

# **TRAVAUX PRATIQUES PROGRAMMATION JAVA EE**

**Etudiants : GLSI A & B**

**Année scolaire : 2025-2026**

**Date de validation : 18/01/2026**

**Indication** : le travail va se faire par groupe de 3 étudiants. Chaque groupe créera un compte GitHub et enverra son travail sur ce dépôt. Le nom du compte sera libellé comme suit : **TP\_JEE\_GLSI(A ou B)\_GROUPE\_NuméroGroupe\_2026**. Afin d'évaluer la participation effective de tous les membres du groupe, une vérification sera faite par rapport aux différents commit réalisés par ces derniers.

## **Enoncé**

La société bancaire « **Ega** » voudrait mettre en place un système de gestion de ses clients et des comptes que ces derniers possèdent. Un client peut avoir plusieurs comptes.

La banque met à disposition des clients deux types de comptes : un compte épargne et un compte courant.

Un compte est caractérisé par le numéro de compte, le type de compte, la date de création du compte, le solde du compte et le propriétaire du compte.

Pour le client on retiendra les informations suivantes : nom, prénom, date de naissance, sexe, adresse, numéro de téléphone, courriel et nationalité.

Afin de formater les numéros de compte selon les formats standards, il vous est conseillé d'utiliser la dépendance iban4j.

Le numéro de compte est unique pour un compte.

Lors de la création d'un compte, son solde est nul.

Le client peut faire plusieurs transactions sur son compte (opération de dépôt, versement et virement d'un montant d'un compte vers un autre compte)

## **TAF** :

### **A. Coté back-end**

1. Mettre en place une API CRUD pour gérer les comptes et les clients.
2. Ajouter les possibilités pour un client de :
  - a. Faire un versement sur son compte ;
  - b. Faire un retrait sur son compte si le solde le permet ;
  - c. Faire un virement d'un compte à un autre
3. Afficher toutes les transactions effectuées sur un compte au cours d'une période donnée ;
4. Donner la possibilité au client d'imprimer son relevé ;
5. Afin d'assurer la fiabilité des informations, vous devez prévoir les validateurs et un gestionnaire global d'exception pour toutes les informations que vous saisissez ;
6. Un test est requis dans l'outil postman afin de s'assurer que toutes les APIs fonctionnent correctement.

## **B. Côté front-end**

Mettre en place grâce au framework angular, des interfaces ergonomiques et conviviales permettant d'utiliser efficacement toutes les APIs du back-end back-end.

## **C. Sécurité de l'application**

Pour effectuer n'importe quelle action sur le système, l'utilisateur doit préalablement s'authentifier. Vous devez mettre en place un système d'authentification permettant de sécuriser votre application. Vous pouvez par exemple utiliser spring security et l'authentification jwt afin de garantir la sécurité de votre système.

**Bon travail**