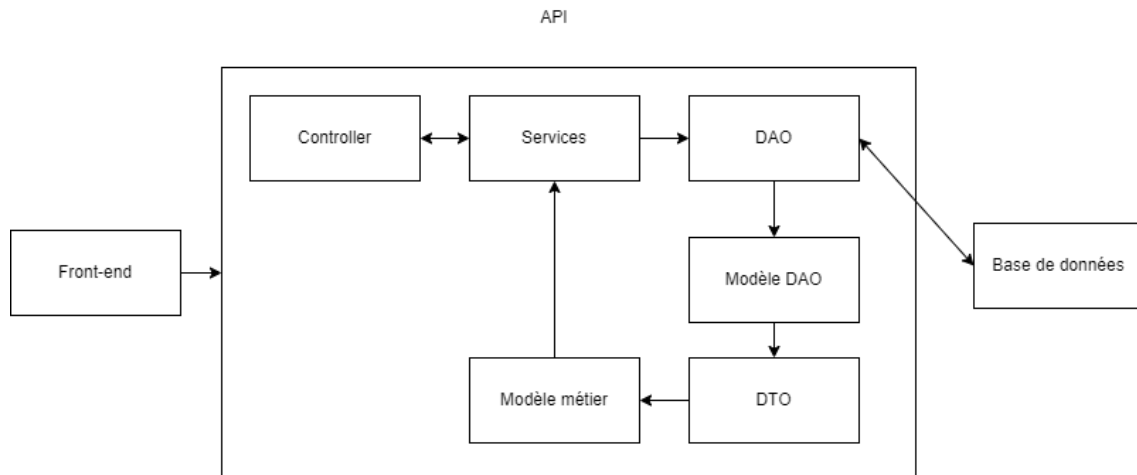
→

☐ Virement interne

Compte cible externe

Valider

7. Documentation technique



Front-end : Interface graphique permettant à l'utilisateur d'interagir avec la partie back-end du programme.

API : Partie du programme servant à faire le lien entre le front-end et la base de données en récupérant et en formatant les données à afficher avant de les transmettre au front-end.

Controller : Composant de l'API qui permet de gérer les routes et de contrôler l'accès à ces dernières.

Services : Implémente la partie logique de l'API.

DAO (Decentralized Autonomous Organization) : C'est la partie de l'API qui va communiquer directement avec la base de données.

Modèle DAO : Sert à rendre la DAO indépendante de la partie métier.

DTO (Data Transfer Object) : Sert à transformer un modèle DAO en modèle métier.

Modèle métier : Représentation concrète des entités de l'application.

8. Diagramme de classes

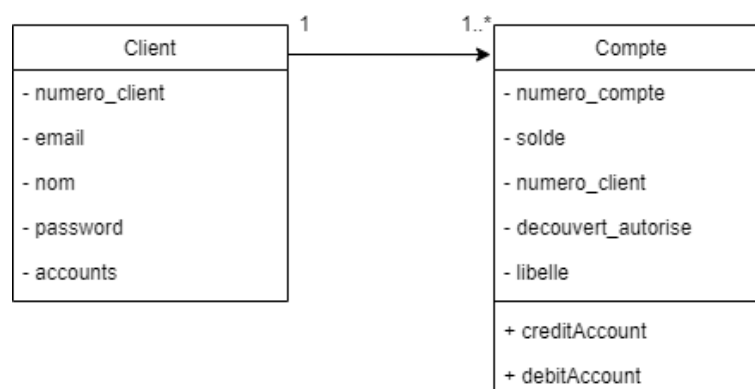


Table des matières

Documentation projet architecture logicielle.....	1
1. Cas d'utilisation	1
Cas d'utilisation simples :	1
Cas d'utilisation spécifiques :	1
2. Front-end.....	1
3. Back-end et API	2
4. Base de données	2
5. Guide d'installation	3
6. Guide d'utilisation	3
7. Documentation technique	5
8. Diagramme de classes.....	5