

# Documentation Technique V1

# Sommaire

Documentation Technique V1 .....	1
1. Répartition des tâches et état d'avancement .....	3
2. Présentation de l'application .....	4
2.1. Use Case global de l'application .....	5
2.2. JavaDoc de l'application .....	6
2.3. Diagramme de classe de la base de données .....	7
3. Architecture .....	8
3.1. Architecture générale (poste client, serveurs, ...) et rôle de chaque élément .....	8
3.2. Ressources externes (.jar, ...) utilisées et rôles .....	8
4. Installation de l'application .....	8
4.1. Structuration en packages de l'application documentée. Principes retenus pour cette structuration .....	9
5. Fonctionnalités .....	10
5.1. Modifier les informations d'un client .....	10
5.2. Créer un nouveau client .....	10
5.3. Consulter un compte .....	11
5.4. Débitier un compte .....	12
5.5. Rendre inactif un client .....	12
5.6. Créditer un compte (Escassut Sean) .....	12
5.7. Cloturer un compte (Mabille Matthis) .....	13
5.8. Créer un compte (Mabille Matthis) .....	13
5.9. Effectuer des virements compte à compte (Wendjane Hugo) .....	14
5.10. Gérer les employés, ou faire le CRUD (Testa Thomas): .....	15
5.10.1. Créer un employé .....	15
5.10.2. Consulter un employé .....	15
5.10.3. Modifier un employé .....	15
5.10.4. Supprimer un employé .....	15

# 1. Répartition des tâches et état d'avancement

Version	User	Use Case	Cas d'utilisation	Sean Escassut	Matthis Mabilie	Hugo Wendjane h	Thomas Testa
1	Chef d'Agence	CRUD employé	Create				
1	Chef d'Agence	CRUD employé	Read				
1	Chef d'Agence	CRUD employé	Update				
1	Chef d'Agence	CRUD employé	Delete				
1	Guichetier	Compte	Créer				
1	Guichetier	Compte	Créditer/Débit				
1	Guichetier	Compte	Effectuer un virement				
1	Guichetier	Compte	Clôturer				
2	Chef d'Agence	Compte	Débit exceptionnel				
2	Chef d'Agence	Emprunt	Simuler emprunt				
2	Chef d'Agence	Emprunt	Simuler assurance				
2	Guichetier	Compte	Relevé PDF				
2	Guichetier	CRUD Prélèvement	Create				
2	Guichetier	CRUD Prélèvement	Read				
2	Guichetier	CRUD Prélèvement	Update				

Version	User	Use Case	Cas d'utilisation	Sean Escassut	Matthis Mabilie	Hugo Wendjane h	Thomas Testa
2	Guichetier	CRUD Prélèvement	Delete				
2	Batch	Prélèvements automatiques					
2	Batch	Relèves mensuels					

F pour Fonctionnel

D pour en Développement

## 2. Présentation de l'application

L'application permet de gérer des comptes bancaires de dépôt pour des clients. Elle permet de débiter, créditer un compte soit par virement c'est à dire un échange d'argent entre deux comptes distincts mais domiciliés dans le réseau bancaire, soit par une personne physique se présentant devant un guichet.

## 2.1. Use Case global de l'application

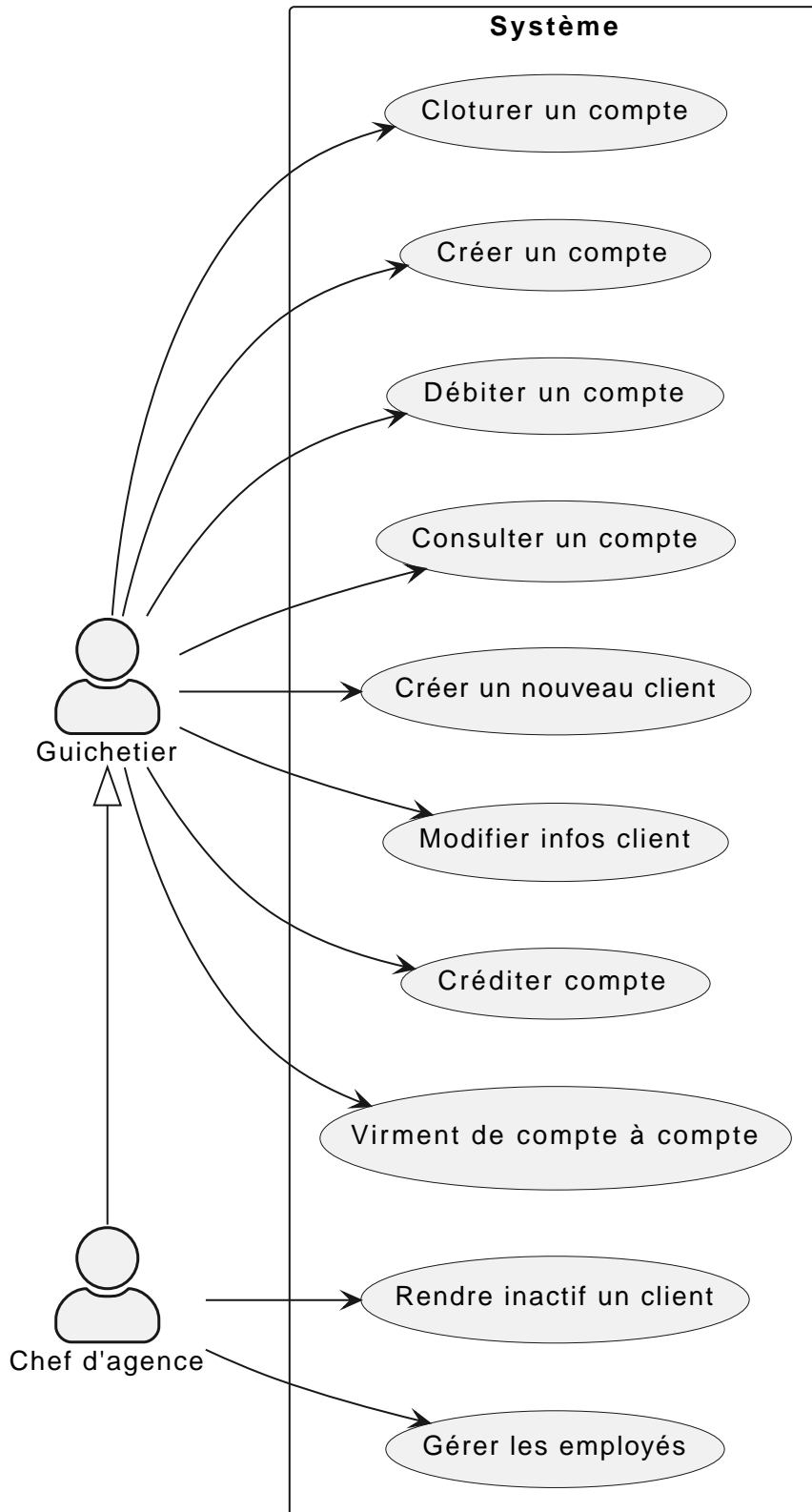


Figure 1. Use Case global de l'application

Dans le diagramme des Use Case ci-dessus, nous pouvons observer qu'il y a deux types d'utilisateurs présents dans l'application, le guichetier et le chef d'agence.

Le guichetier est l'employé "de base" du système. Il peut réaliser certaines opérations:

- Modifier les informations client (adresse, téléphone, ...)
- Créer un nouveau client
- Consulter un compte
- Débitier un compte (BD)
- Créditer un compte
- Cloturer un compte
- Créer un compte
- Effectuer des virements compte à compte

Le chef d'agence est un type d'utilisateur héritant de guichetier, il peut effectuer toutes les opérations de celui-ci en plus de:

- rendre inactif un client.
- Gérer les employés, ou faire le CRUD (Create Read Update Delete)
  - Créer un employé
  - Consulter un employé
  - Modifier un employé
  - Supprimer un employé

## 2.2. JavaDoc de l'application

Lien vers la [liendelajavadocdegithub.com\[javadoc\]](https://liendelajavadocdegithub.com/javadoc)

## 2.3. Diagramme de classe de la base de données

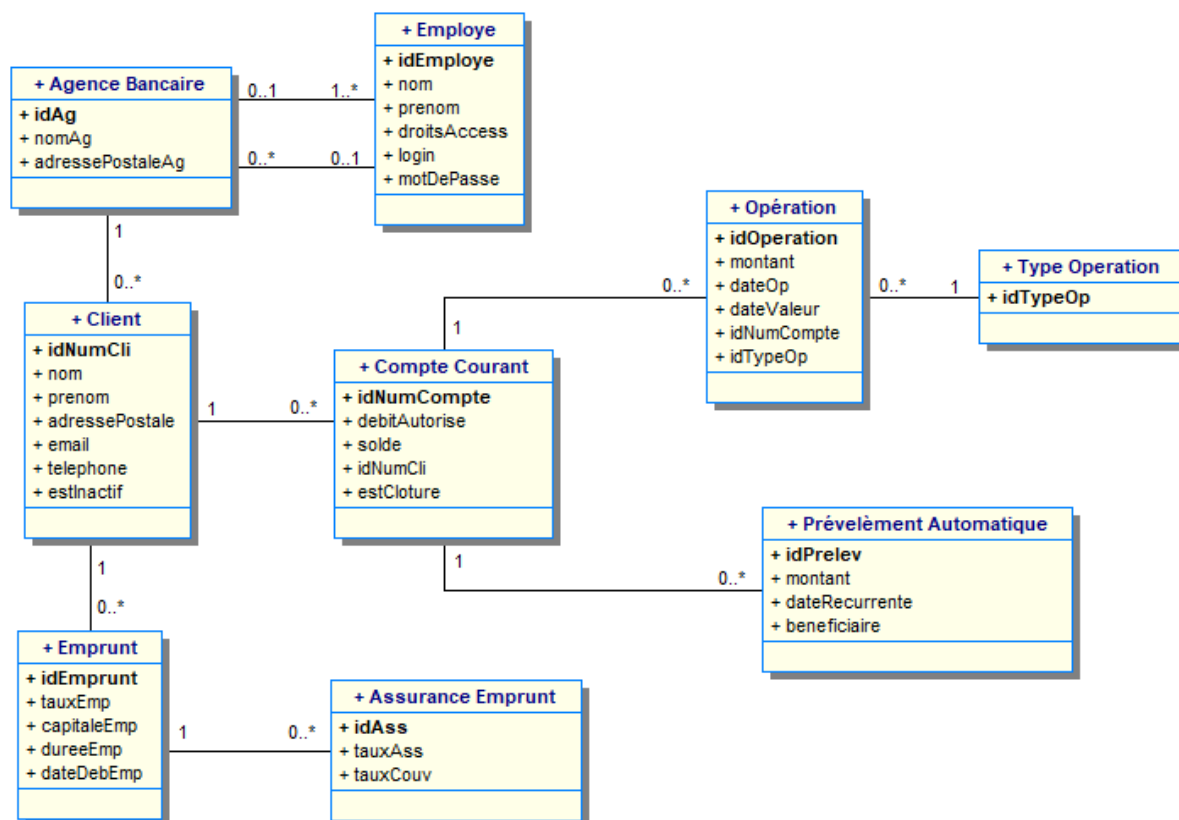


Figure 2. Diagramme de classe UML de la base de données

- Une agence bancaire est composée de plusieurs employés.
- Un employé peut être chef d'agence d'une ou plusieurs agences.
- Un employé peut être un chef d'agence ou un guichetier selon les droits d'accès qu'il possède
- Une agence bancaire est composée de plusieurs clients.
- Un client possède des informations qu'il est possible de modifier.
- Un client peut devenir inactif.
- Un client peut effectuer des opérations sur un compte par l'intermédiaire des employés (débit, crédit, emprunt).
- Un compte contient des informations auxquelles il est possible d'accéder.
- Un client peut avoir plusieurs comptes.
- Un emprunt peut être assuré.

Pour l'instant l'application ne possède que certaines des fonctionnalités du diagramme ci-dessus.

## 3. Architecture

### 3.1. Architecture générale (poste client, serveurs, ...) et rôle de chaque élément

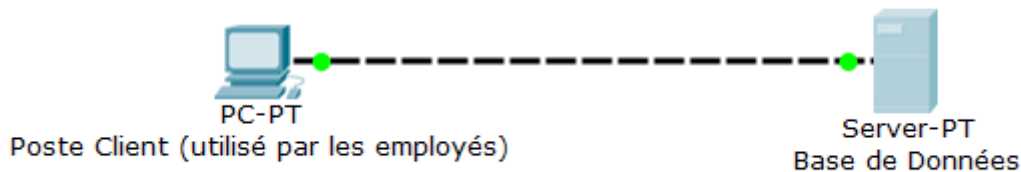


Figure 3. Schéma de l'architecture du fonctionnement de l'application

Les postes client exécutent l'application faite en javaFX et se connectent à la base de données oracle, permettant de manipuler les données des clients et des employés.

### 3.2. Ressources externes (.jar, ...) utilisées et rôles

L'application utilise la librairie JavaFX pour permettre de disposer d'une interface graphique.

Elle utilise également la librairie externe ojdbc6.jar pour interagir avec la base de données du système des agences bancaires et exécuter des instructions SQL avec Java.

## 4. Installation de l'application

Pré-requis :

- La version 1.8 de [Java](#)
- Les deux fichiers suivants : [Fichier 1](#) et [Fichier 2](#)
- L'application en .jar : [lien](#)

*Remarque : les deux éléments ci-dessus doivent se trouver dans le même dossier pour plus de simplicité*

Installation de l'application :

- Ouvrir l'application "Powershell"
- Vérifier votre version de Java installée à l'aide de la commande suivante : `java -version`  
*Remarque : Si la version installée est Java 1.8.\*; vous devriez pouvoir double cliquer sur le .jar que vous venez de télécharger*
- Se déplacer dans le workspace de l'application avec la commandes suivantes : `cd .|Téléchargements|`  
*Remarque : Si la version installée est Java 1.8.\*; vous pouvez ignorer les trois étapes suivantes.*



Sinon :

- Exécuter ensuite la commande suivante :  
`cd setenv-powershell.ps1 ou cd .\setenv-powershell.ps1`
- Tapez ensuite la commande :  
`cmd`
- Et vérifier maintenant votre version de Java :  
`java -version`

Vous pouvez maintenant lancer l'application à l'aide de la commande suivante :

`java -jar DailyBankV0.jar`

## 4.1. Structuration en packages de l'application documentée. Principes retenus pour cette structuration

L'application est structurée en plusieurs packages. Le modèle retenu est le modèle est le modèle MVC (modèle, view, controller). De plus on regroupe les packages en deux catégories:

*Application :*

- Package tools : Contient contenant les outils de l'application.
- Package view : Contient les classes de l'interface graphique
- Package control : Contient les fonctionnalités de l'application

Cette catégorie contient les données des différentes classes pour pouvoir utiliser les différentes fonctionnalités de l'application.

*Model :*

- Package data : Contient les classes de données.
- Package orm : Contient les classes de gestion des données.
- Package orm.exception : Contient les classes d'exceptions.

Il contient des classes de base de données correspondant à certaines tables de la base de données. Il permet également de manipuler les données de la base de données (classe orm).

## 5. Fonctionnalités

### 5.1. Modifier les informations d'un client

Partie "modifier infos client" du diagramme des use case

Partie du diagramme de classes données nécessaires :

- En lecture : client
- En mise à jour : client

cf. doc. utilisateur "Comment modifier les informations personnelles d'un client ?"

Diagramme de séquence:

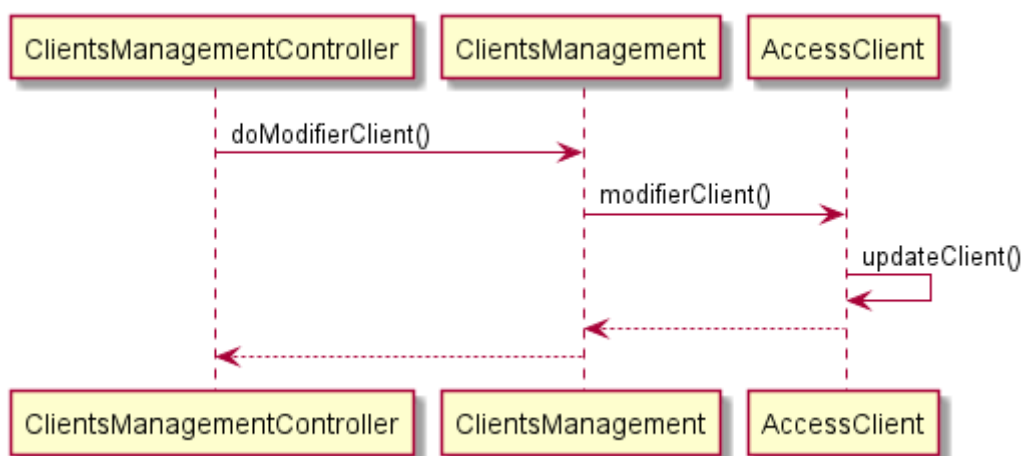


Figure 4. Diagramme de séquence pour la modification d'un client

### 5.2. Créer un nouveau client

Partie "créer un nouveau client" du diagramme des use case

Partie du diagramme de classes données nécessaires :

- En lecture : Client
- En mise à jour : Client

cf. doc. utilisateur "Comment ajouter un client ?"

Diagramme de séquence :

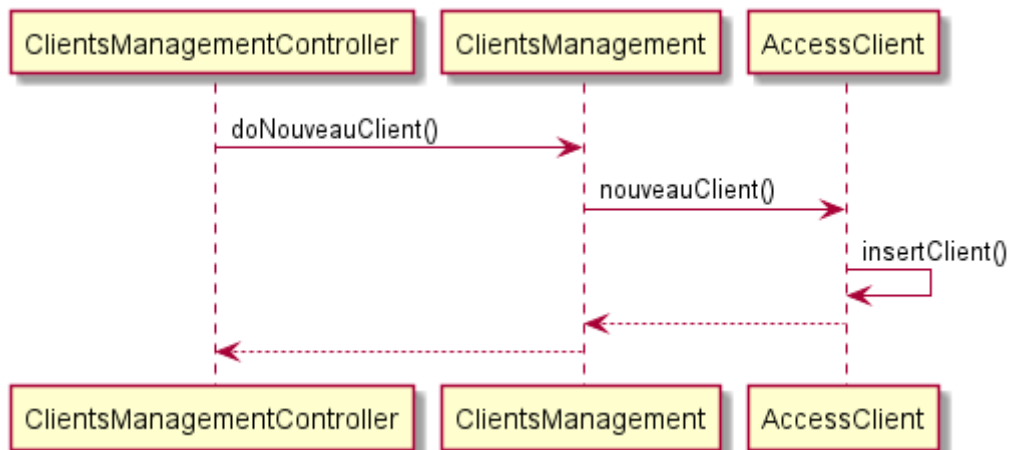


Figure 5. Diagramme de séquence pour la création d'un client

## 5.3. Consulter un compte

Partie "consulter un compte" du diagramme des UC

Partie du diagramme de classes données nécessaires :

- En lecture : Client, Compte Courant, Opération, Type Opération

cf. doc. utilisateur "Comment accéder aux comptes d'un client ?"

Diagramme de séquence :

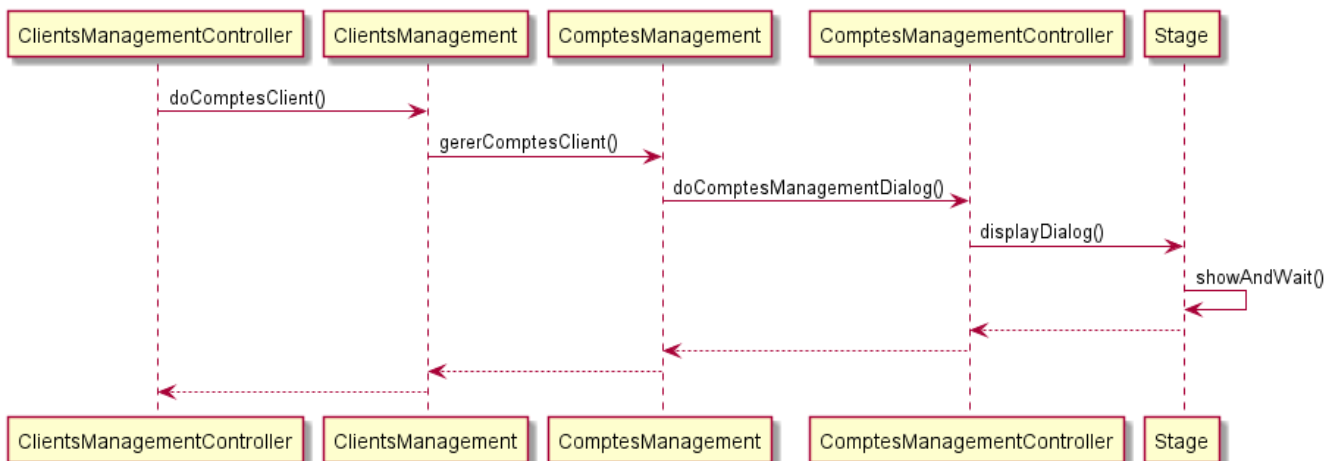


Figure 6. Diagramme de séquence pour la consultation d'un compte

## 5.4. Débiter un compte

Partie "Débiter un compte" du diagramme des UC

Partie du diagramme de classes données nécessaires :

- En lecture : Client, Compte Courant, Opération, Type Opération
- En mise à jour : Opération

cf. doc. utilisateur "Comment enregistrer un débit manuellement ?"

## 5.5. Rendre inactif un client

Partie "rendre inactif un client" du diagramme des UC

Partie du diagramme de classes données nécessaires :

- En lecture : Client
- En mise à jour : Client

cf. doc. utilisateur "Comment modifier les informations personnelles d'un client ?"

## 5.6. Créditer un compte (Escassut Sean)

Partie "Créditer un compte" du diagramme des UC

Partie du diagramme de classes données nécessaires :

- En lecture : Client, Compte Courant, Opération, Type Opération
- En mise à jour : Opération

cf. doc. utilisateur "Comment enregistrer un crédit manuellement ?"

Diagramme de séquence:

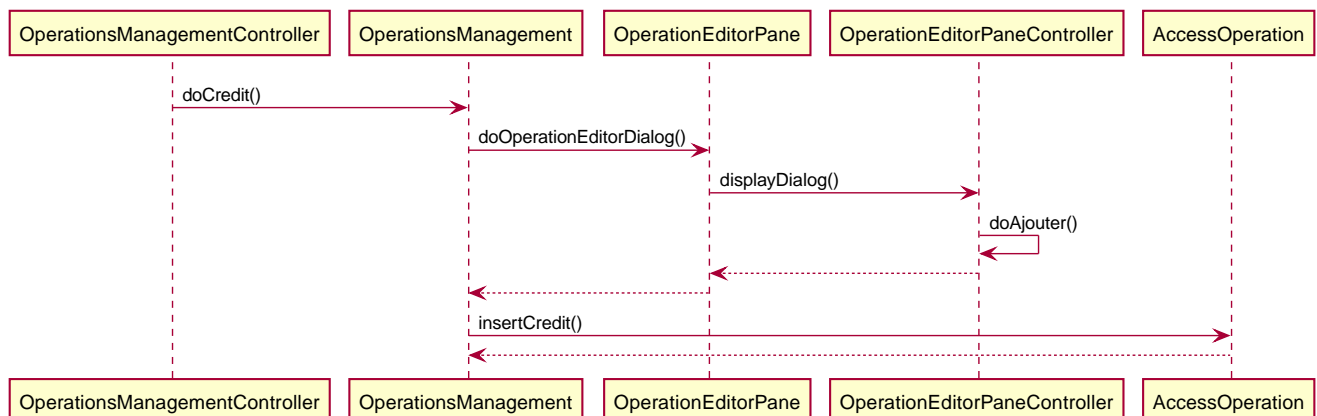


Figure 7. Diagramme de séquence pour le crédit d'un compte

## 5.7. Cloturer un compte (Mabille Matthis)

Partie "cloturer un compte" du diagramme des UC

Partie du diagramme de classes données nécessaires :

- En lecture : Client, Compte Courant
- En mise à jour : Client, Compte Courant

cf. doc. utilisateur "Comment cloturer un compte client déjà existant ?"

Diagramme de séquence :

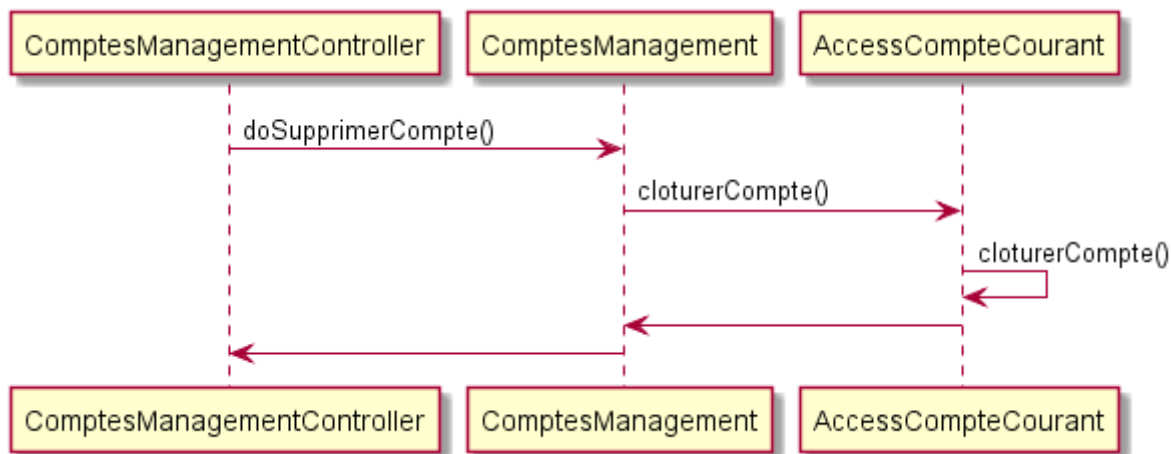


Figure 8. Diagramme de séquence pour la cloturation d'un compte

## 5.8. Créer un compte (Mabille Matthis)

Partie "créer un compte" du diagramme des UC

Partie du diagramme de classes données nécessaires :

- En lecture : Client, Compte Courant
- En mise à jour : Client, Compte Courant

cf. doc. utilisateur "Comment créer un nouveau compte client ?"

Diagramme de séquence :

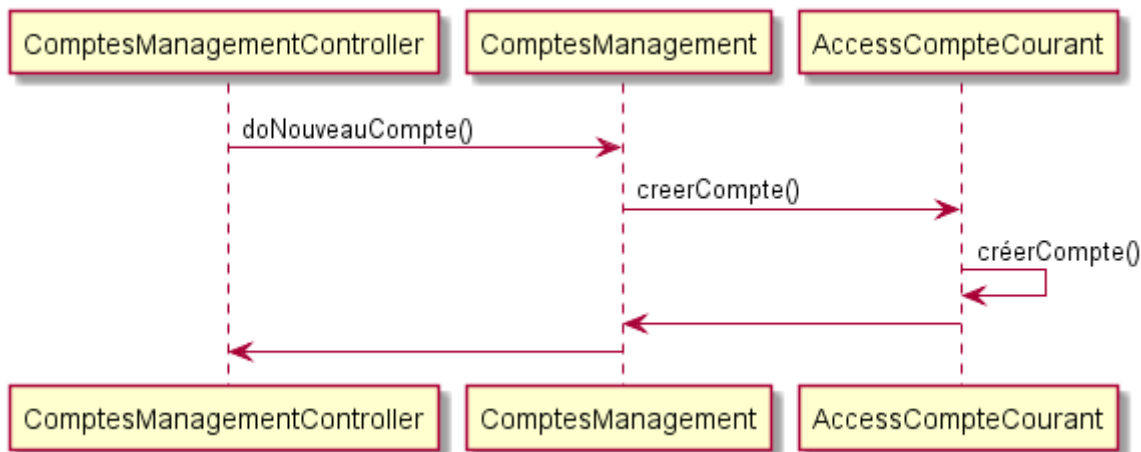


Figure 9. Diagramme de séquence pour la création d'un compte

## 5.9. Effectuer des virements compte à compte (Wendjaneh Hugo)

Partie "Virement de compte à compte" du diagramme des UC

Partie du diagramme de classes données nécessaires :

- En lecture : Client, Compte Courant, Opération, Type Opération
- En mise à jour : Opération

cf. doc. utilisateur "Comment effectuer un virement ?"

Diagramme de séquence:

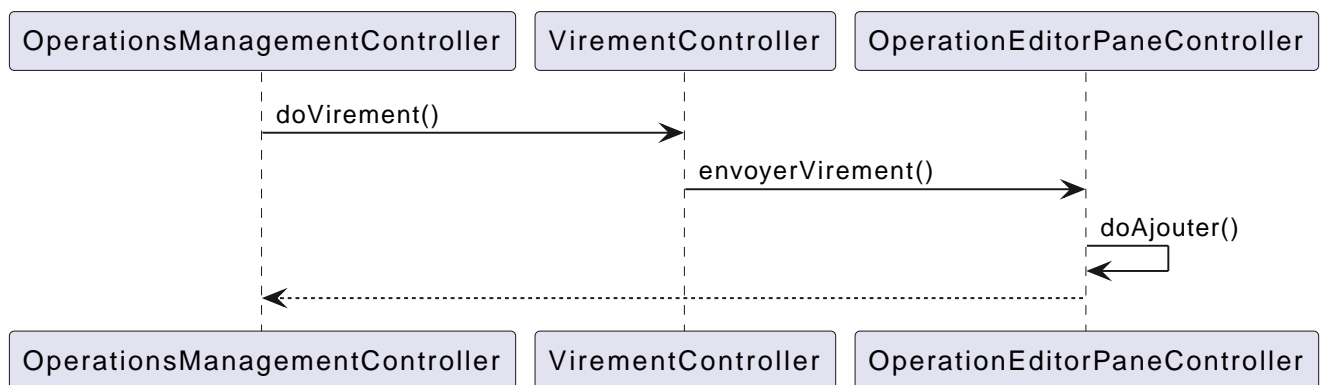


Figure 10. Diagramme de séquence pour le virement d'un compte à un autre

## **5.10. Gérer les employés, ou faire le CRUD (Testa Thomas):**

*En cours de développement*

### **5.10.1. Créer un employé**

### **5.10.2. Consulter un employé**

### **5.10.3. Modifier un employé**

### **5.10.4. Supprimer un employé**