# Analyse de domaine de système bancaire.

Domaine du système en fonction des règles métier : Le système bancaire doit être capable de gérer les clients, leurs comptes et les transactions des clients. Les clients peuvent avoir différents types de comptes, y compris des comptes courants, des comptes d'épargne et des comptes de crédit. Les comptes ont des caractéristiques différentes telles que le numéro de compte, le type de compte, la date d'ouverture, le solde disponible et une liste de transactions. Les transactions ont des informations telles que le numéro de transaction, la description, la date de transaction, le montant et le type de transaction. Les clients peuvent fermer des comptes à tout moment, sauf le compte courant, et les comptes courants ont un nombre limité de transactions gratuites par mois avec des frais supplémentaires pour les transactions supplémentaires. Le système doit être capable de créer des clients, d'ouvrir des comptes pour les clients, de fermer les comptes des clients et de supprimer les clients.

### Diagrammes UML :

Diagramme de classes : Le diagramme de classes doit inclure les classes suivantes : Client, Compte, CompteCourant, CompteEpargne, CompteCredit et Transaction. La classe Client doit avoir des attributs tels que le numéro de client, le nom et l'e-mail, ainsi que des méthodes pour ajouter et supprimer des comptes. La classe Compte doit avoir des attributs tels que le numéro de compte, le type de compte, la date d'ouverture et le solde disponible, ainsi que des méthodes pour effectuer des transactions. Les classes CompteCourant, CompteEpargne et CompteCredit doivent hériter de la classe Compte et ajouter des attributs et des comportements spécifiques à chaque type de compte. La classe Transaction doit avoir des attributs tels que le numéro de transaction, la description, la date de transaction, le montant et le type de transaction, ainsi que des méthodes pour récupérer et modifier ces attributs.

Diagramme de cas d'utilisation : Le diagramme de cas d'utilisation doit inclure des cas d'utilisation tels que "Créer un client", "Ouvrir un compte", "Fermer un compte", "Effectuer une transaction" et "Supprimer un client".

### Diagrammes de séquence et d'activité UML :

### Diagramme de séquence : Le diagramme de séquence doit montrer les interactions entre les acteurs et les objets du système, y compris la création de clients, l'ouverture et la fermeture de comptes, et les transactions.

### Diagramme d'activité : Le diagramme d'activité doit montrer les étapes nécessaires pour effectuer des actions telles que l'ouverture d'un compte, la fermeture d'un compte et l'exécution d'une transaction.

Couche métier : La couche métier doit être mise en œuvre en utilisant les piliers de la programmation orientée objet tels que l'abstraction de données, l'héritage, l'encapsulation, le polymorphisme statique et dynamique, ainsi que des interfaces pour assurer la cohérence et la flexibilité du système.API du système bancaire : L'API du système bancaire doit être générée pour permettre aux utilisateurs d'interagir avec le système en utilisant des requêtes pour créer des clients, ouvrir des comptes, fermer des comptes, effectuer des transactions et supprimer des clients. L'API doit être sécurisée pour protéger les données des clients et le système bancaire contre les attaques malveillantes.

En résumé, pour mettre en place le système bancaire pour Fortis Bank, il est important de comprendre les règles métier et de concevoir des diagrammes UML tels que le diagramme de classes et le diagramme de cas d'utilisation. Ensuite, il faut mettre en œuvre la couche métier en utilisant les piliers de la programmation orientée objet et générer l'API pour permettre aux utilisateurs d'interagir avec le système en toute sécurité.