

REPUBLICQUE DU SENEGAL

UN PEUPLE-UN BUT-UNE FOI

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA FORMATION PROFESSIONNELLE



INSTITUT SUPERIEUR  
D'INFORMATIQUE

*L'institut de référence dans les TIC*

## MEMOIRE DE FIN DE CYCLE

EN VUE DE L'OBTENTION DU DIPLOME D'INGENIEUR EN METHODES  
INFORMATIQUES APPLIQUEES A LA GESTION DES ENTREPRISES

**SUJET :**

INFORMATISATION DES PROCEDURES DE  
GESTION D'UN SYSTEME FINANCIER  
DECENTRALISE: CAS DU PAMECAS

**Présenté par :**

**M. Papa Moussa Cissé**

**Directeur de mémoire :**

**M. Niasse**

**Année académique : 2013-2014**

## **TABLE DES MATIERES**

REMERCIEMENTS.....	Erreur ! Signet non défini.
AVANT PROPOS .....	4
INTRODUCTION .....	5
PREMIERE PARTIE : CADRE THEORIQUE ET PRESENTATION DU PROJET .....	6
CHAPITRE 1 : CADRE THEORIQUE.....	6
I. Présentation du domaine de l'étude et de la problématique .....	6
II. Objectifs de l'étude .....	7
1. Objectif général.....	7
2. Objectifs spécifiques .....	7
CHAPITRE 2: PRESENTATION DU PROJET .....	8
I. Spécifications des besoins fonctionnels .....	8
II. Délimitation du champ ou domaine de l'étude .....	8
1. Délimitation procédurale.....	8
2. Délimitation organisationnelle.....	8
DEUXIEME PARTIE : ANALYSE ET CONCEPTION FUTURE .....	9
CHAPITRE 1 : ANALYSE (ETUDE) DE LA SITUATION ACTUELLE .....	9
CHAPITRE 2 : CONCEPTION .....	12
I. Diagramme de cas d'utilisation .....	12
1. Définition et formalisme .....	12
2. Exemple de diagramme de cas d'utilisation .....	13
II. Le diagramme de séquence.....	13
1. Définition et formalisme.....	13
2. Exemple de diagramme de séquence.....	15
III. Schémas des diagrammes de cas d'utilisation et de séquence .....	16
IV. Diagramme de classe.....	26
1. Définition et formalisme.....	26
2. Schéma du diagramme de classe .....	30
3. Schéma relationnel de la base de données .....	33
CHAPITRE 3 : SOLUTIONS TECHNIQUES .....	35
I. Présentation de l'architecture 3-tiers.....	35
1. Présentation de l'atelier de développement.....	35
2. Présentation de la base de données: SQL-SERVEUR .....	36
3. Présentation de JEE .....	37
II. Le diagramme de déploiement.....	39

1. Définition et formalisme.....	39
2. Schéma du diagramme de déploiement.....	40
TROISIEME PARTIE : IMPLEMENTATION DE LA SOLUTION .....	41
CHAPITRE1 : IMPLEMENTATION DE LA BASE DE DONNEES .....	41
CHAPITRE 2 : LES INTERFACES USERS.....	42
CHAPITRE 3 : LES SCRIPTS D'IMPLEMENTATION .....	45
I. Script d'implémentation de la base de données .....	45
II. L'agent SQL-Serveur .....	61
CONCLUSION .....	62
ANNEXES .....	63

## REMERCIEMENTS

Je veux adresser toute ma reconnaissance et mes sincères remerciements à :

- Monsieur le directeur général du groupe ISI, Mr Abdou Samb et l'ensemble du personnel de l'Institut Supérieur d'Informatique.
- Monsieur Niasse pour ses conseils et son attention tout au long de la rédaction de ce document.
- A mes parents qui m'ont soutenu tout au long de mes études.

## AVANT PROPOS

L'Institut Supérieur d'informatique est un établissement d'enseignement supérieur avec une expérience de plus de dix ans dans la formation professionnelle et particulièrement dans des domaines tels que l'informatique et les sciences de gestion.

Dans le processus d'acquisition de connaissances et de savoir-faire est inscrit pour chaque étudiant une étape d'imprégnation en entreprise, d'incubation et de stage pendant laquelle il sera mise en situation réelle pour confronter le savoir acquis à la réalité de l'entreprise.

C'est dans ce contexte que nous a été soumis un sujet de mémoire visant à présenter l'application de technologies informatiques qui peuvent améliorer la qualité de travail d'une entreprise.

## INTRODUCTION

Depuis les années 50, le monde économique a enrichi son jargon par l'expression pays du tiers-monde. Ce qualificatif s'attribue à tous les pays dont l'économie est précaire. Les organismes de développement tels que l'ONU, l'UNICEF et la FMI se sont engagés dans la lutte contre la pauvreté en mettant en place des programmes de développement. Dans la même optique, les institutions de microfinance ont été créées dans le but de lutter contre la pauvreté ainsi que participer au développement des pays du tiers-monde.

Pour la bonne marche d'un système financier décentralisé, il faut que l'information soit disponible, donc il est important d'avoir un bon système d'information et de gestion. Les besoins des institutions en système d'information varient en taille et en complexité. Ces différences relèvent de divers éléments organisationnels, comme le volume des transactions, la méthodologie, l'environnement réglementaire, l'infrastructure, et d'un désir général de changer ainsi que des ressources disponibles. Il est donc important de savoir comment concevoir et réaliser un système d'information, d'où l'intérêt du projet qui sera étudié : « INFORMATISATION DES PROCEDURES DE GESTION D'UN SYSTEME FINANCIER DECENTRALISE: CAS DU PAMECAS ».

Tout au long du développement de ce sujet, des techniques informatiques seront expliquées et appliquées afin de montrer comment obtenir la bonne information en temps réel.

La partie développement du sujet va prendre le plan suivant :

La première partie est intitulée cadre théorique et présentation du projet.

La deuxième partie est appelée analyse et conception future.

La troisième partie est l'implémentation de la solution.

# **PREMIERE PARTIE : CADRE THEORIQUE ET PRESENTATION DU PROJET**

## **CHAPITRE 1 : CADRE THEORIQUE**

### **I. Présentation du domaine de l'étude et de la problématique**

Le Partenariat pour la mobilisation de l'épargne et du crédit au Sénégal (PAMECAS) est fort de seize années d'expérience dans le domaine de la microfinance. Aujourd'hui, elle compte plusieurs agences dans les différentes régions du Sénégal.

Le PAMECAS présente les objectifs suivants :

- Faciliter l'accès à l'Épargne et au Crédit, quels que soient les moyens des souscripteurs et leur secteur d'activité;
- Promouvoir l'amélioration du bien-être économique et social de ses membres et des communautés de base dans un esprit de solidarité, de responsabilité et de gestion démocratique par :
  - Le développement de Mutuelles d'Épargne et de Crédit viables;
  - La mobilisation de l'épargne locale et la recherche de ressources extérieures;
  - La mise en œuvre de politiques administratives, comptables et financières performantes dans le respect de la réglementation en vigueur;
  - La promotion et la gestion rationnelle de services financiers accessibles et adaptées;
  - La formation permanente de qualité du personnel et des dirigeants;
  - La coopération locale, sous-régionale et internationale;
  - Une mutuelle de santé qui se propose de mener dans l'intérêt des membres et de leur famille, une action de prévoyance, d'entraide et de solidarité ;
  - Un fonds social alimenté par 5% des excédents des caisses pour intervenir par des subventions dans la prise en charge de la protection sociale des populations vulnérables

Tout institut de microfinance doit travailler pour une meilleure qualité de service afin de satisfaire ses clients. Donc, des technologies doivent être proposées et exploitées afin de permettre l'informatisation des procédures de gestion du SFD.

## **II. Objectifs de l'étude**

### **1. Objectif général**

Informatiser les procédures de gestion du SFD à temps réel.

### **2. Objectifs spécifiques**

- Automatiser et sécuriser l'ouverture des comptes et les opérations bancaires (débit, crédit, transaction).
- Permettre aux clients de consulter les informations de leurs comptes.
- Centraliser l'administration des données.
- Assurer la sécurité des données.
- Disponibilité, cohérence et intégrité des données.
- Assurer un meilleur traitement des données.



## **CHAPITRE 2: PRESENTATION DU PROJET**

### **I. Spécifications des besoins fonctionnels**

L'informatisation du SFD inclut une phase de recueil des besoins fonctionnels. Ils sont déclinés comme suit :

- Gérer les employés.
- Gérer l'inscription des membres.
- Gérer l'ouverture des comptes.
- Gérer les opérations bancaires (Débit, Crédit, Virement).
- Gérer les prêts, les remboursements et les échéances.
- Offrir au client la possibilité de consulter leur compte par internet.
- Gérer les états des comptes.
- Gérer les rapports.

### **II. Délimitation du champ ou domaine de l'étude**

#### **1. Délimitation procédurale**

Les procédures suivantes sont prises en compte :

PROCEDURE D'ADHESION.

PROCEDURE D'OUVERTURE DE COMPTE (compte épargne prévoyance, compte courant salarié et compte épargne bloqué).

PROCEDURE D'OBTENTION DE PRET ET DE REMBOURSEMENT.

#### **2. Délimitation organisationnelle**

Les services concernés par l'étude sont les suivants :

- la comptabilité
- la caisse
- le comité de crédit

## **DEUXIEME PARTIE : ANALYSE ET CONCEPTION FUTURE**

### **CHAPITRE 1 : ANALYSE (ETUDE) DE LA SITUATION ACTUELLE**

Le système financier décentralisé PAMECAS est une structure de microfinance qui a plusieurs bureaux régionaux. Chaque bureau régional gère un ou plusieurs agences. Une agence est caractérisée par (Numéro, Nom, Adresse, Tel, Email, Chef Agence, Bureau Régional).

Chaque agence possède plusieurs caisses, une caisse est gérée par un caissier.

#### **PROCEDURE D'ADHESION :**

Le gestionnaire de compte effectue la demande d'adhésion.

Pour une personne :

Le membre verse un montant fixé à 10.000 F CFA, 3 photos d'identité et une photocopie de la pièce d'identité.

Le gestionnaire de compte remplit la fiche d'adhésion (Nom, Prénom, Date de naissance, Adresse, Téléphone, E-mail, Situation matrimoniale (marié ou célibataire), Nombre d'enfants, Profession, Type de rémunération (Salarié ou non salarié) Date adhésion, N° PI, la personne ayant droit en cas de décès.

Si le membre est un salarié, il doit renseigner les informations de son employeur (Employeur, Adresse, Tel, Email).

Pour les GIE ou ASSOCIATION (groupement) :

Les renseignements suivants sont fournis : Raison Sociale, le nom et le prénom et le numéro de pièce d'identification de 2 mandataires, Adresse, Tel, E-mail, numéro de récépissé.

Les 2 mandataires sont chargés de faire des opérations bancaires sur le compte qui sera ouvert par le groupement.

Une carte de membre sera éditée pour la personne qui est inscrite.

Une carte sera éditée pour chaque mandataire du groupement.

#### **PROCEDURE D'OUVERTURE DE COMPTE :**

Un employeur peut domicilier le salaire de ses employés en leur ouvrant un compte d'épargne salarié. Cependant le membre peut ouvrir d'autres comptes différents de son compte épargne salarié. Pour le membre non salarié et le groupement, le compte épargne prévoyance est le seul qu'ils peuvent ouvrir. Pour les membres évoluant dans le secteur informel, l'ouverture d'un compte d'épargne prévoyance est une condition pour accéder aux services financiers (le compte épargne bloqué et compte prêt), alors que pour les membres salariés l'ouverture d'un compte courant salarié ou d'un compte épargne prévoyance représente la condition pour accéder aux services financiers.

Pour les comptes courants salariés, un agios est retiré tous les 3 mois. Avec le compte épargne prévoyance et le compte courant salarié, on peut réaliser des opérations de débit,

de crédit et de virement entre comptes au sein du SFD. Chaque compte est géré par un gestionnaire de compte.

#### PROCEDURE D'OUVERTURE DE COMPTE EPARGNE BLOQUE:

Le membre doit ouvrir un compte épargne prévoyance et faire des opérations pour une durée supérieure à 6 mois. Un intérêt égal à 4% est versé à la fin de la durée du blocage si celle-ci est entre 6 et 12 mois et 5% au-delà. Le compte épargne prévoyance est débité par le montant bloqué plus les intérêts bancaires du compte épargne bloqué.

Un compte épargne prévoyance ou courant salarié peut être sujet à plusieurs opérations de blocage d'argent à des périodes différentes.

#### PROCEDURE D'OUVERTURE D'UN COMPTE PRET :

Le client doit fixer un rendez-vous auprès de l'agent de crédit. L'entretien consiste l'importance du projet pour lequel le client sollicite un prêt, le montant de l'apport et les possibilités de remboursement. Le dossier est mis à la disposition du comité de crédit qui accepte ou rejette la demande. En cas d'approbation, le gestionnaire de compte est avisé et le client continue la procédure avec celui-ci.

Un apport est versé pour bénéficier d'un prêt. Le montant de l'apport dépend de la fréquence de demande de prêt du client :

- si c'est le premier prêt : 25 % du montant demandé,
- si c'est le deuxième prêt : 20% du montant demandé,
- si c'est le troisième prêt : 15 % du montant demandé.

A partir du troisième prêt, le taux d'apport ne subit plus de changement.

Il existe plusieurs types de prêt (habitat, personnel et commercial). Le montant et la durée du prêt accordé dépend du type de prêt.

Le prêt habitat varie entre 3 et 15 millions et une durée de remboursement de 5 ans avec un intérêt de 30%.

Le prêt personnel varie entre 100000 FCFA et 5000000 millions et une durée de remboursement de 3 ans avec un intérêt de 15 %.

Le prêt commercial varie entre 200000 FCFA et 10000000 et une durée de remboursement de 2 ans avec un intérêt de 10 %. Le montant du prêt accordé est débité aussi bien dans le compte prêt que dans le compte épargne prévoyance.

#### PROCEDURE DE REMBOURSEMENT D'UN PRET BANCAIRE

Le remboursement est conditionné par une échéance à respecter.

Montant à rembourser= montant prêt+ (montant prêt \*intérêt) /durée

Chaque remboursement versé diminue le montant du compte prêt. Si la date de remboursement dépasse le 10 du mois courant, une somme est versée à la banque comme une pénalité de retard de paiement et déduite sur le montant de l'apport à reverser à la fin du remboursement total du prêt. Elle est calculée de la manière suivante:

$((\text{Montant prêt}-\text{somme des remboursements})*10\%/360)*(\text{nombre de jours de retard}).$

A la fin du remboursement total du prêt, le montant de l'apport est reversé au client dans son compte épargne prévoyance et le compte passe à l'état inactif.

L'administrateur paramètre tout le système (ajout des agences, des employés et des caisses).

## **Critique de la situation actuelle**

Le système actuel est manuel et présente un certain nombre d'insuffisances :

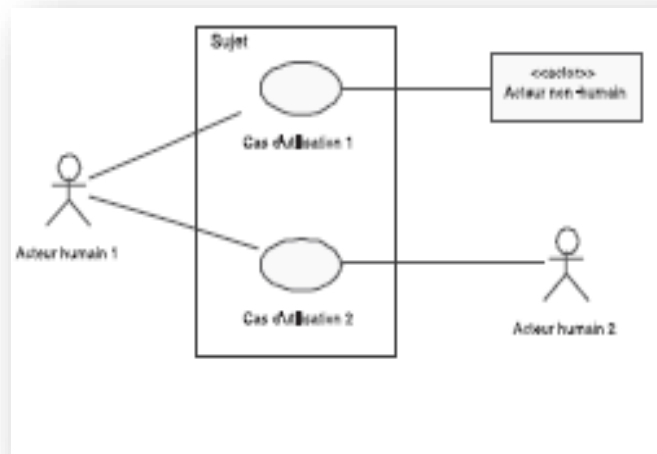
- Les procédures d'adhésion sont souvent longues
- La gestion actuelle des comptes entraîne beaucoup de contentieux
- Pertes d'informations et de documents
- Manque d'informations pour le client.



Exemple :



## 2. Exemple de diagramme de cas d'utilisation



## II. Le diagramme de séquence

### 1. Définition et formalisme

#### ❖ Définition

Le diagramme de séquence est un diagramme d'interaction. Les diagrammes d'interaction permettent d'établir un lien entre les diagrammes de cas d'utilisation et les diagrammes de classes : ils montrent comment des objets (i.e. des instances de classes) communiquent pour réaliser une certaine fonctionnalité. Ils apportent un aspect dynamique à la modélisation du système.

#### ❖ Formalisme

Les éléments du diagramme de séquence sont :

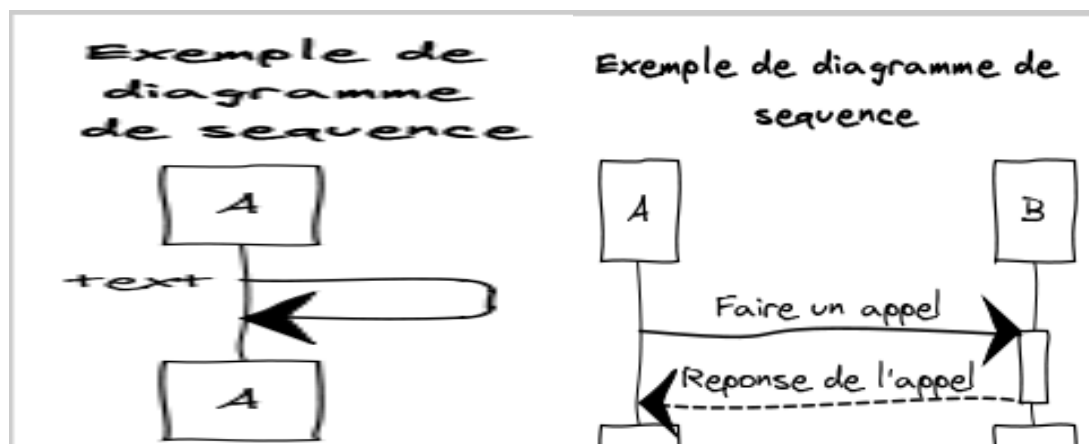
- Acteurs
- Objets (instances)
- Messages (cas d'utilisation, appels d'opération)

Principe de base : Représentation graphique de la chronologie des échanges de messages avec le système ou au sein du système.

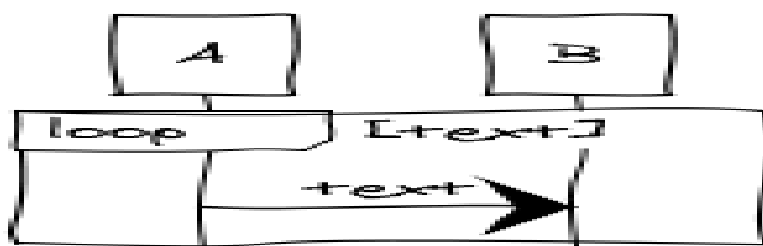
- « Vie » de chaque entité représentée verticalement
- Échanges de messages représentés horizontalement

La liste suivante regroupe les opérateurs d'interaction par fonctions :

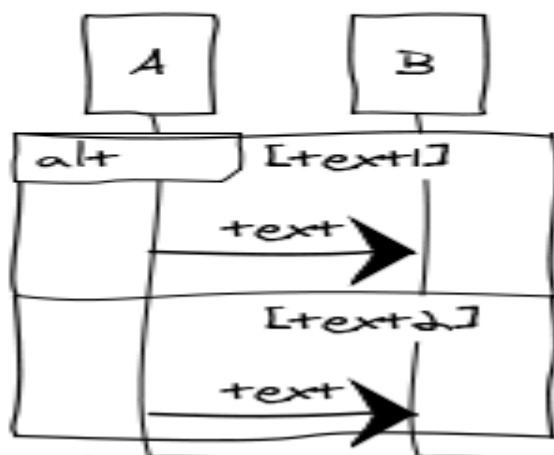
- les opérateurs de choix et de boucle : alternative, option, break et loop;
- les opérateurs contrôlant l'envoi en parallèle de messages : parallelet criticalregion;
- les opérateurs contrôlant l'envoi de messages : ignore, consider, assertion et negative;
- les opérateurs fixant l'ordre d'envoi des messages : weaksequencing,strictsequencing.



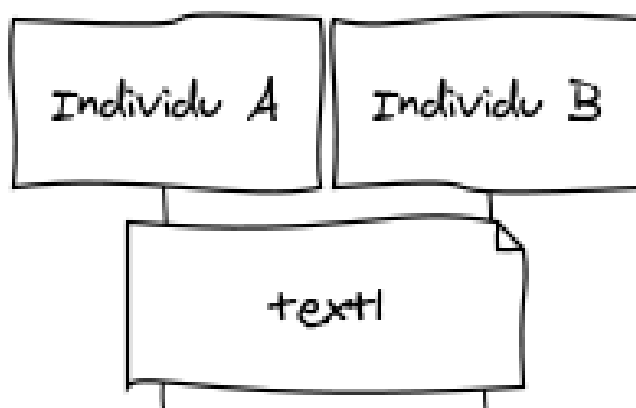
Exemple de diagramme de sequence



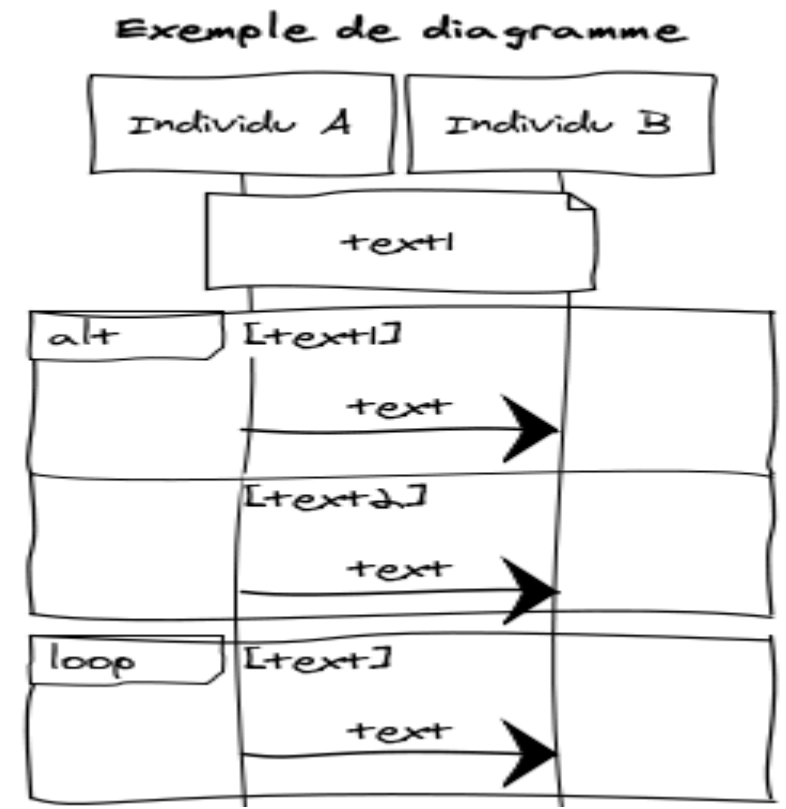
Exemple de diagramme de sequence



Exemple de diagramme de sequence



## 2. Exemple de diagramme de séquence



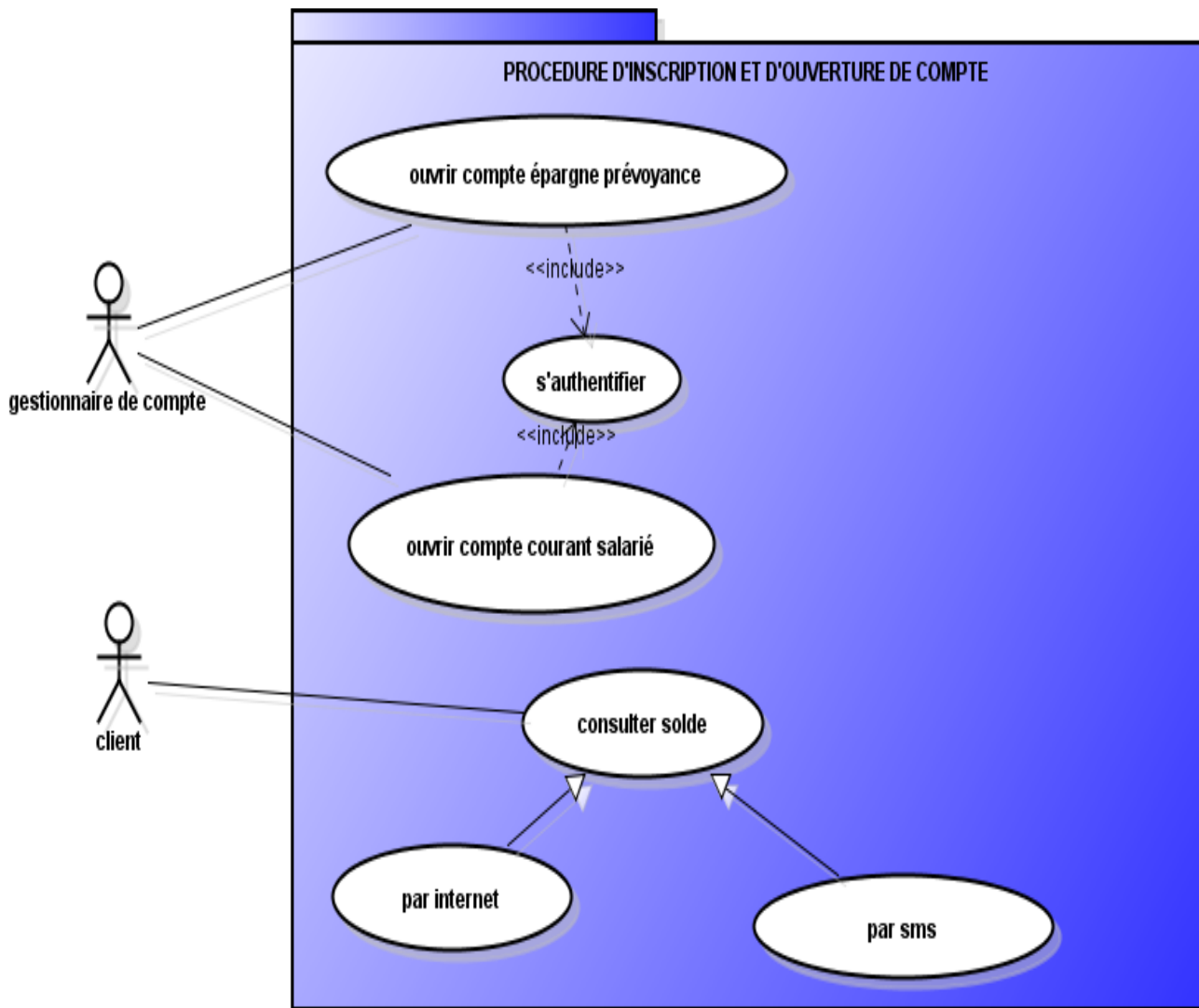


### III. Schémas des diagrammes de cas d'utilisation et de séquence

Plusieurs diagrammes de cas d'utilisation vont être établis à partir de l'analyse de la situation actuelle.

Pour chacun d'eux, une description visant à sortir les postconditions, préconditions, les scénarios normaux, les enchainements alternatifs et d'erreurs sera faite. Il sera exposé pour quelques cas d'utilisation, leur diagramme de séquence respectif.

Diagramme de cas d'utilisation : Procédure d'inscription et d'ouverture de compte



### Cas d'utilisation : Authentification

Résumé : Ce cas d'utilisation montre la procédure d'authentification d'un employé.

Acteurs : employé et le système

Préconditions :

La connexion avec le système doit être toujours opérationnelle.

Scénario nominal :

1. Le système demande à l'employé de s'authentifier.
2. L'employé saisit son login et son mot de passe.
3. Le système vérifie si le login et le mot de passe sont corrects.
4. Si les informations sont valides, la connexion au système est établie.

Enchainements alternatifs :

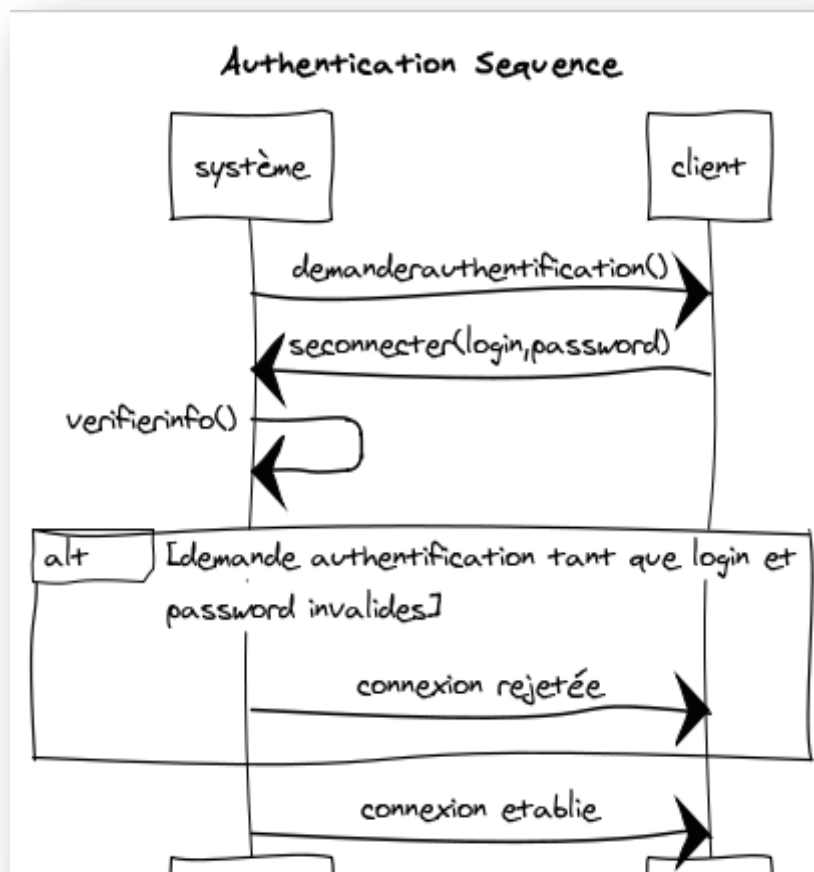
A1 : Les informations ne sont pas valides

On retourne au niveau 2 du schéma nominal

Postconditions :

Authentification réussie.

Echec d'authentification.



Cas d'utilisation : Ouvrir un compte pour un membre

Résumé : Déroulement de la procédure d'ouverture d'un compte pour un client de type membre.

Acteurs : Le gestionnaire de compte et le système

Préconditions :

La connexion avec le système doit être toujours opérationnelle.

Scénario nominal

1. Le gestionnaire de compte s'authentifie au système.
2. Le gestionnaire de compte saisit le statut du membre.
3. Le système modifie le formulaire.
4. Le gestionnaire de compte saisit les informations du compte.
5. Le gestionnaire de compte valide le formulaire.
6. Le système vérifie si les informations renseignées sont corrects.
7. Le compte épargne prévoyance est créé.
8. Le gestionnaire de compte édite de la fiche d'inscription du client.

Enchaînements alternatifs :

A1 : Les informations ne sont pas valides.

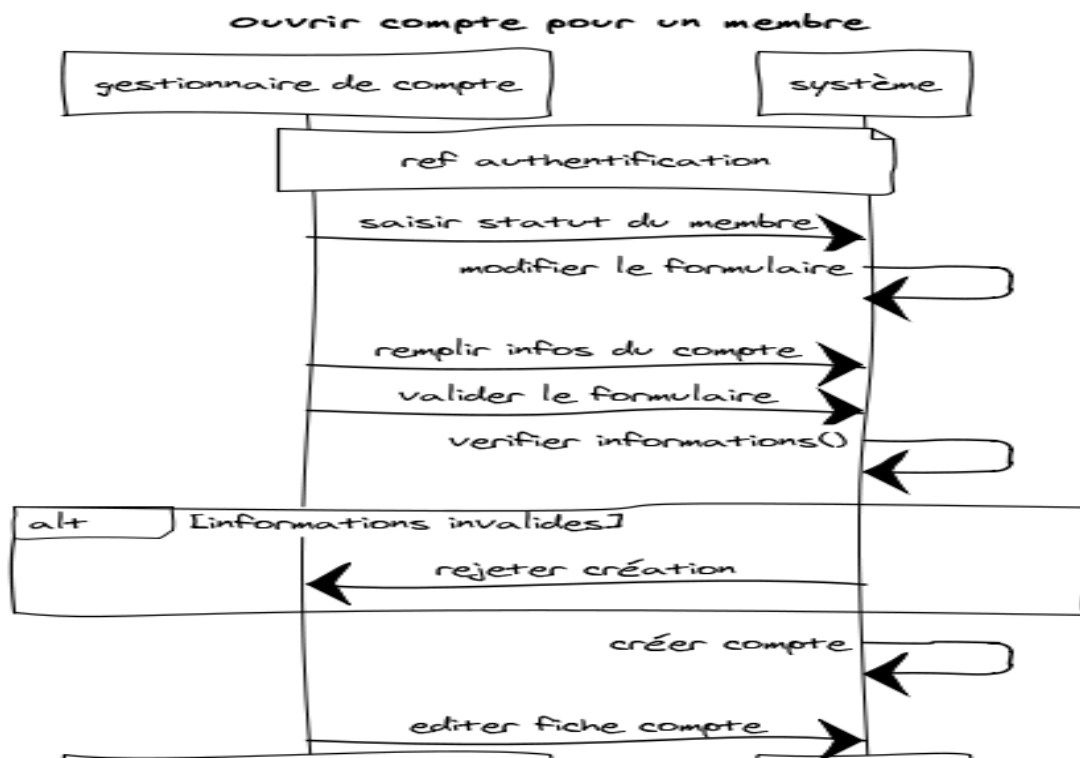
On retourne au niveau 2 du schéma nominal

Postconditions :

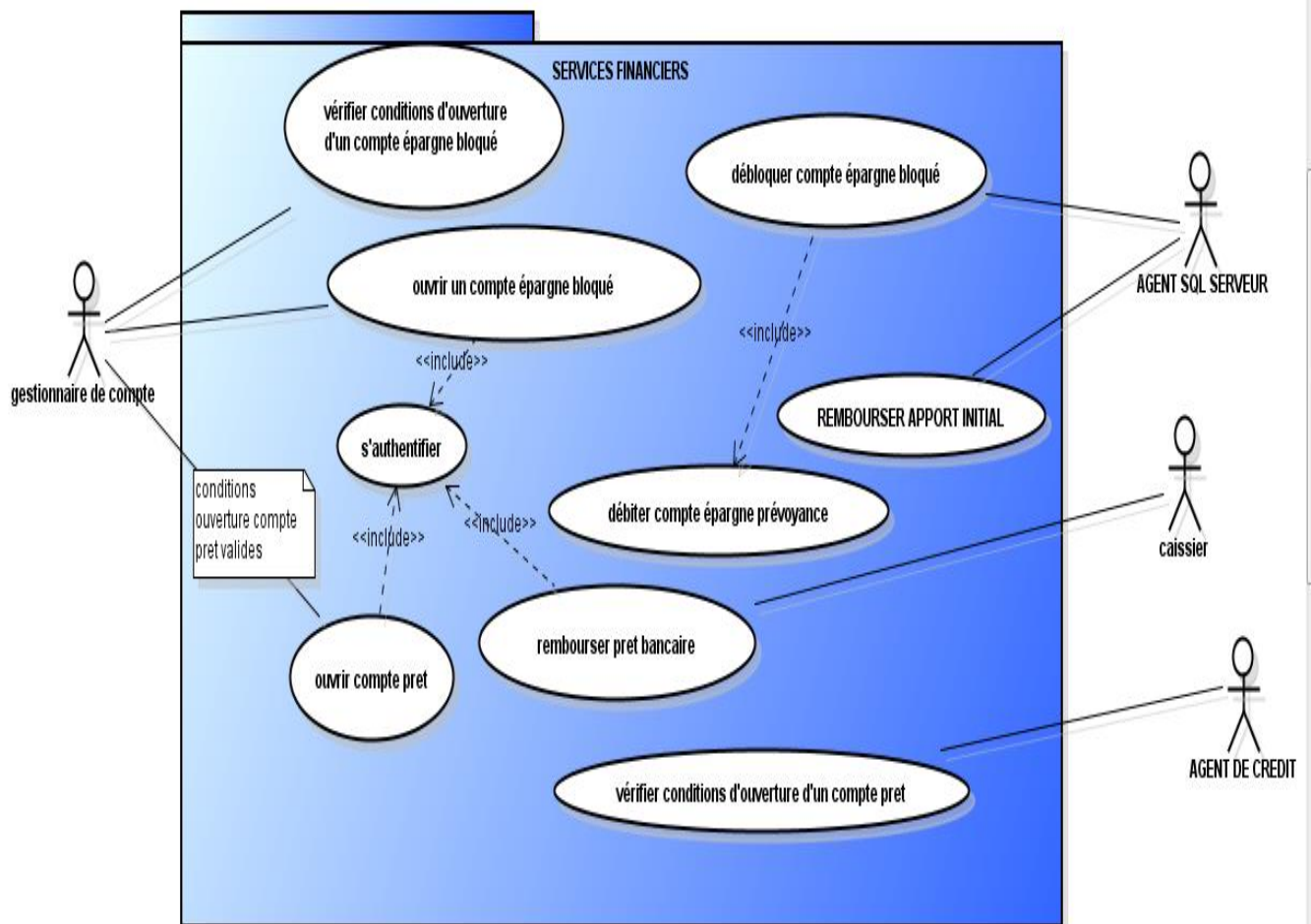
Le client est enregistré dans la base de données.

Le compte épargne prévoyance est créé.

Les informations visant à ouvrir un compte épargne prévoyance ne sont pas valides, impossible de créer le compte.



## Diagramme de cas d'utilisation : Accès aux services financiers



Cas d'utilisation : Ouvrir un compte épargne bloqué

Résumé : Ce cas d'utilisation montre la procédure d'ouverture du compte épargne bloqué

Acteurs : assistant-comptable et le système

Préconditions :

La connexion avec le système doit être opérationnelle.

Les conditions d'ouverture du compte épargne bloqué doivent être valides.

Scénario nominal :

1. Appel du cas d'utilisation s'authentifier
2. l'assistant remplit les informations du compte épargne bloqué.
3. Le système vérifie si les informations remplies sont correctes.
4. Si les informations sont correctes, le système crée le compte épargne bloqué.

Postconditions :

Le compte épargne bloqué est ouvert.

Conditions d'ouverture non valides, impossible d'ouvrir le compte épargne bloqué.

Enchainements alternatifs :

A1 : Si le montant à bloquer est supérieur au solde du compte épargne prévoyance.

Le système indique que le solde du compte épargne prévoyance est insuffisant.

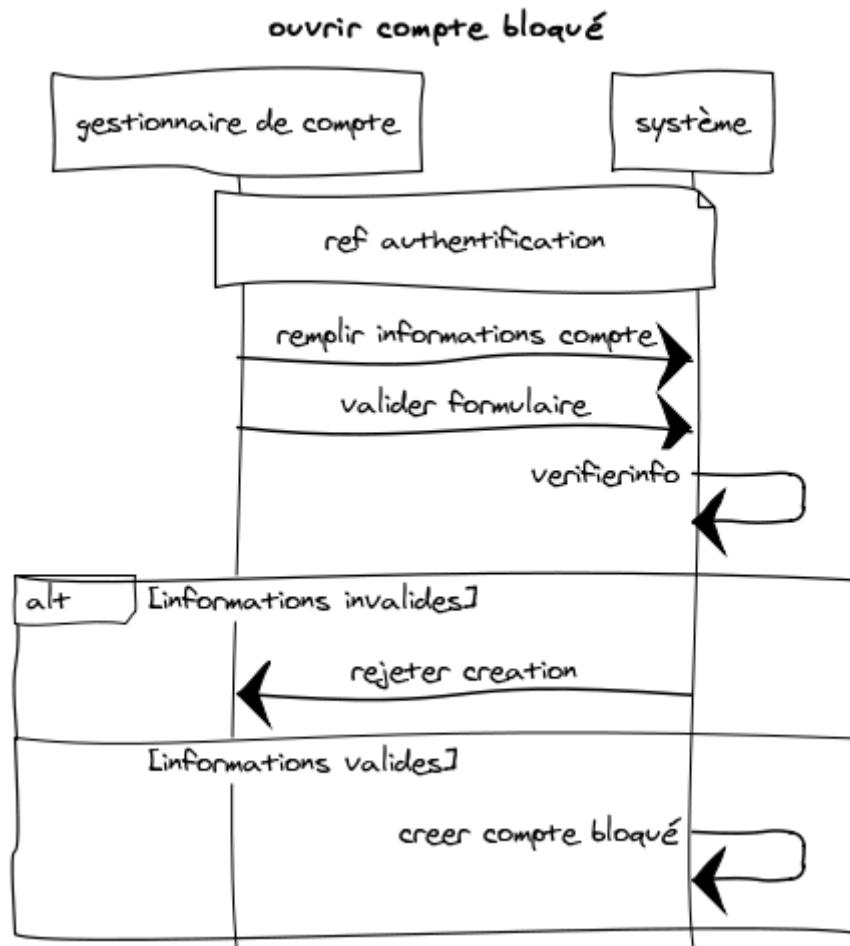
Le scénario nominal reprend au niveau 3.

Enchainements d'erreur :

E1 : Conditions d'ouverture non valides

L'enchainement E1 démarre au niveau 1 du schéma nominal

3- Le système ne peut pas permettre au client d'ouvrir un compte, échec du cas d'utilisation.



Cas d'utilisation : Vérifier conditions d'ouverture compte épargne bloqué

Résumé : Ce cas d'utilisation montre la procédure d'autorisation d'ouverture du compte épargne bloqué

Acteurs : gestionnaire de compte et le système

Préconditions :

La connexion avec le système doit être opérationnelle.

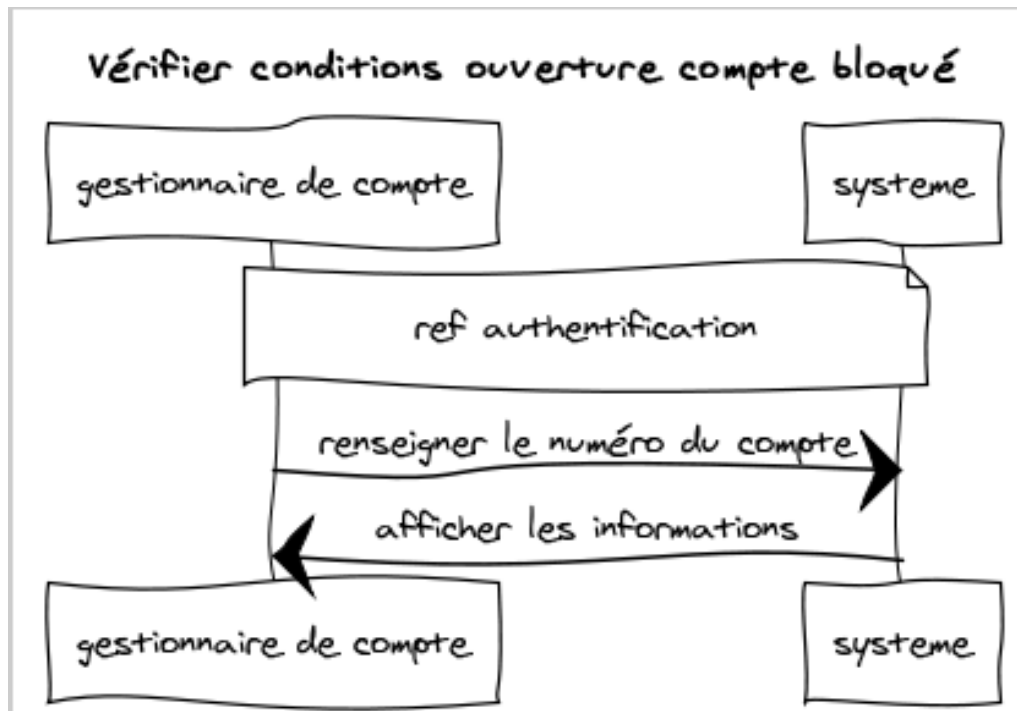
Scénario nominal :

1. Appel du cas d'utilisation s'authentifier
2. Le gestionnaire de compte renseigne le numéro du compte
3. Le système affiche les informations permettant au gestionnaire de compte de prendre une décision

Postconditions :

Conditions d'ouverture non valides, impossible d'ouvrir un compte épargne bloqué.

Conditions d'ouverture valides, ouvrir un compte épargne bloqué.



Cas d'utilisation : Débloquer compte épargne bloqué

Résumé : Ce cas d'utilisation montre le déblocage du compte épargne bloqué

Acteur non humain : Agent SQL SERVEUR

Préconditions :

Echéance de la durée du blocage.

Scénario nominal :

1. Débloquer le compte épargne bloqué.

Postconditions :

Compte épargne bloqué débloquent.

Cas d'utilisation : Débiter compte épargne prévoyance

Résumé : Ce cas d'utilisation décrit le débit du compte épargne prévoyance par la somme du montant bloqué et des intérêts bancaires

Acteurs : Agent SQL SERVEUR

Préconditions :

Compte épargne bloqué débloquent

Scénario nominal :

1. Débitierle compte épargne prévoyance.

Postconditions :

Compte épargne prévoyance débité.

Cas d'utilisation : Ouvrir un compte prêt

Résumé : Ce cas d'utilisation montre la procédure d'ouverture du compte prêt

Acteurs : assistant-comptable et le système

Préconditions :

La connexion avec le système doit être opérationnelle.

Les conditions d'ouverture du compte prêt doivent être valides

Scénario nominal :

1. Appel du cas d'utilisation s'authentifier
2. L'assistant-comptable remplit les informations du compte prêt.
3. Le système vérifie si les informations du compte prêt sont correctes.
4. Le système débite le compte épargne prévoyance.
5. Le compte prêt est créé.
6. Editer fiche création du compte

Postconditions :

Le compte prêt est créé.

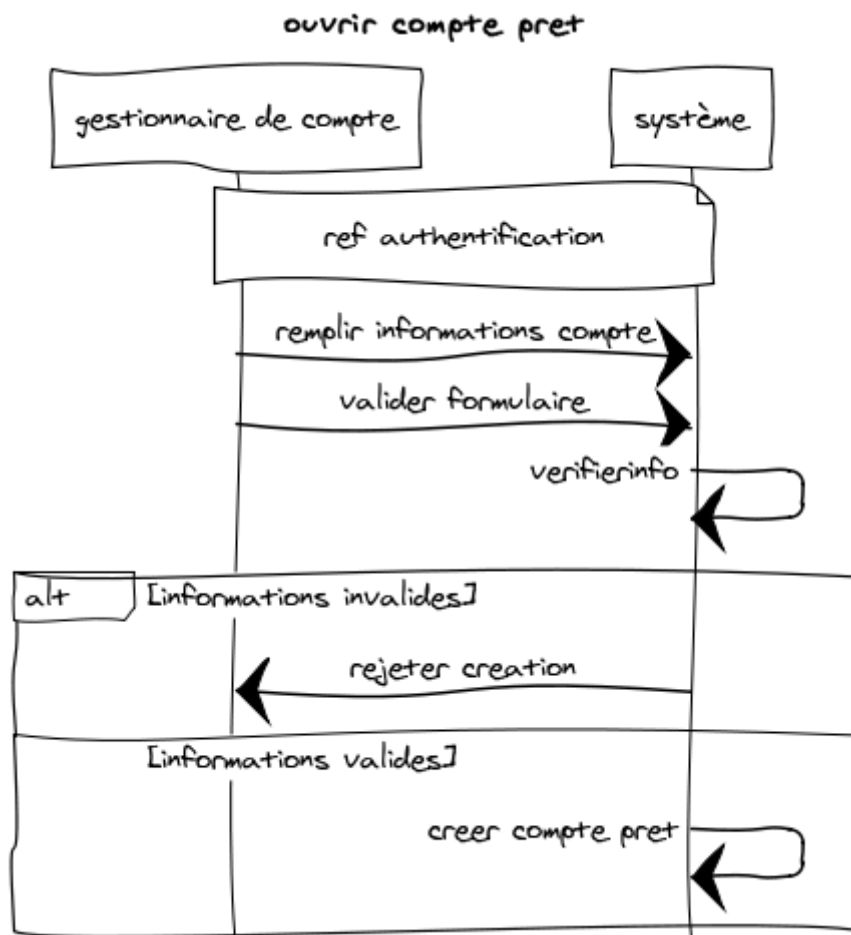
Informations non valides, impossible d'ouvrir le compte prêt.

Enchainements d'erreur :

E1 : informations du compte prêt non valides

L'enchainement E1 démarre au niveau 1

3- Le système ne peut pas permettre au client d'ouvrir un compte, échec du cas d'utilisation.





Cas d'utilisation : Vérifier conditions d'ouverture d'un compte prêt

Résumé : Montrer la procédure de vérification des conditions d'ouverture du compte prêt

Acteurs : agent de crédit et le système

Préconditions :

La connexion avec le système doit être opérationnelle

Scénario nominal :

1. L'agent de crédit entre le numéro d'identification du membre
2. Le système vérifie si le membre a un compte épargne prévoyance et s'il n'a pas de compte prêt en cours de validité.
3. Le système renvoie le résultat de la vérification
4. L'agent de crédit regarde si les conditions sont valides à partir du résultat renvoyé par le système.

Postconditions :

Conditions d'ouverture non valides, impossibilité d'ouverture d'un compte prêt.

Conditions d'ouverture valides, possibilité d'ouverture d'un compte prêt

Cas d'utilisation : Rembourser un prêt bancaire

Résumé : Ce cas d'utilisation montre la procédure de remboursement d'un prêt bancaire

Acteurs : le caissier et le système

Préconditions :

La connexion avec le système doit être opérationnelle

Scénario nominal :

1. Le caissier remplit les informations du compte remboursement.
2. Si les informations sont valides et que la date de remboursement n'est pas dépassée le système les enregistre.
3. Le compte prêt est crédité du montant du remboursement mensuel.

Enchainements alternatifs :

A1 : Si le montant à rembourser est inférieur au montant de remboursement mensuel.

Le système indique que le solde du compte épargne prévoyance est insuffisant.

Le scénario nominal reprend au niveau 1.

A2 : Si la date de remboursement est dépassée, le système applique des pénalités

Des pénalités sont calculées et débitées par le montant de l'apport à rembourser à la fin du remboursement total du prêt.

Postconditions :

Remboursement effectué et le compte prêt crédité

Cas d'utilisation : Rembourser apport initial

Résumé : Ce cas d'utilisation montre le remboursement de l'apport initial

Acteur : Agent SQL-SERVEUR

Préconditions :

Le solde du compte prêt doit être nul.

Scénario nominal :

1. Déduire le montant des pénalités du montant de l'apport initial.
2. Débitier le compte épargne prévoyance par le montant de l'apport.

Postconditions :

Le compte épargne prévoyance est débité par le montant de l'apport initial.

## IV. Diagramme de classe

### 1. Définition et formalisme

#### ❖ Définition

Le diagramme de classes est le point central dans un développement orienté objet.

En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs.

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Les cas d'utilisation ne réalisent donc pas une partition des classes du diagramme de classes. Un diagramme de classes n'est donc pas adapté (sauf cas particulier) pour détailler, décomposer, ou illustrer la réalisation d'un cas d'utilisation particulier.

#### ❖ Formalisme

##### Classe et objet

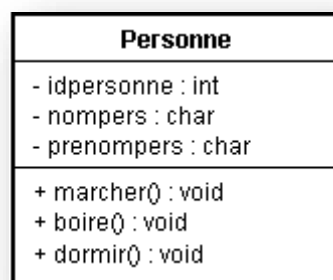
Une classe représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques. On peut parler également de type.

Un attribut représente un type d'information contenu dans une classe.

Exemple : identifiant, nom, prénom.

Les opérations ou méthodes décrivent les éléments individuels d'un comportement quel'on peut invoquer. Ce sont des fonctions qui peuvent prendre des valeurs en entrée et modifier les attributs ou produire des résultats.

Exemple : la classe Personne.



Un objet est une entité aux frontières bien définies, possédant une identité et encapsulant un état et un comportement. Un objet est une instance (ou occurrence) d'une classe.

Exemple : Pascal Roques est un objet instance de la classe Personne.

## Association

Une association représente une relation sémantique durable entre deux classes.

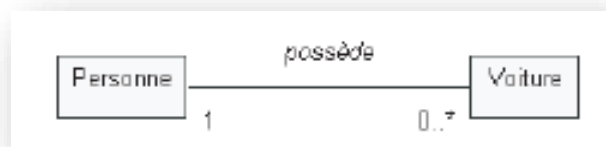
Les 2 extrémités d'une association possèdent une indication de multiplicité. Elle spécifie sous la forme d'un intervalle d'entiers positifs ou nuls le nombre d'objets qui peuvent participer à une relation avec un objet de l'autre classe dans le cadre d'une association.

### Multiplicité ou cardinalité

La multiplicité associée à une terminaison d'association, d'agrégation ou de composition déclare le nombre d'objets susceptibles d'occuper la position définie par la terminaison d'association. Voici quelques exemples de multiplicité :

- exactement un : 1 ou 1..1
- plusieurs : \* ou 0..\*
- au moins un : 1..\*
- de un à six : 1..6

Exemple : une personne peut posséder plusieurs voitures (entre 0 et un nombre quelconque) ; une voiture est possédée par une seule personne.



## Agrégation et composition

Une agrégation est une association qui représente une relation d'inclusion structurelle ou comportementale d'un élément dans un ensemble. Graphiquement, on ajoute un losange vide du côté de l'agregat. Contrairement à une association simple, l'agrégation est transitive.



Une composition est une agrégation plus forte impliquant que :

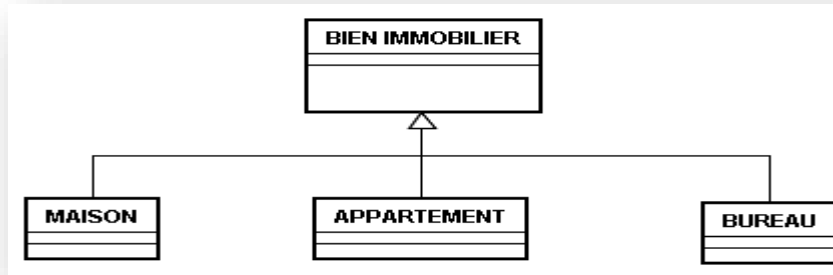
- un élément ne peut appartenir qu'à un seul agrégat composite (agrégation non partagée)
- la destruction de l'agregat composite entraîne la destruction de tous ses éléments (le composite est responsable du cycle de vie des parties).



## Généralisation, super-classe, sous-classe

Une super-classe est une classe plus générale reliée à une ou plusieurs autres classes plus spécialisées appelées sous-classes par une relation de généralisation. Les sous-classes héritent des propriétés de leur super-classe et peuvent comporter des propriétés spécifiques supplémentaires.

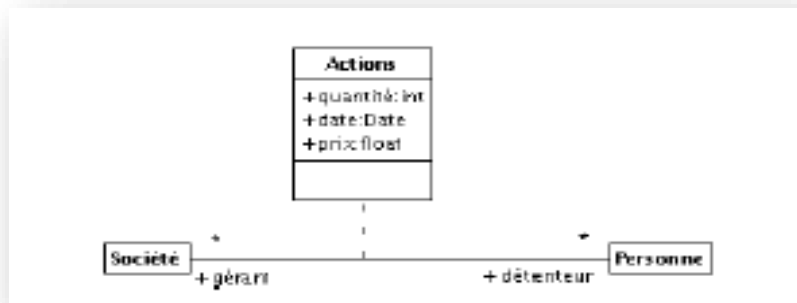
Exemple :



## Classe-association

Une classe-association possède les propriétés des associations et des classes : elle se connecte à deux ou plusieurs classes et possède également des attributs et des opérations.

Exemple :



## DEPENDANCE

Une dépendance est une relation unidirectionnelle exprimant une dépendance sémantique entre les éléments du modèle. Elle est représentée par un trait discontinu orienté. Elle indique que la modification de la cible implique une modification de la source. La dépendance est souvent stéréotypée pour mieux expliciter le lien sémantique entre les éléments du modèle.

Exemple :



## Package

C'est un mécanisme général de regroupement d'éléments en UML, qui est principalement utilisé en analyse et conception objet pour regrouper des classes et des associations.

Les packages sont des espaces de noms.

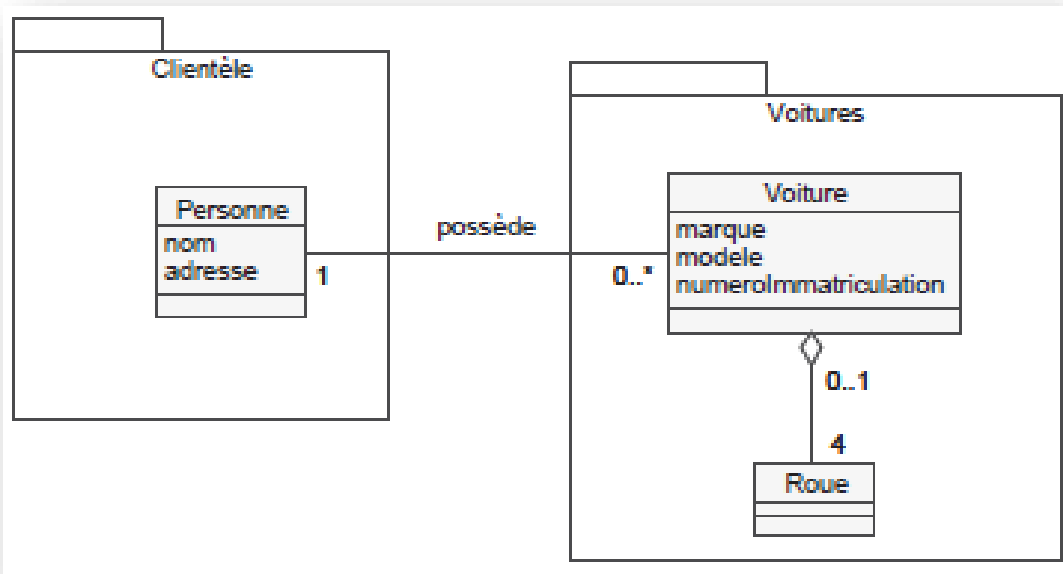
Deux éléments ne peuvent pas porter le même nom au sein du même package.

Par contre, deux éléments appartenant à des packages différents peuvent porter le même nom.

La structuration d'un modèle en packages est une activité délicate. Elle doit s'appuyer sur deux principes fondamentaux : *cohérence* et *indépendance*.

Le premier principe consiste à regrouper les classes qui sont proches d'un point de vue sémantique. Un critère intéressant consiste à évaluer les durées de vie des instances de concept et à rechercher l'homogénéité. Le deuxième principe s'efforce de minimiser les relations entre packages, c'est-à-dire plus concrètement les relations entre classes de packages différents.

### Exemple :

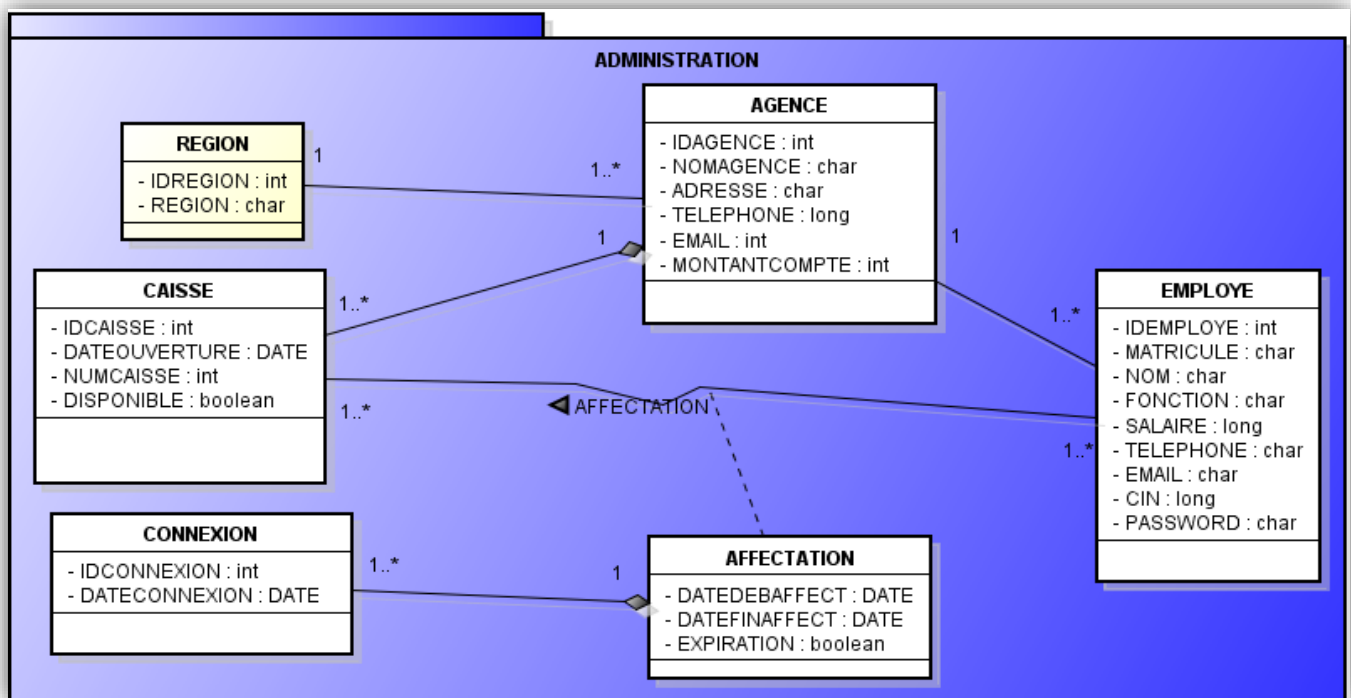


## 2. Schéma du diagramme de classe

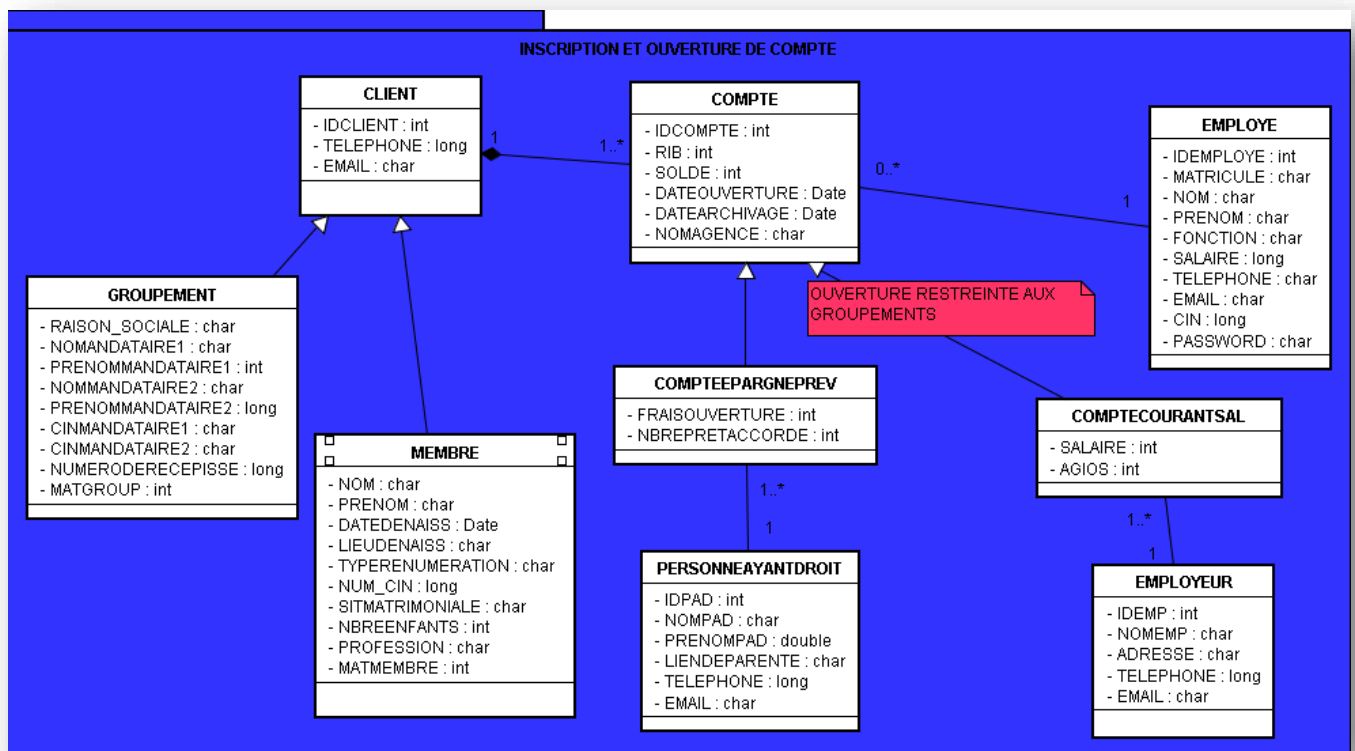
### ELABORATION DES REGLES DE GESTION

- Les régions sont les suivants : Dakar, Thiès, Louga, Saint-Louis, Ziguinchor, Kolda, Kédougou, Fatick, Kaolack, Matam, Tambacounda, Louga, Kaffrine.
- Un caissier ne travaille que dans une et une seule agence.
- Une personne ne peut s'inscrire qu'une et une seule fois.
- Un groupement ne peut s'inscrire qu'une et une seule fois.
- Une personne peut être mandataire de plusieurs groupements.
- Un membre salarié peut ouvrir plusieurs comptes de type salarié et compte de type épargne courant.
- Le membre non salarié et le groupement peuvent ouvrir plusieurs comptes épargne prévoyance.
- Les comptes sont sujets à plusieurs opérations de débit ou de crédit et de virement bancaire.
- On doit déterminer le caissier qui a fait la transaction ou le virement bancaire.
- Chaque compte est géré par un gestionnaire de compte.
- Le membre ayant un compte épargne prévoyance ne peut faire qu'une opération de blocage d'argent au plus pendant une période donnée.
- Le membre salarié ou non salarié ne peut solliciter qu'un seul prêt bancaire pendant une période donnée.
- Un compte prêt est associé à un ou plusieurs comptesremboursements.

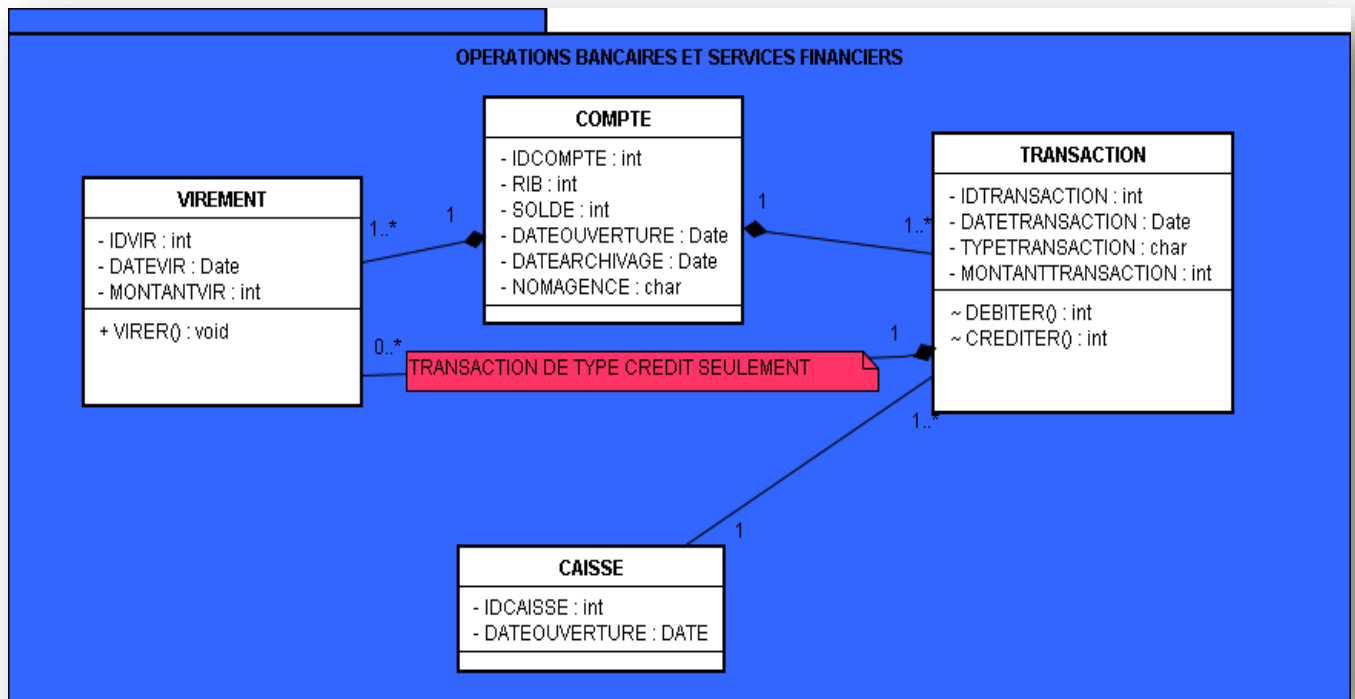
### Administration



## Inscription et ouverture de compte

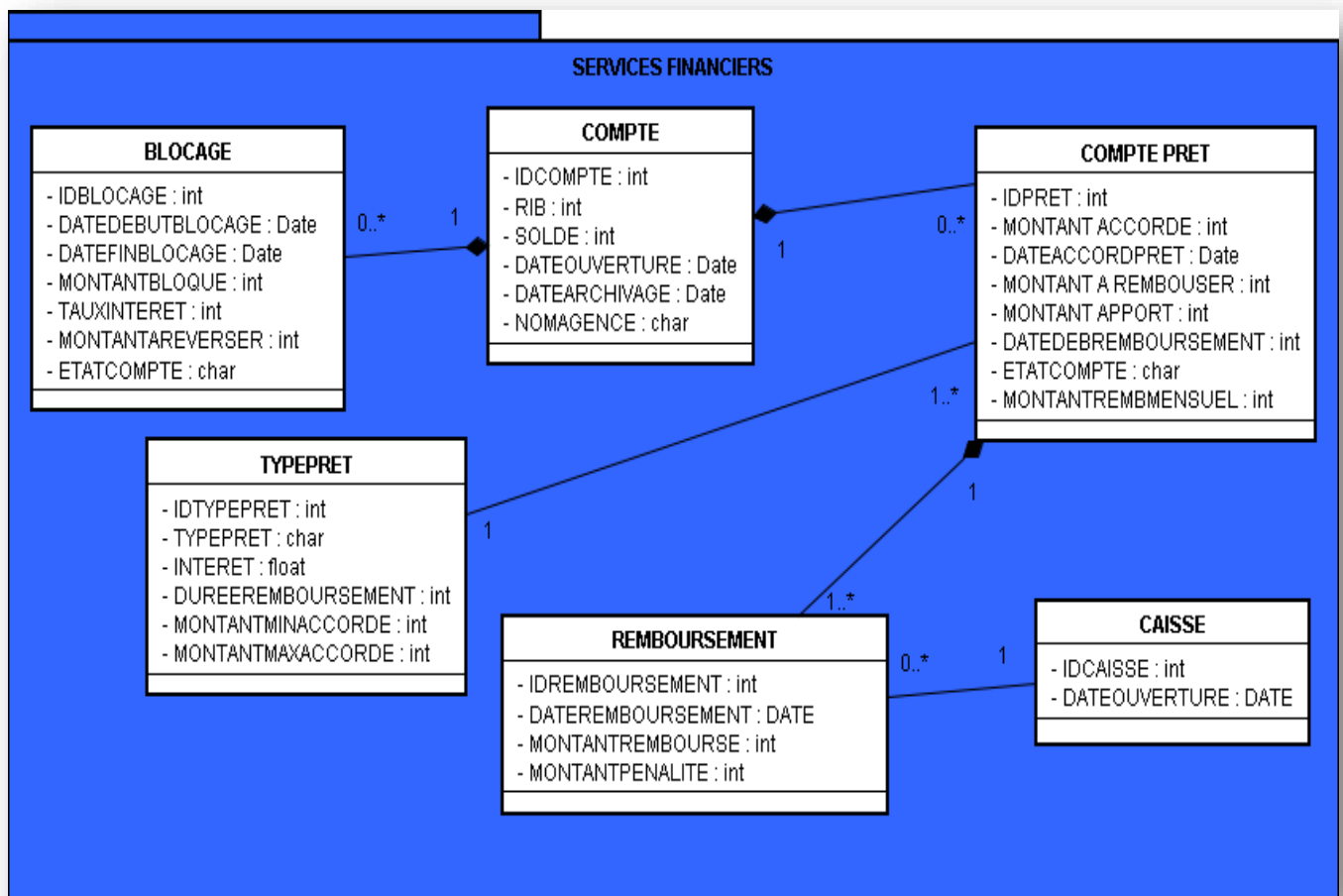


## Opérations bancaires



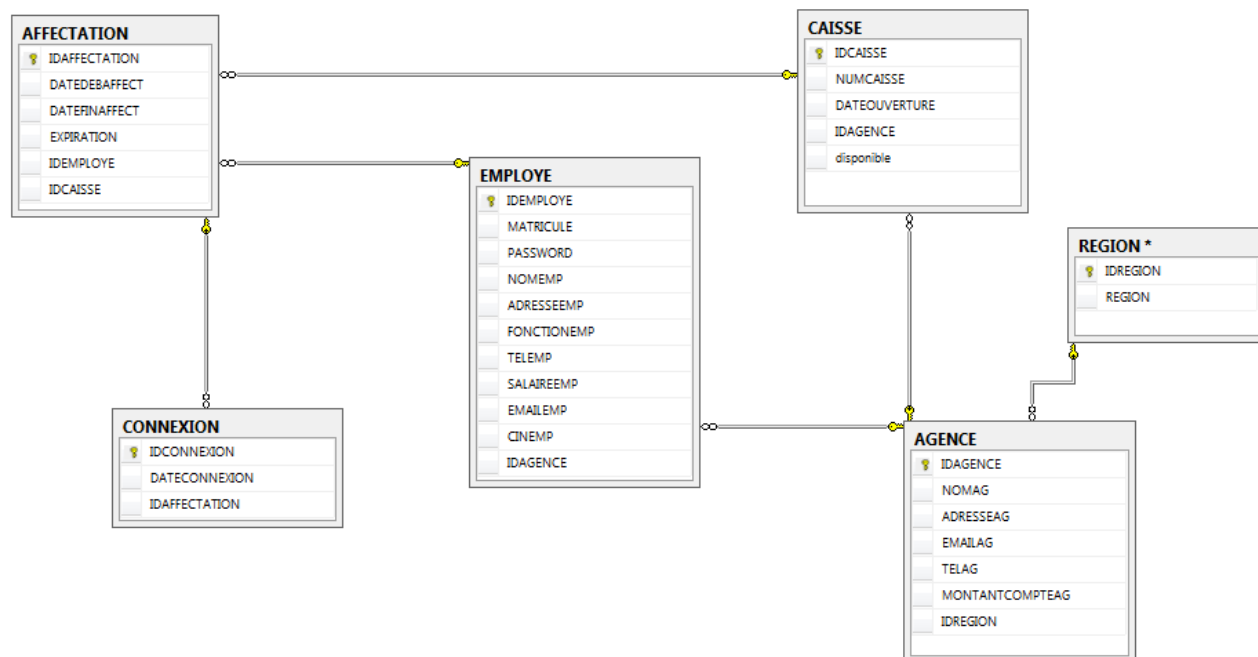


## Services financiers

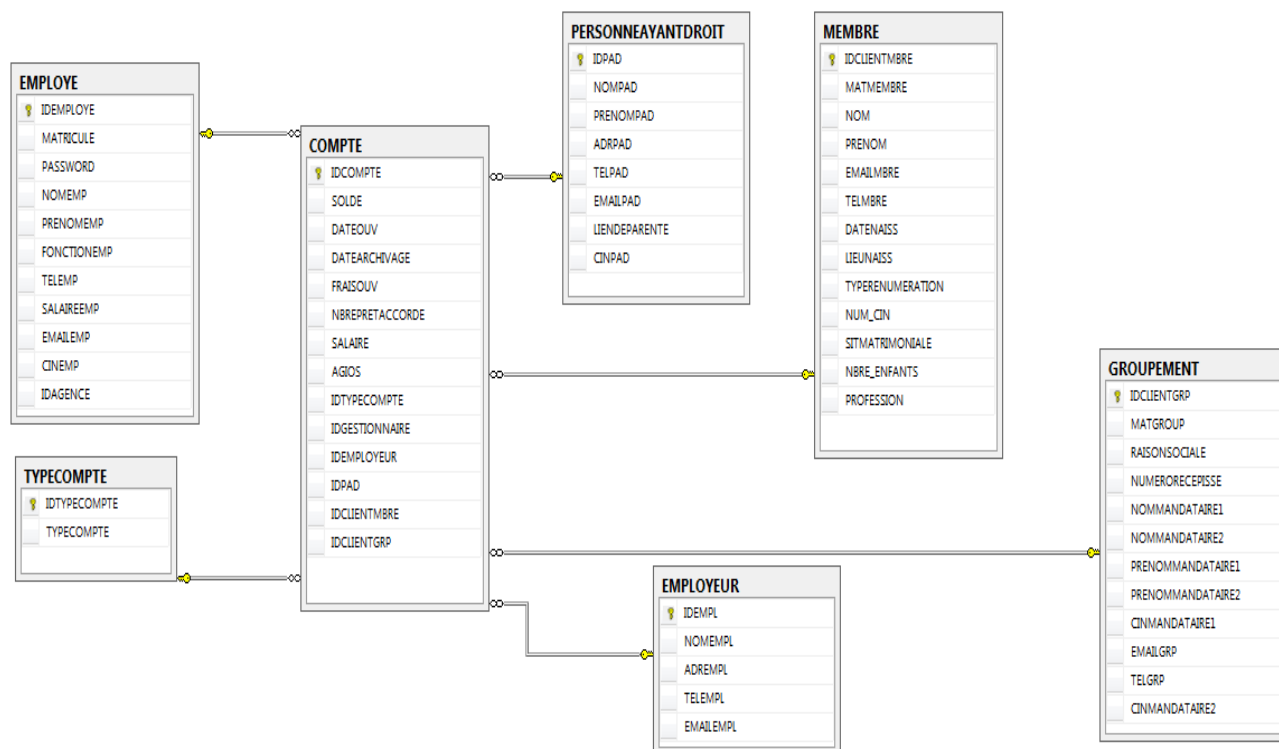


### 3. Schéma relationnel de la base de données

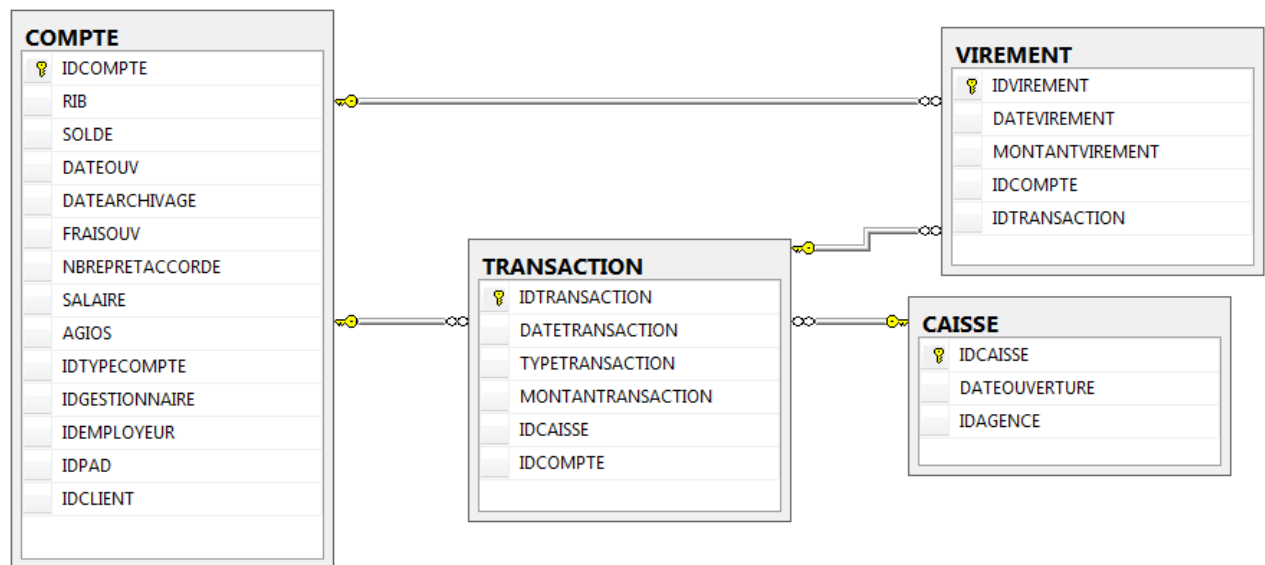
#### Administration



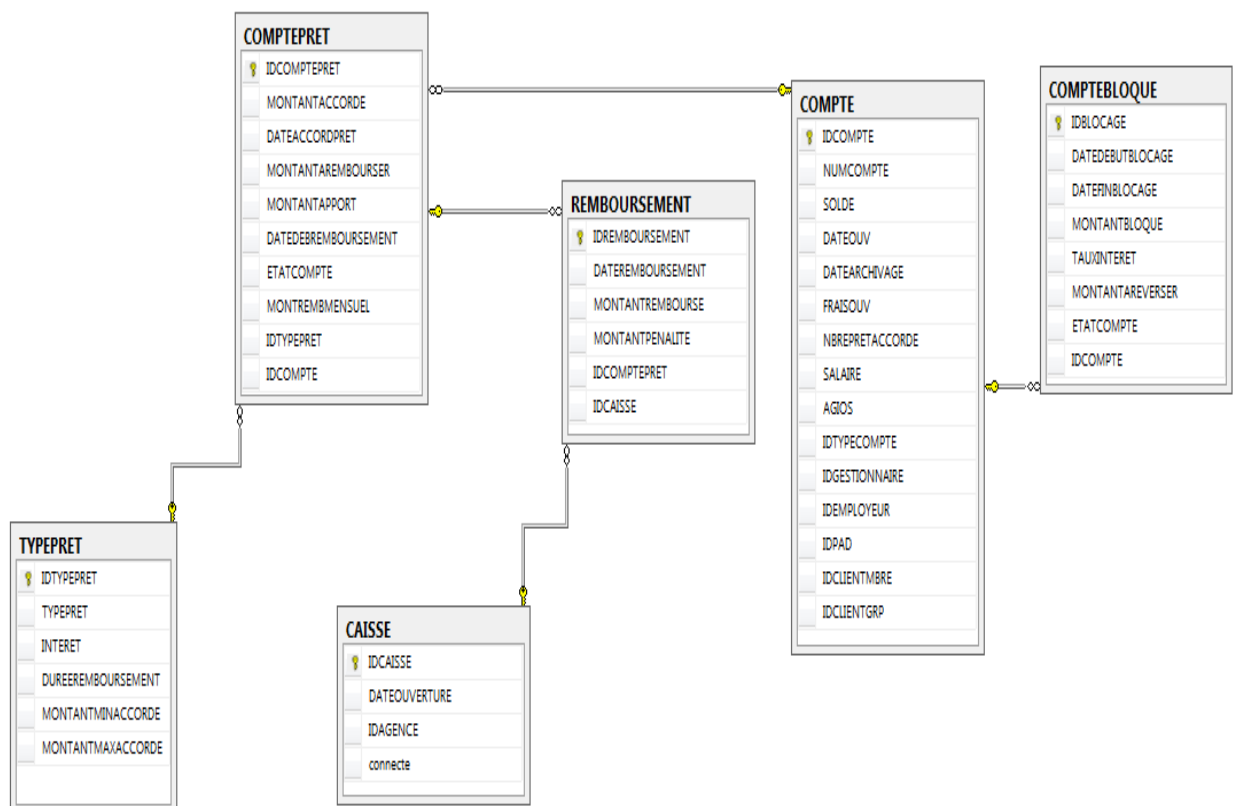
#### Inscription et ouverture de compte



## Opérations bancaires



## Services financiers

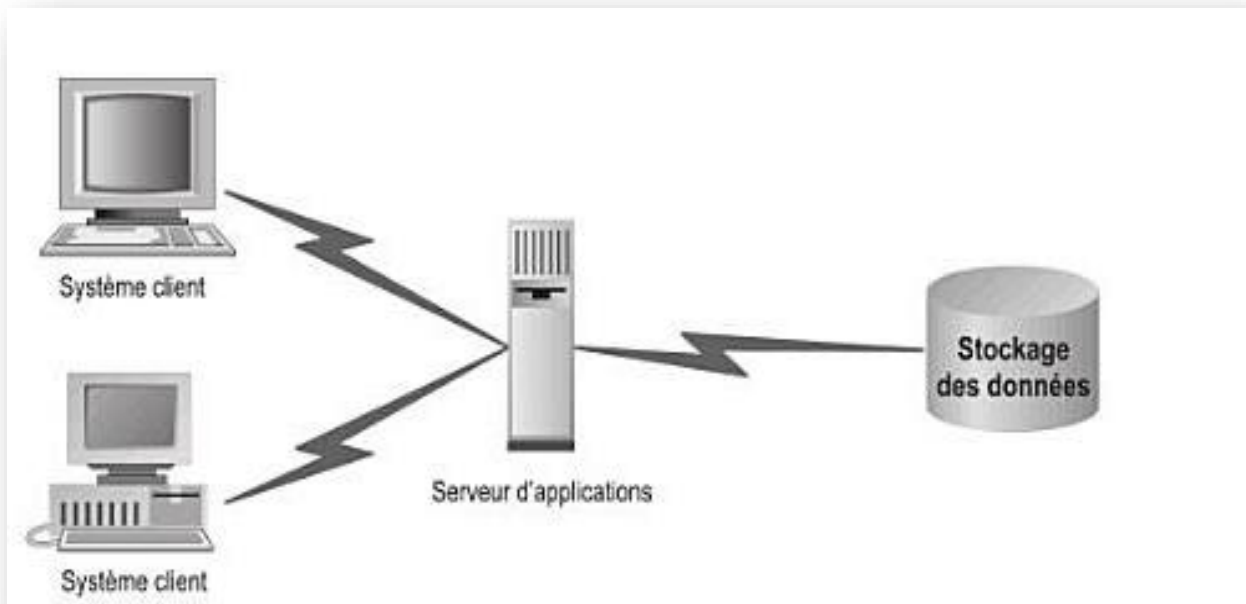


## **CHAPITRE 3 : SOLUTIONS TECHNIQUES**

### **I. Présentation de l'architecture 3-tiers**

Une architecture *3-tier* et plus généralement *n-tier* est un système composé de *n* couches. Dans le cas du Web, on parle souvent d'architecture *3-tier* ou *3 étages*. En fait, on distingue :

- d'une part, le navigateur de l'internaute (Firefox, IE, Safari, ...) qui a la charge d'afficher correctement les données transmises par le serveur,
- la couche serveur avec PHP par exemple (notez que ça pourrait très bien être un autre langage comme Python) qui traite les données à la fois venant du navigateur mais aussi celles issues de la base de données (c'est en quelque sorte le messager qui assure du transfert des données),
- le niveau qui permet le stockage et la conservation des informations : c'est la base de données (exemple : MySQL, PostGreSQL, Oracle...).



### **Schéma de l'architecture 3-tiers**

#### **1. Présentation de l'atelier de développement**

L'atelier de développement est composé des plateformes suivantes :

- Le système de gestion de bases de données SQL SERVEUR
- L'outil de développement intégré Netbeans
- Modélisation UML avec astahcommunity et websequenceuml.

## **2. Présentation de la base de données: SQL-SERVEUR**

### **❖ Historique de SQL-SERVEUR**

Microsoft SQL Server est un système de gestion de base de données (abrégé en SGBD ou SGBDR pour « Système de gestion de base de données relationnelles ») développé et commercialisé par la société Microsoft.

Bien qu'il ait été initialement développé par Sybase et Microsoft, Ashton-Tate a également été associé à sa première version, sortie en 1989. Cette version est sortie sur les plates-formes Unix et OS/2. Depuis, Microsoft a porté ce système de base de données sous Windows et il est désormais uniquement pris en charge par ce système.

En 1994, le partenariat entre les deux sociétés ayant été rompu, Microsoft a sorti la version 6.0 puis 6.5 seul, sur la plate-forme Windows NT.

Microsoft a continué de commercialiser le moteur de base de données sous le nom de SQL Server.

Tandis que Sybase, pour éviter toute confusion, a renommé Sybase SQL Server en Sybase Adaptive Server Enterprise.

Microsoft SQL Server fait désormais partie de la stratégie technique de Microsoft en matière de base de données. Le moteur MSDE, qui est la base de SQL Server, doit à terme remplacer le moteur Jet (celui qui gère les bases Access) dans les applications telles qu'Exchange et Active Directory.

La version 2005 de SQL Server est sortie le 3 novembre 2005 en même temps que Visual Studio 2005. La prise en charge de Windows Vista et de Windows Server 2008 n'a été ajoutée qu'à partir du Service Pack 2 (SP2). Actuellement le Service Pack 3 est disponible.

La version 2008 de SQL Server (nom de code Katmai) est disponible depuis août 2008. Elle est actuellement au niveau de service pack 3. Elle est disponible en 9 langues, dont le français.

La version 2012 de SQL Server est disponible depuis avril 2012.

### **❖ Les composants de SQL-SERVEUR**

Les composants de SQL-SERVEUR sont:

Les composants de type serveur :

- Moteur de base de données SQL Server (MS SQL SERVEUR)
- Analysis Services
- Reporting Services
- Integration Services
- Agent SQL Serveur
- Master Data Services

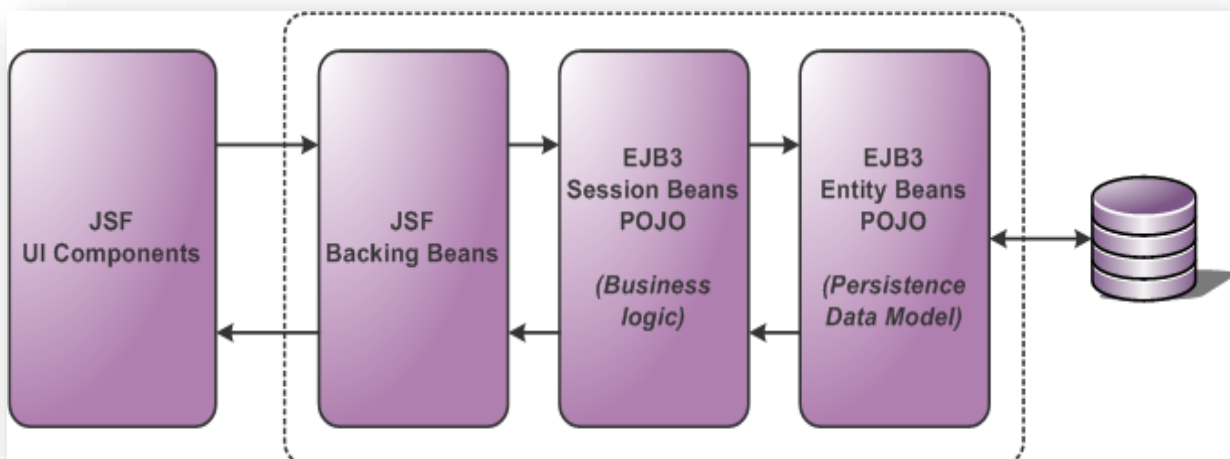
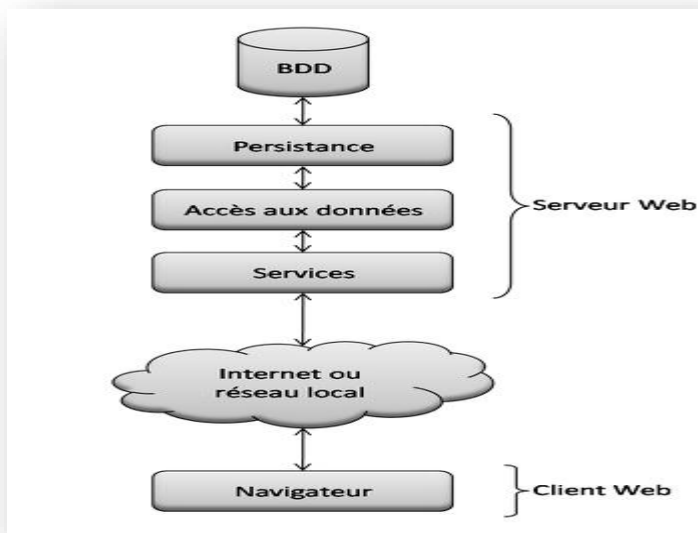
Les outils d'administration :

- SQL Server Management Studio
- Gestionnaire de configuration SQL Server

- SQL Server Profiler
- Assistant Paramétrage du Moteur de base de données
- Client de la qualité des données
- Outils de données SQL Server
- Composants de connectivité
- Business Intelligence Development Studio

### 3. Présentation de JEE

#### ❖ Architecture de JEE utilisée



La couche présentation est constituée d'un ensemble de composants JSF. L'objectif de JSF est de faciliter le développement des interfaces utilisateurs des applications web, or les deux de composants standards de JSF (html et core) s'avèrent limités et insuffisants pour le développement d'applications d'entreprise. Des jeux de composants additionnels

qui offrent de nouveaux composants plus riches sont indispensables pour le développement en JSF, Primefaces en offre un qui a prouvé son efficacité.

Primefaces est un jar facile à ajouter dans votre projet jsf pour obtenir un jeu de composants additionnel beaucoup avancé qui prend en compte les nouveautés des technologies du web surtout Ajax, les différents composants de primefaces sont à découvrir en détail sur la page officielle de demo :

<http://www.primefaces.org/showcase-labs/ui/home.jsf>.

Les EJB session jouent un double rôle puisqu'ils permettent l'échange des données entre la couche présentation et la couche métier, la validation des données, et la gestion des transactions et des sessions des utilisateurs. Les EJB entités représentent le modèle des données. Ceux-ci seront directement persistés dans la base de données grâce à l'api de persistance utilisée.

### ❖ Outil de développement : Netbeans

NetBeans est un environnement de développement intégré (IDE) placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution Licence). En plus de java, NetBeans permet également de supporter différents autres langages comme python, C, C++, XML, HTML, et PHP. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projet multi-langage, refactoring, éditeur d'interfaces et de pages web). NetBeans est disponible sous Windows, Linux, Solaris, Mac OS et Open VMS.

## II. Le diagramme de déploiement

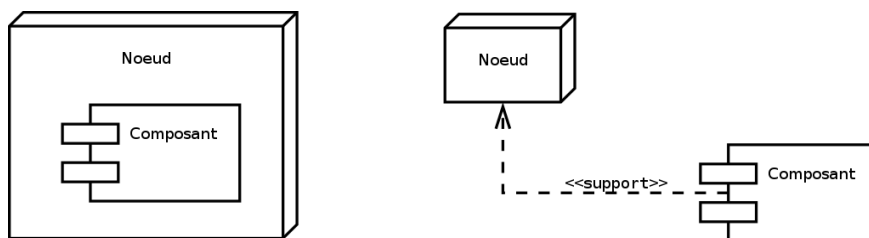
### 1. Définition et formalisme

#### ❖ Définition

Les diagrammes de composants et les diagrammes de déploiement sont les deux derniers types de vues statiques en UML. Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds. Les diagrammes de déploiement existent sous deux formes : spécification et instance.

#### ❖ Formalisme

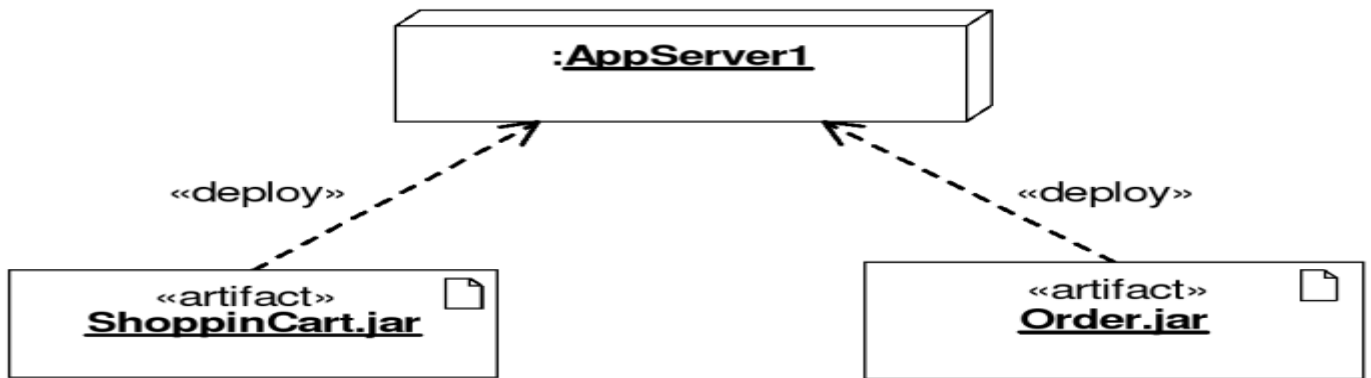
Chaque ressource est matérialisée par un nœud représenté par un cube comportant un nom. Un nœud est un classeur et peut posséder des attributs (quantité de mémoire, vitesse du processeur...). Pour montrer qu'un composant est affecté à un nœud, il faut soit placer le composant dans le nœud, soit les relier par une relation de dépendance stéréotypée « *support* » orientée du composant vers le nœud.



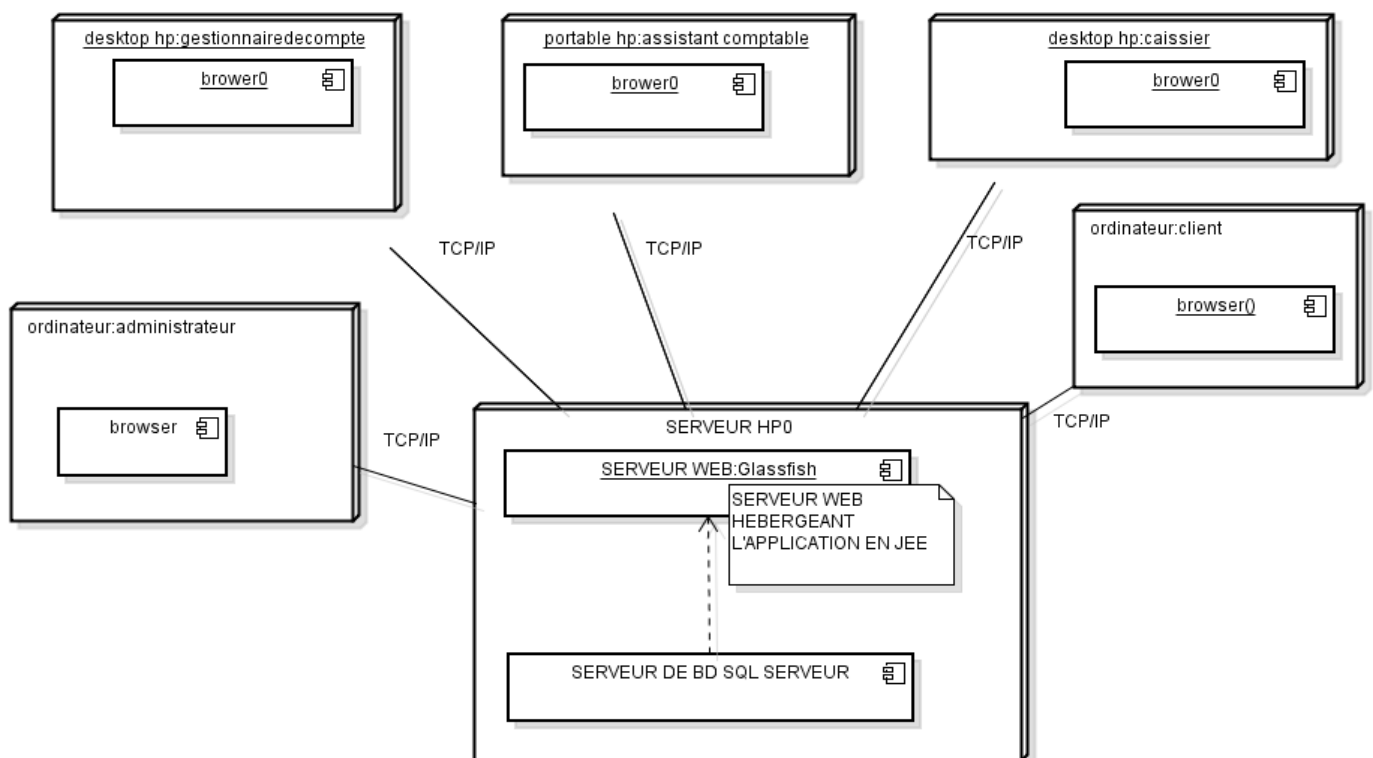
Un artefact correspond à un élément concret existant dans le monde réel (document, exécutable, fichier, tables de bases de données, script...). Il se représente comme un classeur par un rectangle contenant le mot-clef « *artifact* » suivi du nom de l'artefact. Une instance d'un artefact se déploie sur une instance de nœud. Graphiquement, on utilise une relation de dépendance (flèche en trait pointillé) stéréotypée « *deploy* » pointant vers le nœud en question. L'artefact peut aussi être inclus directement dans le cube représentant le nœud.



## Exemple :



## 2. Schéma du diagramme de déploiement



## **TROISIEME PARTIE : IMPLEMENTATION DE LA SOLUTION**

### **CHAPITRE1 : IMPLEMENTATION DE LA BASE DE DONNEES**

Nous avons utilisé l'outil d'administration SQL-Serveur Management Studio pour implémenter la base de données.

SQL Serveur Management Studio dispose d'une interface qui permet de créer des serveurs, des bases de données et de gestion facile des ressources à utiliser. En l'utilisant, vous pouvez gérer des serveurs locaux ou distants en établissant une connexion à une instance de SQL SERVEUR et administrer ainsi ces ressources.

Si vous avez désactivé les connexions à un serveur distant à un serveur particulier, vous pouvez travailler seulement avec un serveur local.

Les vues de SQL Serveur Management Studio sont les suivantes :

- ✓ Object Explorer

Permet de voir et de se connecter aux instances de SQL-SERVEUR, Analysis Services, Integration Services, and Reporting Services. Une fois que vous vous êtes connectés à un serveur particulier, il est possible de voir ses composants en arborescence.

- ✓ Registered Servers

Affiche les serveurs actuellement enregistrés. Utiliser les serveurs enregistrés pour enregistrer les informations de connexion aux serveurs sur lesquels vous accédez fréquemment. La barre supérieure de la vue vous permet choisir rapidement un type particulier de serveur (SQL Server, Analysis Server, Integration Server, or Report Server).

- ✓ Template Explorer

Fournit un accès rapide à un modèle par défaut pour éditer une requête, organisé par action, et n'importe quel modèle de modèle que vous pouvez créer. Il est possible de créer des templates pour n'importe quel langage de script supporté par SQL-SERVEUR Management Studio, SQL Server, et Analysis Server.

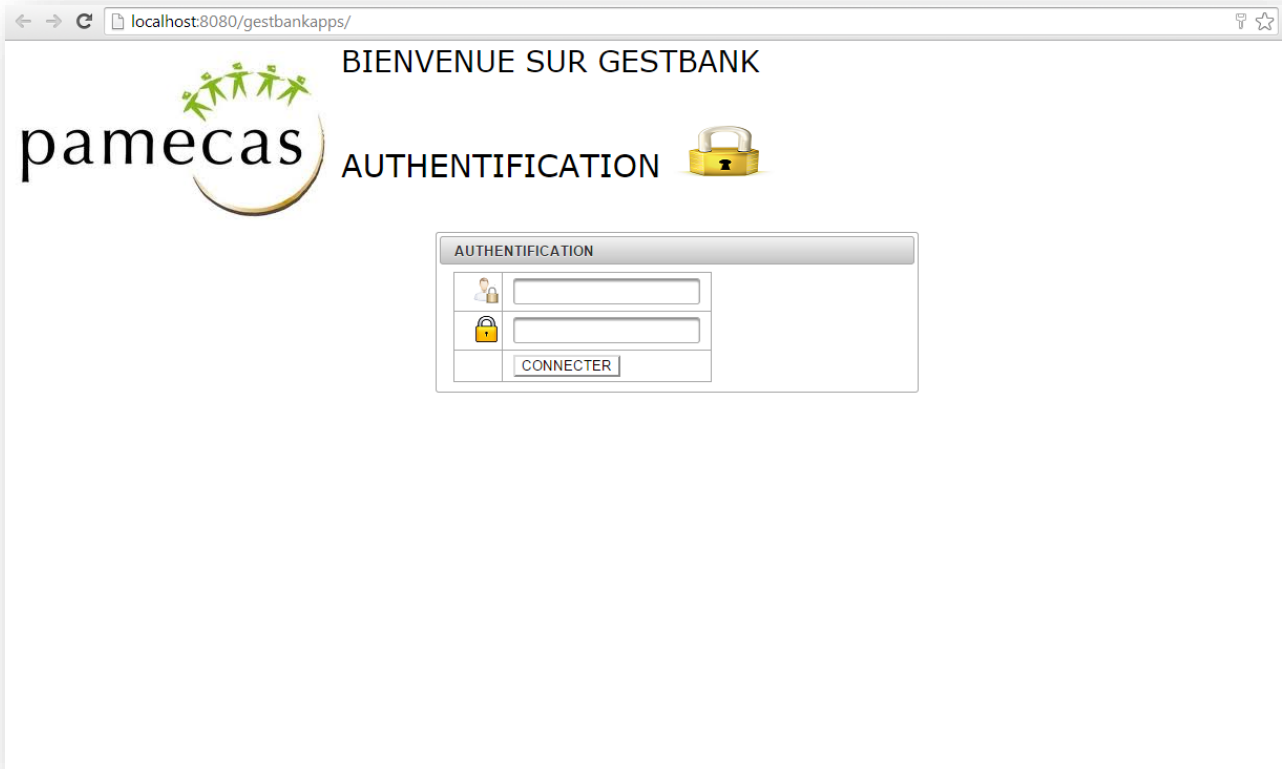
- ✓ Solution Explorer

Fournit un accès rapide aux projets SQL Server et Analysis Server existants. Les détails du projet (connexions, requêtes et les autres fonctions) sont exécutés lorsque le projet est exécuté.

## CHAPITRE 2 : LES INTERFACES USERS

Les interfaces utilisateurs de l'application ont été développées avec l'outil de développement intégré Netbeans. Nous vous présentons quelques-unes de ces interfaces.

### ✓ Interface de connexion



### ✓ Ecran d'accueil pour l'ouverture et la gestion des comptes



← → ↻ localhost:8080/gestbankapps/faces/addcompte.xhtml



# GESTBANK PAMECAS

## Ouverture de compte

**COMPTE** COMPTE BLOQUE COMPTE PRET EMPLOYEUR **Papa Moussa Cissé**

MEMBRE

STATUT:	NON SALARIE	NOM:		PRENOM:	
ADRESSE:		EMAIL:		TELEPHONE:	
DATE DE NAISSANCE:		LIEU DE NAISSANCE:		CARTE D'IDENTITE:	
SITUATION MATRIMONIALE:		NOMBRE ENFANTS:		PROFESSION:	

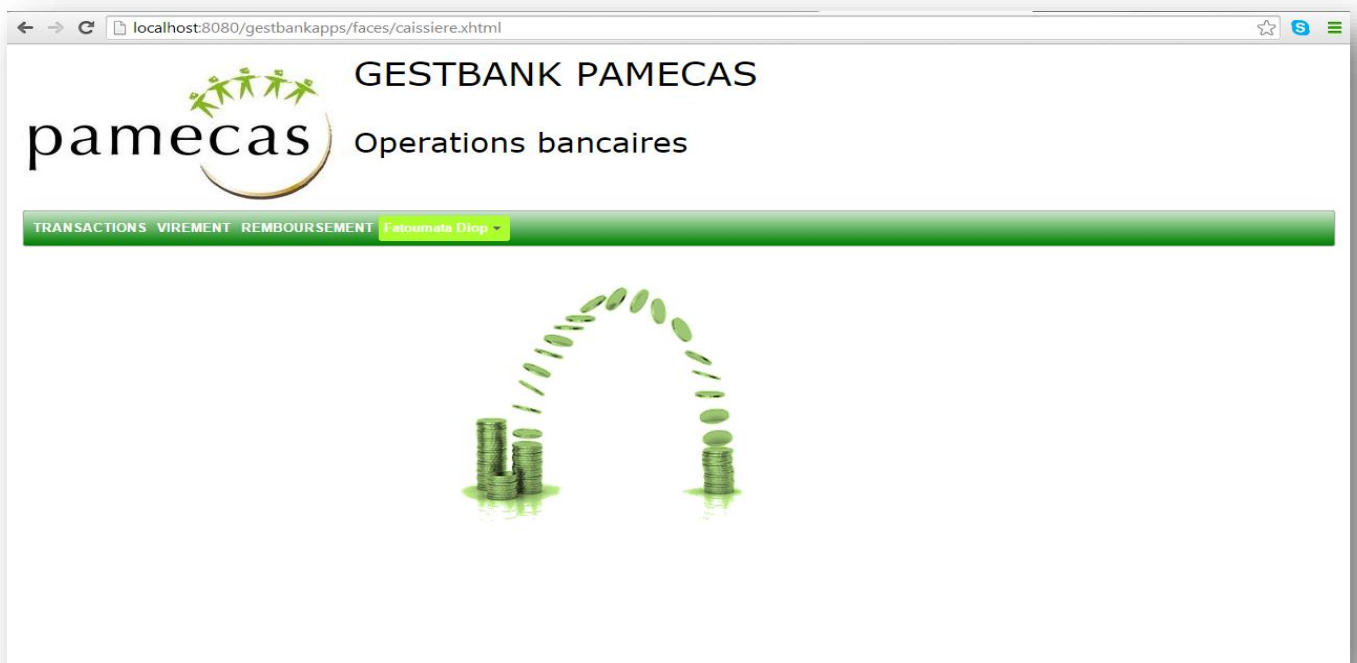
PERSONNE AYANT DROIT

NOM:		ADRESSE:		TELEPHONE:	
EMAIL:		LIEN DE PARENTE:		CARTE D'IDENTITE:	

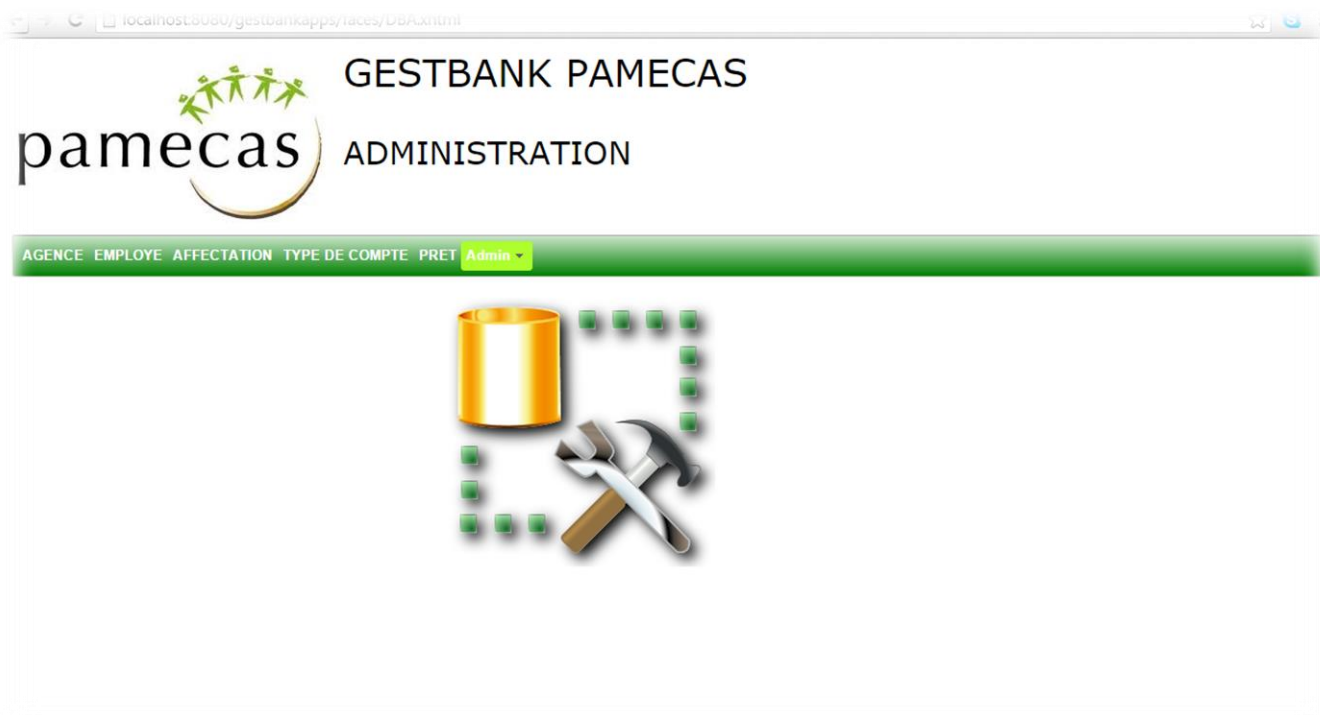
COMPTE EPARGNE PREVOYANCE

SOLDE:		AJOUTER
--------	--	---------

✓ [Interface pour la gestion des opérations bancaires](#)



✓ [Interface de paramétrage](#)



**GESTBANK PAMECAS ADMINISTRATION**

AGENCE EMPLOYE AFFECTATION TYPE DE COMPTE PRET **Admin**

#### LISTE DES CAISSIERS

Mec Yoff **SEARCH** **RESET**

(1 of 1) 1 10

ID	NOM	TELEPHONE	EMAIL	ADRESSE
4	Fatoumata Diop	771276449	fatdiop@yahoo.fr	Cité Biagui

(1 of 1) 1 10

#### FAIRE UNE AFFECTATION

DATE DEBUT:

DATE FIN:

CAISSE:

EMPLOYE:

**AJOUTER**

#### LISTE DES AFFECTATIONS

(1 of 1) 1 10

ID	DEBUT	FIN	NOM	CA
1	11/11/2014	27/11/2014	Fatoumata Diop	1

(1 of 1) 1 10

## **CHAPITRE 3 : LES SCRIPTS D'IMPLEMENTATION**

### **I. Script d'implémentation de la base de données**

```
USE [master]
GO
/***** Object: Database [gestionbancaire]  Script Date: 12/11/2014 23:32:55 *****/
CREATE DATABASE [gestionbancaire]
    CONTAINMENT = NONE
ON PRIMARY
( NAME = N'gestionbancaire', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\gestionbancaire.mdf' , SIZE =
5120KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = N'gestionbancaire_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\gestionbancaire_log.ldf' , SIZE =
2048KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [gestionbancaire] SET COMPATIBILITY_LEVEL = 110
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [gestionbancaire].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [gestionbancaire] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [gestionbancaire] SET ANSI_NULLS OFF
GO
ALTER DATABASE [gestionbancaire] SET ANSI_PADDING OFF
GO
ALTER DATABASE [gestionbancaire] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [gestionbancaire] SET ARITHABORT OFF
GO
ALTER DATABASE [gestionbancaire] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [gestionbancaire] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [gestionbancaire] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [gestionbancaire] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [gestionbancaire] SET CURSOR_CLOSE_ON_COMMIT OFF
```

```

GO
ALTER DATABASE [gestionbancaire] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [gestionbancaire] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [gestionbancaire] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [gestionbancaire] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [gestionbancaire] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [gestionbancaire] SET DISABLE_BROKER
GO
ALTER DATABASE [gestionbancaire] SET AUTO_UPDATE_STATISTICS_ASYNC
OFF
GO
ALTER DATABASE [gestionbancaire] SET
DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [gestionbancaire] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [gestionbancaire] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [gestionbancaire] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [gestionbancaire] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [gestionbancaire] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [gestionbancaire] SET RECOVERY FULL
GO
ALTER DATABASE [gestionbancaire] SET MULTI_USER
GO
ALTER DATABASE [gestionbancaire] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [gestionbancaire] SET DB_CHAINING OFF
GO
ALTER DATABASE [gestionbancaire] SET FILESTREAM(
NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [gestionbancaire] SET TARGET_RECOVERY_TIME = 0
SECONDS
GO
EXEC sys.sp_db_vardecimal_storage_formatN'gestionbancaire', N'ON'
GO

```

```

USE [gestionbancaire]
GO
/***** Object: Table [dbo].[AFFECTATION]   Script Date: 12/11/2014 23:32:55
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AFFECTATION](
    [IDAFFECTATION] [int] IDENTITY(1,1) NOT NULL,
    [DATEDEBAFFECT] [date] NULL,
    [DATEFINAFFECT] [date] NULL,
    [EXPIRATION] [bit] NULL,
    [IDEMPLOYE] [int] NULL,
    [IDCAISSE] [int] NULL,
    CONSTRAINT [PK_AFFECTATION] PRIMARY KEY CLUSTERED
(
    [IDAFFECTATION] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
/***** Object: Table [dbo].[AGENCE]   Script Date: 12/11/2014 23:32:55 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[AGENCE](
    [IDAGENCE] [int] IDENTITY(1,1) NOT NULL,
    [NOMAG] [varchar](50) NULL,
    [ADRESSEAG] [varchar](50) NULL,
    [EMAILAG] [nvarchar](50) NULL,
    [TELAG] [varchar](50) NULL,
    [MONTANTCOMPTEAG] [float] NULL,
    [IDREGION] [int] NULL,
    CONSTRAINT [PK_AGENCE] PRIMARY KEY CLUSTERED
(
    [IDAGENCE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

```



) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

/\*\*\*\*\* Object: Table [dbo].[CAISSE] Script Date: 12/11/2014 23:32:55 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[CAISSE](
    [IDCAISSE] [int] IDENTITY(1,1) NOT NULL,
    [NUMCAISSE] [int] NULL,
    [DATEOUVERTURE] [date] NULL,
    [IDAGENCE] [int] NULL,
    [disponible] [bit] NULL,
CONSTRAINT [PK_CAISSE] PRIMARY KEY CLUSTERED
(
    [IDCAISSE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

/\*\*\*\*\* Object: Table [dbo].[COMPTE] Script Date: 12/11/2014 23:32:55 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[COMPTE](
    [IDCOMPTE] [numeric](18, 0) IDENTITY(100,1) NOT NULL,
    [SOLDE] [float] NULL,
    [DATEOUV] [date] NULL,
    [DATEARCHIVAGE] [date] NULL,
    [FRAISOUV] [float] NULL,
    [NBREPRETACCORDE] [int] NULL,
    [SALAIRE] [float] NULL,
    [AGIOS] [float] NULL,
    [IDTYPECOMPTE] [int] NULL,
    [IDGESTIONNAIRE] [int] NULL,
    [IDEMPLOYEUR] [int] NULL,
    [IDPAD] [int] NULL,
    [IDCLIENTMBRE] [numeric](18, 0) NULL,
    [IDCLIENTGRP] [numeric](18, 0) NULL,
```

```

CONSTRAINT [PK_COMPTE] PRIMARY KEY CLUSTERED
(
    [IDCOMPTE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

/\*\*\*\*\* Object: Table [dbo].[COMPTEBLOQUE] Script Date: 12/11/2014 23:32:55  
\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING OFF

GO

```

CREATE TABLE [dbo].[COMPTEBLOQUE](
    [IDBLOCAGE] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DATEDEBUTBLOCAGE] [date] NULL,
    [DATEFINBLOCAGE] [date] NULL,
    [MONTANTBLOQUE] [float] NULL,
    [TAUXINTERET] [int] NULL,
    [MONTANTAREVERSER] [float] NULL,
    [ETATCOMPTE] [varchar](50) NULL,
    [IDCOMPTE] [numeric](18, 0) NULL,
    CONSTRAINT [PK_COMPTEBLOQUE] PRIMARY KEY CLUSTERED
(
    [IDBLOCAGE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

SET ANSI\_PADDING OFF

GO

/\*\*\*\*\* Object: Table [dbo].[COMPTEPRET] Script Date: 12/11/2014 23:32:55  
\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

```

CREATE TABLE [dbo].[COMPTEPRET](
    [IDCOMPTEPRET] [int] IDENTITY(1,1) NOT NULL,
    [MONTANTACCORDE] [float] NULL,
    [DATEACCORDPRET] [date] NULL,
    [MONTANTAREMBOURSER] [float] NULL,
    [MONTANTAPPORT] [float] NULL,
    [DATEDEBREMBOURSEMENT] [date] NULL,
    [ETATCOMPTE] [varchar](50) NULL,
    [MONTREMBMENSUEL] [float] NULL,
    [IDTYPEPRET] [int] NULL,
    [IDCOMPTE] [numeric](18, 0) NULL,
    CONSTRAINT [PK_COMPTEPRET] PRIMARY KEY CLUSTERED
(
    [IDCOMPTEPRET] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

SET ANSI\_PADDING OFF

GO

/\*\*\*\*\* Object: Table [dbo].[CONNEXION] Script Date: 12/11/2014 23:32:55  
\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```

CREATE TABLE [dbo].[CONNEXION](
    [IDCONNEXION] [int] IDENTITY(1,1) NOT NULL,
    [DATECONNEXION] [datetime] NOT NULL,
    [IDAFECTATION] [int] NULL,
    CONSTRAINT [PK_CONNEXION] PRIMARY KEY CLUSTERED
(
    [IDCONNEXION] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

/\*\*\*\*\* Object: Table [dbo].[EMPLOYEE] Script Date: 12/11/2014 23:32:55 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

```

GO
SET ANSI_PADDING OFF
GO
CREATE TABLE [dbo].[EMPLOYE](
    [IDEMPLOYE] [int] IDENTITY(1,1) NOT NULL,
    [MATRICULE] [nchar](10) NULL,
    [PASSWORD] [nchar](10) NULL,
    [NOMEMP] [varchar](100) NOT NULL,
    [ADRESSEEMP] [varchar](50) NULL,
    [FONCTIONEMP] [varchar](50) NULL,
    [TELEMP] [varchar](50) NULL,
    [SALAIREEMP] [bigint] NOT NULL,
    [EMAILEMP] [varchar](50) NULL,
    [CINEMP] [varchar](50) NULL,
    [IDAGENCE] [int] NULL,
    CONSTRAINT [PK_EMPLOYE] PRIMARY KEY CLUSTERED
(
    [IDEMPLOYE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[EMPLOYEUR]   Script Date: 12/11/2014 23:32:55
*****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING OFF
GO
CREATE TABLE [dbo].[EMPLOYEUR](
    [IDEMPLOYEUR] [int] IDENTITY(1,1) NOT NULL,
    [NOMEMPL] [varchar](50) NULL,
    [ADREMP] [varchar](50) NULL,
    [TELEMP] [varchar](50) NULL,
    [EMAILEMP] [varchar](50) NULL,
    CONSTRAINT [PK_EMPLOYEUR] PRIMARY KEY CLUSTERED
(
    [IDEMPLOYEUR] ASC
)

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

SET ANSI\_PADDING OFF

GO

```
/***** Object: Table [dbo].[GROUPEMENT]    Script Date: 12/11/2014 23:32:55
*****/
```

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING OFF

GO

```
CREATE TABLE [dbo].[GROUPEMENT](
    [IDCLIENTGRP] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [MATGROUP] [varchar](50) NOT NULL,
    [RAISONSOCIALE] [varchar](50) NULL,
    [NUMERORECEPISSE] [varchar](50) NULL,
    [NOMMANDATAIRE1] [varchar](50) NULL,
    [NOMMANDATAIRE2] [varchar](50) NULL,
    [CINMANDATAIRE1] [varchar](50) NULL,
    [EMAILGRP] [varchar](50) NULL,
    [TELGRP] [varchar](50) NULL,
    [CINMANDATAIRE2] [varchar](50) NULL,
    CONSTRAINT [PK_GROUPEMENT] PRIMARY KEY CLUSTERED
(
    [IDCLIENTGRP] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

SET ANSI\_PADDING OFF

GO

```
/***** Object: Table [dbo].[MEMBRE]    Script Date: 12/11/2014 23:32:55 *****/
```

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING OFF

GO

```

CREATE TABLE [dbo].[MEMBRE](
    [IDCLIENTMBRE] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [MATMEMBRE] [varchar](50) NOT NULL,
    [NOM] [varchar](50) NULL,
    [PRENOM] [varchar](50) NULL,
    [EMAILMBRE] [varchar](50) NULL,
    [TELEMBRE] [varchar](50) NULL,
    [DATENAISS] [date] NULL,
    [LIEUNAISS] [varchar](50) NULL,
    [TYPERENUMERATION] [varchar](50) NULL,
    [NUM_CIN] [varchar](50) NULL,
    [SITMATRIMONIALE] [varchar](50) NULL,
    [NBRE_ENFANTS] [nvarchar](50) NULL,
    [PROFESSION] [varchar](50) NULL,
    [ADRESSE] [varchar](50) NULL,
    CONSTRAINT [PK_MEMBRE] PRIMARY KEY CLUSTERED
(
    [IDCLIENTMBRE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[PERSONNEAYANTDROIT]    Script Date: 12/11/2014
23:32:55 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING OFF
GO
CREATE TABLE [dbo].[PERSONNEAYANTDROIT](
    [IDPAD] [int] IDENTITY(1,1) NOT NULL,
    [NOMPAD] [varchar](50) NULL,
    [ADRPAD] [varchar](50) NULL,
    [TELPAD] [varchar](50) NULL,
    [EMAILPAD] [varchar](50) NULL,
    [LIENDEPARENTE] [varchar](50) NULL,
    [CINPAD] [bigint] NULL,
    CONSTRAINT [PK_PERSONNEAYANTDROIT] PRIMARY KEY CLUSTERED
(
    [IDPAD] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]
```

GO

```
SET ANSI_PADDING OFF
```

GO

```
/***** Object: Table [dbo].[REGION]    Script Date: 12/11/2014 23:32:55 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[REGION](  
    [IDREGION] [int] IDENTITY(1,1) NOT NULL,  
    [REGION] [nvarchar](50) NULL,  
    CONSTRAINT [PK_REGION] PRIMARY KEY CLUSTERED
```

```
(  
    [IDREGION] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]
```

GO

```
/***** Object: Table [dbo].[REMBOURSEMENT]    Script Date: 12/11/2014 23:32:55
```

```
*****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[REMBOURSEMENT](  
    [IDREMBOURSEMENT] [numeric](18, 0) IDENTITY(1,1) NOT NULL,  
    [DATEREMBOURSEMENT] [date] NOT NULL,  
    [MONTANTREMBOURSE] [float] NULL,  
    [MONTANTPENALITE] [float] NULL,  
    [IDCOMPTEPRET] [int] NULL,  
    [IDCAISSE] [int] NULL,
```

```
    CONSTRAINT [PK_REMBOURSEMENT] PRIMARY KEY CLUSTERED
```

```
(  
    [IDREMBOURSEMENT] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]
```

GO

/\*\*\*\*\* Object: Table [dbo].[TRANSACTIONS] Script Date: 12/11/2014 23:32:55  
\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

```
CREATE TABLE [dbo].[TRANSACTIONS](
    [IDTRANSACTION] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DATETRANSACTION] [date] NULL,
    [TYPETRANSACTION] [varchar](50) NULL,
    [MONTANTRANSACTION] [float] NULL,
    [IDCAISSE] [int] NULL,
    [IDCOMPTE] [numeric](18, 0) NULL,
    CONSTRAINT [PK_TRANSACTION] PRIMARY KEY CLUSTERED
(
    [IDTRANSACTION] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

SET ANSI\_PADDING OFF

GO

/\*\*\*\*\* Object: Table [dbo].[TYPECOMPTE] Script Date: 12/11/2014 23:32:55  
\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[TYPECOMPTE](
    [IDTYPECOMPTE] [int] IDENTITY(1,1) NOT NULL,
    [TYPECOMPTE] [nvarchar](50) NULL,
    CONSTRAINT [PK_TYPECOMPTE] PRIMARY KEY CLUSTERED
(
    [IDTYPECOMPTE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```



```

GO
/***** Object: Table [dbo].[TYPEPRET]   Script Date: 12/11/2014 23:32:55 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING OFF
GO
CREATE TABLE [dbo].[TYPEPRET](
    [IDTYPEPRET] [int] IDENTITY(1,1) NOT NULL,
    [TYPRET] [varchar](50) NULL,
    [INTERET] [float] NULL,
    [DUREEREMBOURSEMENT] [int] NULL,
    [MONTANTMINACCORDE] [float] NULL,
    [MONTANTMAXACCORDE] [float] NULL,
    CONSTRAINT [PK_TYPEPRET] PRIMARY KEY CLUSTERED
(
    [IDTYPEPRET] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[VIREMENT]   Script Date: 12/11/2014 23:32:55 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[VIREMENT](
    [IDVIREMENT] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DATEVIREMENT] [date] NULL,
    [MONTANTVIREMENT] [float] NULL,
    [IDCOMPTE] [numeric](18, 0) NULL,
    [IDTRANSACTION] [numeric](18, 0) NULL,
    CONSTRAINT [PK_VIREMENT] PRIMARY KEY CLUSTERED
(
    [IDVIREMENT] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

```
ALTER TABLE [dbo].[AFFECTATION] WITH CHECK ADD CONSTRAINT  
[FK_AFFECTATION_CAISSE] FOREIGN KEY([IDCAISSE])  
REFERENCES [dbo].[CAISSE] ([IDCAISSE])
```

GO

```
ALTER TABLE [dbo].[AFFECTATION] CHECK CONSTRAINT  
[FK_AFFECTATION_CAISSE]
```

GO

```
ALTER TABLE [dbo].[AFFECTATION] WITH CHECK ADD CONSTRAINT  
[FK_AFFECTATION_EMPLOYE] FOREIGN KEY([IDEMPLOYE])  
REFERENCES [dbo].[EMPLOYE] ([IDEMPLOYE])
```

GO

```
ALTER TABLE [dbo].[AFFECTATION] CHECK CONSTRAINT  
[FK_AFFECTATION_EMPLOYE]
```

GO

```
ALTER TABLE [dbo].[AGENCE] WITH CHECK ADD CONSTRAINT  
[FK_AGENCE_REGION] FOREIGN KEY([IDREGION])  
REFERENCES [dbo].[REGION] ([IDREGION])
```

GO

```
ALTER TABLE [dbo].[AGENCE] CHECK CONSTRAINT [FK_AGENCE_REGION]
```

GO

```
ALTER TABLE [dbo].[CAISSE] WITH CHECK ADD CONSTRAINT  
[FK_CAISSE_AGENCE] FOREIGN KEY([IDAGENCE])  
REFERENCES [dbo].[AGENCE] ([IDAGENCE])
```

GO

```
ALTER TABLE [dbo].[CAISSE] CHECK CONSTRAINT [FK_CAISSE_AGENCE]
```

GO

```
ALTER TABLE [dbo].[COMPTE] WITH CHECK ADD CONSTRAINT  
[FK_COMPTE_EMPLOYE] FOREIGN KEY([IDGESTIONNAIRE])  
REFERENCES [dbo].[EMPLOYE] ([IDEMPLOYE])
```

GO

```
ALTER TABLE [dbo].[COMPTE] CHECK CONSTRAINT  
[FK_COMPTE_EMPLOYE]
```

GO

```
ALTER TABLE [dbo].[COMPTE] WITH CHECK ADD CONSTRAINT  
[FK_COMPTE_EMPLOYEUR] FOREIGN KEY([IDEMPLOYEUR])  
REFERENCES [dbo].[EMPLOYEUR] ([IDEMPLOYEUR])
```

GO

```
ALTER TABLE [dbo].[COMPTE] CHECK CONSTRAINT  
[FK_COMPTE_EMPLOYEUR]
```

GO

```
ALTER TABLE [dbo].[COMPTE] WITH CHECK ADD CONSTRAINT  
[FK_COMPTE GROUPEMENT] FOREIGN KEY([IDCLIENTGRP])  
REFERENCES [dbo].[GROUPEMENT] ([IDCLIENTGRP])
```

```

GO
ALTER TABLE [dbo].[COMPTE] CHECK CONSTRAINT
[FK_COMPTE_GROUPEMENT]
GO
ALTER TABLE [dbo].[COMPTE] WITH CHECK ADD CONSTRAINT
[FK_COMPTE_MEMBRE] FOREIGN KEY([IDCLIENTMBRE])
REFERENCES [dbo].[MEMBRE] ([IDCLIENTMBRE])
GO
ALTER TABLE [dbo].[COMPTE] CHECK CONSTRAINT [FK_COMPTE_MEMBRE]
GO
ALTER TABLE [dbo].[COMPTE] WITH CHECK ADD CONSTRAINT
[FK_COMPTE_PERSONNEAYANTDROIT] FOREIGN KEY([IDPAD])
REFERENCES [dbo].[PERSONNEAYANTDROIT] ([IDPAD])
GO
ALTER TABLE [dbo].[COMPTE] CHECK CONSTRAINT
[FK_COMPTE_PERSONNEAYANTDROIT]
GO
ALTER TABLE [dbo].[COMPTE] WITH CHECK ADD CONSTRAINT
[FK_COMPTE_TYPECOMPTE] FOREIGN KEY([IDTYPECOMPTE])
REFERENCES [dbo].[TYPECOMPTE] ([IDTYPECOMPTE])
GO
ALTER TABLE [dbo].[COMPTE] CHECK CONSTRAINT
[FK_COMPTE_TYPECOMPTE]
GO
ALTER TABLE [dbo].[COMPTEBLOQUE] WITH CHECK ADD CONSTRAINT
[FK_COMPTEBLOQUE_COMPTE] FOREIGN KEY([IDCOMPTE])
REFERENCES [dbo].[COMPTE] ([IDCOMPTE])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[COMPTEBLOQUE] CHECK CONSTRAINT
[FK_COMPTEBLOQUE_COMPTE]
GO
ALTER TABLE [dbo].[COMPTEPRET] WITH CHECK ADD CONSTRAINT
[FK_COMPTEPRET_COMPTE] FOREIGN KEY([IDCOMPTE])
REFERENCES [dbo].[COMPTE] ([IDCOMPTE])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[COMPTEPRET] CHECK CONSTRAINT
[FK_COMPTEPRET_COMPTE]
GO
ALTER TABLE [dbo].[COMPTEPRET] WITH CHECK ADD CONSTRAINT
[FK_COMPTEPRET_TYPEPRET] FOREIGN KEY([IDTYPEPRET])
REFERENCES [dbo].[TYPEPRET] ([IDTYPEPRET])

```

```

ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[COMPTEPRET] CHECK CONSTRAINT
[FK_COMPTEPRET_TYPEPRET]
GO
ALTER TABLE [dbo].[CONNEXION] WITH CHECK ADD CONSTRAINT
[FK_CONNEXION_AFFECTATION] FOREIGN KEY([IDAFFECTATION])
REFERENCES [dbo].[AFFECTATION] ([IDAFFECTATION])
GO
ALTER TABLE [dbo].[CONNEXION] CHECK CONSTRAINT
[FK_CONNEXION_AFFECTATION]
GO
ALTER TABLE [dbo].[EMPLOYE] WITH CHECK ADD CONSTRAINT
[FK_EMPLOYE_AGENCE] FOREIGN KEY([IDAGENCE])
REFERENCES [dbo].[AGENCE] ([IDAGENCE])
GO
ALTER TABLE [dbo].[EMPLOYE] CHECK CONSTRAINT
[FK_EMPLOYE_AGENCE]
GO
ALTER TABLE [dbo].[REMBOURSEMENT] WITH CHECK ADD CONSTRAINT
[FK_REMBOURSEMENT_CAISSE] FOREIGN KEY([IDCAISSE])
REFERENCES [dbo].[CAISSE] ([IDCAISSE])
GO
ALTER TABLE [dbo].[REMBOURSEMENT] CHECK CONSTRAINT
[FK_REMBOURSEMENT_CAISSE]
GO
ALTER TABLE [dbo].[REMBOURSEMENT] WITH CHECK ADD CONSTRAINT
[FK_REMBOURSEMENT_COMPTEPRET1] FOREIGN KEY([IDCOMPTEPRET])
REFERENCES [dbo].[COMPTEPRET] ([IDCOMPTEPRET])
GO
ALTER TABLE [dbo].[REMBOURSEMENT] CHECK CONSTRAINT
[FK_REMBOURSEMENT_COMPTEPRET1]
GO
ALTER TABLE [dbo].[TRANSACTIONS] WITH CHECK ADD CONSTRAINT
[FK_TRANSACTION_CAISSE] FOREIGN KEY([IDCAISSE])
REFERENCES [dbo].[CAISSE] ([IDCAISSE])
GO
ALTER TABLE [dbo].[TRANSACTIONS] CHECK CONSTRAINT
[FK_TRANSACTION_CAISSE]
GO
ALTER TABLE [dbo].[TRANSACTIONS] WITH CHECK ADD CONSTRAINT
[FK_TRANSACTION_COMPTE] FOREIGN KEY([IDCOMPTE])
REFERENCES [dbo].[COMPTE] ([IDCOMPTE])
GO

```

```

ALTER TABLE [dbo].[TRANSACTIONS] CHECK CONSTRAINT
[FK_TRANSACTION_COMPTE]
GO
ALTER TABLE [dbo].[VIREMENT] WITH CHECK ADD CONSTRAINT
[FK_VIREMENT_COMPTE] FOREIGN KEY([IDCOMPTE])
REFERENCES [dbo].[COMPTE] ([IDCOMPTE])
GO
ALTER TABLE [dbo].[VIREMENT] CHECK CONSTRAINT
[FK_VIREMENT_COMPTE]
GO
ALTER TABLE [dbo].[VIREMENT] WITH CHECK ADD CONSTRAINT
[FK_VIREMENT_TRANSACTION] FOREIGN KEY([IDTRANSACTION])
REFERENCES [dbo].[TRANSACTIONS] ([IDTRANSACTION])
GO
ALTER TABLE [dbo].[VIREMENT] CHECK CONSTRAINT
[FK_VIREMENT_TRANSACTION]
GO
USE [master]
GO
ALTER DATABASE [gestionbancaire] SET READ_WRITE
GO

```

## II. L'agent SQL-Serveur

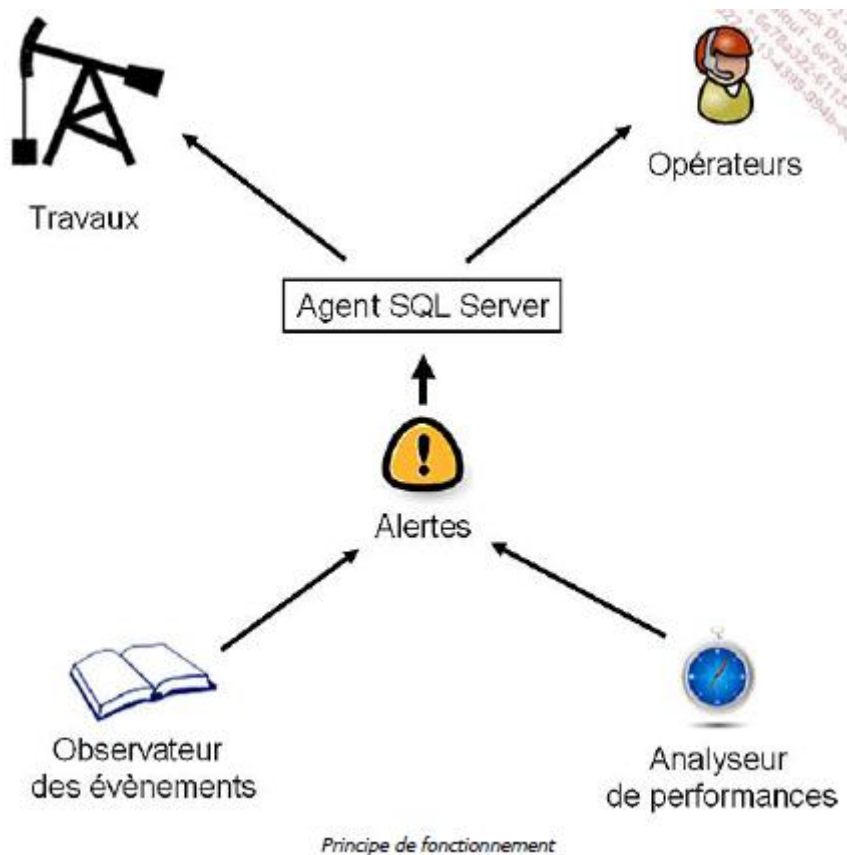
SQL Server Agent est un service Microsoft Windows qui vous permet d'automatiser certaines tâches d'administration. Le service SQL Server Agent exécute des travaux, surveille SQL Server et traite les alertes. Le service SQL Server Agent doit être en cours d'exécution pour que les travaux d'administration locale ou multiserveur puissent être exécutés automatiquement.

Dans notre cas, nous lui donnerons certaines tâches à exécuter comme le déblocage du compte bloqué à une heure donnée de la journée

```
UPDATE COMPTEBLOQUE
```

```
SET ETATCOMPTE='off'
```

```
WHERE DATEDIFF (YEAR, DATEDEBUTBLOCAGE, DATEFINBLOCAGE) =0;
```



## CONCLUSION

Ce projet a été abordé dans le but d'améliorer le système d'information du système financier décentralisé pamecas. La problématique ainsi que les objectifs de l'étude nous ont permis de proposer une solution qui est l'informatisation du SI. Une critique de l'analyse de l'existant a donné comme résultats de déceler les insuffisances au niveau de son système d'information.

L'étude du projet ainsi que sa conception ont été réalisées et sa phase de réalisation a été abordée. Le résultat du projet est le passage des procédures de gestion manuelles vers les procédures de gestion informatisées.

Après la phase de réalisation de l'informatisation, des études pour une migration vers les technologies mobiles, la consultation du solde par sms et le renforcement de la sécurité de l'application seront des perspectives pour mieux atteindre l'objectif général d'un système d'information : « obtenir la bonne information au bon moment ».

## ANNEXES

### BIBLIOGRAPHIE

UML 2.0, (IUT, département informatique, 1re année), Laurent AUDIBERT  
JEE7 recipes Josh Juneau, EDITION APRESS  
SQL-Serveur pocket consultant

### WEBOGRAPHIE

[www.primefaces.org](http://www.primefaces.org)

[www.wikipedia.org](http://www.wikipedia.org)

<http://www-igm.univ-mlv.fr/>

<https://www.websequencediagrams.com/>