***Fortis Bank*** vous a embauché pour mettre en place un système d'application bancaire pour gérer les clients et leurs comptes, ainsi que les transactions des clients.

Un client a un numéro de client, un nom, un e-mail, ... (cette liste n'est pas exhaustive). Un client doit avoir au moins un compte courant.

Un client peut ouvrir un compte d'épargne, un compte de crédit et un compte en devises. Un client peut fermer un compte à tout moment (sauf le compte courant).

Un compte est identifié par un numéro de compte, un type de compte, une date d'ouverture, un solde disponible.

Un compte a une liste de transactions.

Un client a deux types de comptes : compte courant et compte d'épargne. Il peut également avoir un compte de crédit.

Un compte courant a un nombre limité de transactions gratuites (seulement 3 transactions gratuites par mois) et des frais supplémentaires.

Un compte d'épargne a : un taux d'intérêt annuel, un gain annuel.

Une transaction est définie par un numéro de transaction, une description, une date de transaction, un montant et un type de transaction (dépôt, retrait).

Le directeur de la banque peut créer un client, ouvrir un compte au client. Il peut également fermer les comptes d'un client, puis supprimer un client.

Vous devez définir les caractéristiques et les comportements de chaque objet dans le système bancaire.

## **Partie I : Analyse**

1. Définir le domaine du système en fonction des règles métier.
2. Concevoir les diagrammes UML (diagramme de classe et cas d'utilisation)
3. Concevoir les diagrammes de séquence et d'activité UML.

Partie II : Implémentation Backend

4. Mettre en œuvre la couche métier.

1. Générer l'API du système bancaire

Exigences : Utiliser les piliers (les concepts fondamentaux) des concepts orientés objet pour mettre en œuvre ce système d'information.

Abstraction de données, Héritage, Encapsulation, Polymorphisme (polymorphisme statique et dynamique), Interface