

# LOG1000 – TP2

Samuel Rondeau

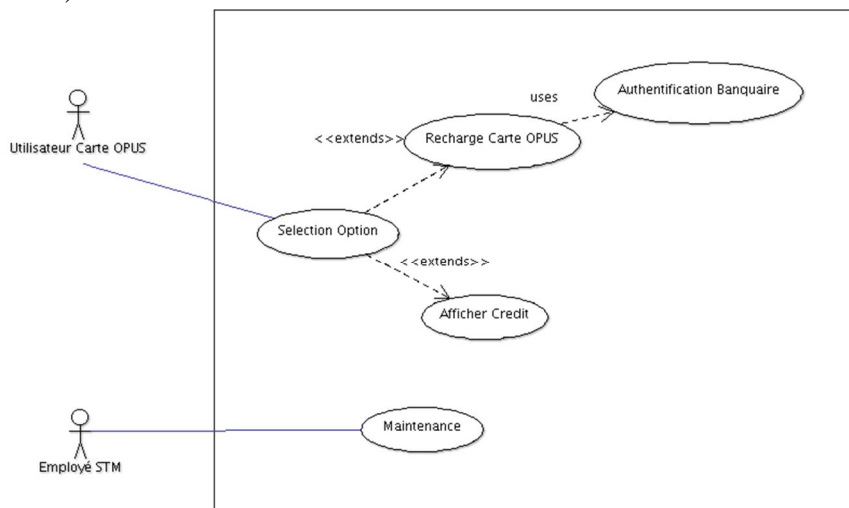
Pacôme Bondet

E 1.1)

Les acteurs sont:

- Client STM
- Employé STM
- Terminal de Paiement (externe)
- Serveur STM (externe)

E 1.2)



E 1.3)

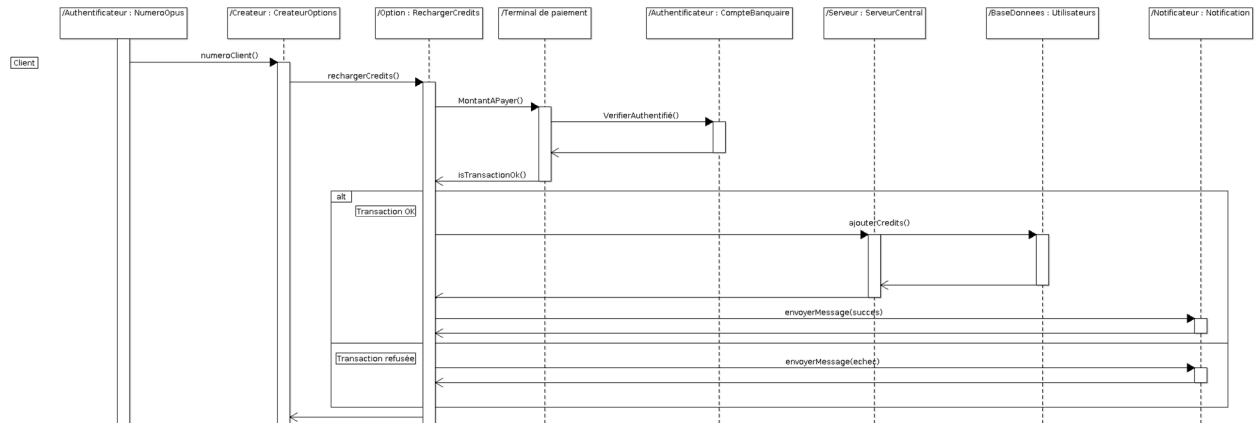
<b>Nom du cas</b>	Recharger carte Opus
<b>Description</b>	Le client désire rajouter des crédits sur sa carte Opus
<b>Précondition</b>	Le client doit posséder une carte Opus
<b>Postcondition (succès)</b>	Le client obtient de nouveaux crédits sur sa carte Opus
<b>Postcondition (échec)</b>	Le client n'obtient pas de nouveaux crédits
<b>Acteurs</b>	Client, (externes : terminal de paiement, serveur central)
<b>Déclencheur</b>	Le client insère sa carte Opus dans le guichet. C'est l'option par défaut.
<b>Scénario principal</b>	<ol style="list-style-type: none"> <li>1. Le système authentifie le client</li> <li>2. Le client demande d'acheter des crédits</li> <li>3. La banque authentifie les informations bancaires du client</li> <li>4. Le serveur central ajoute les crédits au compte du client</li> <li>5. Le guichet affiche un message de succès et ferme la session</li> </ol>
<b>Extensions</b>	<ol style="list-style-type: none"> <li>1.1. Le système ne peut authentifier la carte <ol style="list-style-type: none"> <li>1.1.1. La session est terminée, le guichet demande au client de contacter un employé</li> </ol> </li> <li>2.1. Le client ne fait aucune action en 60 secondes ou annule</li> <li>3.1. La banque refuse la transaction (mauvais NIP, fonds, etc.) <ol style="list-style-type: none"> <li>3.1.1. Le système affiche un message d'échec et ferme la session</li> </ol> </li> <li>4.1. Le système ne peut communiquer avec le serveur <ol style="list-style-type: none"> <li>4.1.1. La transaction est annulée</li> <li>4.1.2. Le système affiche un message d'erreur et ferme la session</li> </ol> </li> </ol>

<b>Nom du cas</b>	Consulter crédits
<b>Description</b>	Le client désire consulter les crédits sur sa carte Opus
<b>Précondition</b>	Le client doit posséder une carte Opus
<b>Postcondition</b>	Le client est informé du nombre de crédits sur sa carte Opus
<b>Acteurs</b>	Client, (externe : serveur central)
<b>Déclencheur</b>	Le client insère sa carte Opus dans le guichet, puis sélectionne l'option « Consulter crédits »
<b>Scénario principal</b>	<ol style="list-style-type: none"> <li>1. Le système authentifie le client</li> <li>2. Le client demande de consulter ses crédits</li> <li>3. Le serveur renvoie les informations du compte du client</li> <li>4. Après 60 secondes ou le retrait de la carte, le système ferme la session</li> </ol>
<b>Extensions</b>	<ol style="list-style-type: none"> <li>1.1. Le système ne peut authentifier la carte <ol style="list-style-type: none"> <li>1.1.1. La session est terminée, le guichet demande au client de contacter un employé</li> </ol> </li> <li>2.1. Le client ne fait aucune action en 60 secondes ou annule</li> <li>3.1. Le système ne peut communiquer avec le serveur <ol style="list-style-type: none"> <li>3.1.1. Le système affiche un message d'erreur et ferme la session</li> </ol> </li> </ol>

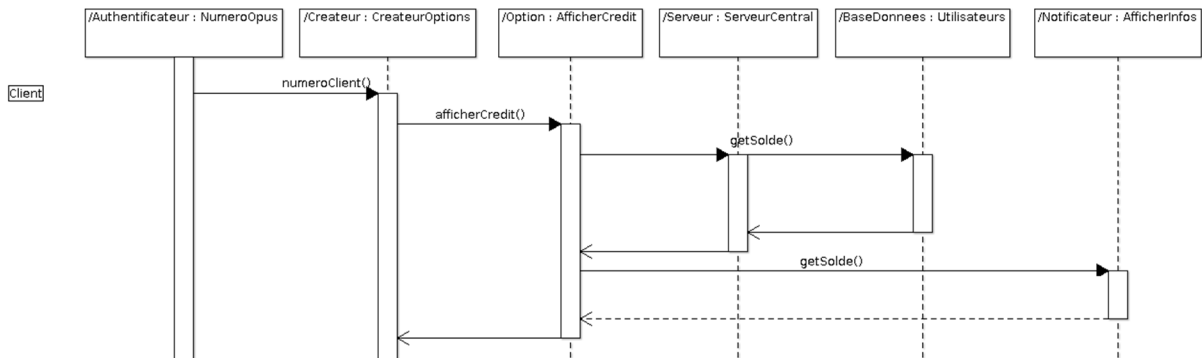
<b>Nom du cas</b>	Vérifier santé système
<b>Description</b>	Un employé vérifie la santé du système
<b>Précondition</b>	L'employé doit posséder une carte d'employé
<b>Postcondition (succès)</b>	L'employé est avisé de la santé du système
<b>Postcondition (échec)</b>	L'employé est avisé d'une erreur du système
<b>Acteurs</b>	Employé, (externe : serveur central)
<b>Déclencheur</b>	L'employé insère sa carte Opus dans le guichet
<b>Scénario principal</b>	<ol style="list-style-type: none"> <li>1. Le système authentifie l'employé</li> <li>2. L'employé démarre la maintenance du système</li> <li>3. Le système communique avec le serveur central</li> <li>4. Le guichet affiche un message de succès et ferme la session</li> </ol>
<b>Extensions</b>	<ol style="list-style-type: none"> <li>1.1. Le système ne peut authentifier la carte <ol style="list-style-type: none"> <li>1.1.1. La session est terminée</li> </ol> </li> <li>2.1. L'employé ne fait aucune action en 60 secondes ou annule</li> <li>3.1. Le système ne peut communiquer avec le serveur en 30 secondes <ol style="list-style-type: none"> <li>3.1.1. Le système affiche un message d'erreur et ferme la session</li> </ol> </li> </ol>

## E 2.1)

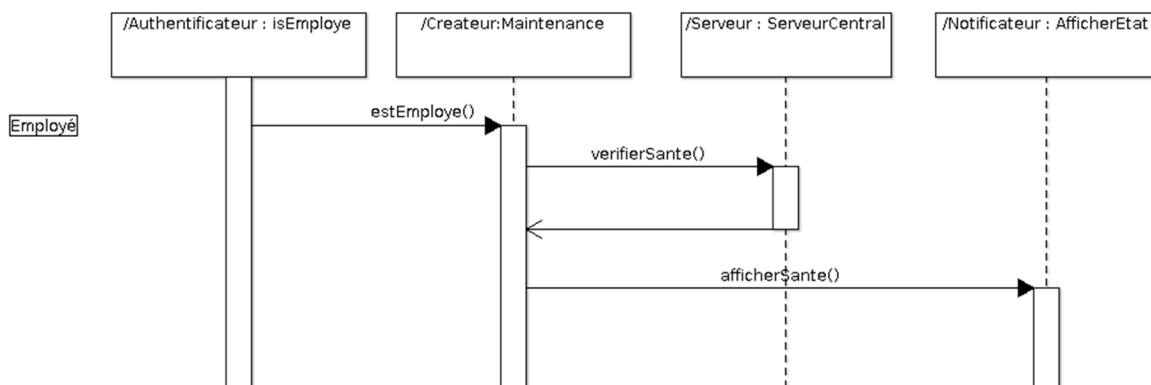
### Cas Recharger carte Opus :



### Cas Consulter crédits :



### Cas Vérifier santé système :



### E 3.1)

Les classes à implémenter dans le système sont les suivantes:

(Dans notre diagramme de classes, toutes les méthodes sont publiques)

Classes:

#### 1. Options

Cette classe inclue deux méthodes: RechargeCredit() et AfficherCredit() qui sont nos deux options d'utilisation du système et utilise les classes: AuthentificateurOpus, CompteUtilisateur et Notificateur.

#### 2. AuthentificateurOpus

Cette class permet d'identifier l'utilisateur, soit l'utilisateur est un client ayant une carte OPUS ou un employé de la STM. Il comporte deux méthodes: GetNumeroClient() (pour connaitre de quel client il s'agit) et isEmployé() qui retourne vrai si l'utilisateur est un employé de la STM et utilise la classe SantéSystem si c'est le cas.

#### 3. CompteUtilisateur

Cette classe comporte les informations d'un client de la STM: son crédit et le numéro de l'utilisateur (qui est le même récupéré par AuthentificateurOpus avec GetNumeroClient). Elle comporte deux méthodes: AjouterCredit() qui permet à l'utilisateur d'ajouter du crédit a son compte et GetCredit qui retourne la valeur du crédit disponible dans le compte.

#### 4. Notificateur

Cette classe permet d'afficher à l'écran des messages pour l'utilisateur du système avec sa méthode afficherMessage().

#### 5. AuthentificateurBancaire

Cette classe est appelée quand l'utilisateur souhaite ajouter du crédit à son compte avec son compte bancaire, elle permet d'effectuer la transaction.

Ces attributs privés sont: le NIP de l'utilisateur, le numéro bancaire et le montant du compte. Elle comporte deux méthodes: VerfierAuthentifié() qui vérifie si le code NIP est bien correct et VerifierSolde() qui vérifie si l'utilisateur a bien assez de fonds pour effectuer la transaction.

#### 6. SantéSysteme

Cette classe est appelée seulement si l'utilisateur a été identifié comme un employé de la STM. Elle comporte une méthode CommuniquerServeur() qui vérifie la santé du système et modifie son attribut "estFonctionel" qui est vrai si le système est en bonne santé et faux autrement.

E 3.2)

