

# Génie logiciel

IFT-2255

## Devoir 3

### Informations générales

**Nom :** Jonathan Larose

**Matricule :** 20066082

**Courriel :** [jonathan.larose@umontreal.ca](mailto:jonathan.larose@umontreal.ca)

**Temps :** 35 heures

**Nom :** Augusto dos Santos Latgé

**Matricule :** 20083794

**Courriel :** [augusto.dos.santos.lagte@umontreal.ca](mailto:augusto.dos.santos.lagte@umontreal.ca)

**Temps :** 35 heures

**Nom :** Dennis Orozco Martinez

**Matricule :** 20031060

**Courriel :** [dennis.alberto.orozco.martinez@umontreal.ca](mailto:dennis.alberto.orozco.martinez@umontreal.ca)

**Temps :** 35 heures

**Nom :** Aliona Cupcea

**Matricule :** 20057887

**Courriel :** [aliona.cupcea@umontreal.ca](mailto:aliona.cupcea@umontreal.ca)

**Temps :** 35 heures

### Distribution des tâches

**Soumissionnaire :** Jonathan Larose

Répartition des classes à implémenter pour le TP3 :

	Jonathan Larose	Aliona Cupcea	Augusto Dos Santos Latgé	Dennis Orozco Martinez
ChocAn.java	x			
UI	X			
UIAcme	X			
UIFournisseur	X			
UIGerant			X	
UIOperateurChocAn			X	
ChocAnUI	x			
Horloge			x	

Fournisseur		x		
Membre		x		
StatutMembre		x		
ListeDeService				x
RepertoireDesFournisseur				X
Service				X
ServiceRendu				x
ListeDeCompte	x			
Compte		X		
Test : ListeDeCompte				x
Test : UITest		x		
Test : RepertoireDesFournisseur trierListeAlpha()	x			
Test : RepertoireDesFournisseur2 trouverService()			X	

## WORKFLOW DES EXIGENCES

### Analyse

Pour que le système soit fonctionnel, les informations des fournisseurs, des membres et des services offerts devront être informatisées dans le centre de données par un opérateur ChocAn. Chacun aura un identifiant numérique unique. Il y aura un serveur sur lequel les différents utilisateurs pourront se connecter via des terminaux (type carte de crédit). Un ordinateur d'Acme et les ordinateurs des opérateurs ChocAn seront également connecté à ce serveur. Il y aura aussi une base de données pour stocker les informations. Les systèmes connectés au serveur auront accès à certaines classes de contrôle via leur interface spécifique à chacun. Chaque interface pourra faire des actions limités et autorisés. Le logiciel recevra des requêtes et accèdera au centre des données pour faire l'opération demandé.

## Glossaire

**Acme** : Organisation tierce de comptabilité responsable des procédures financières.

**Centre de données ChocAn** : Base de données contenant toutes les informations sur les membres et fournisseurs *ChocAn*.

**ChocAn** : Abréviation de Chocolique Anonyme.

**Code de service** : Code à 6 chiffres correspondant au service fourni (ex. 598470 -> diététicien)

**Données de TEF** : Les données de TEF regroupent :

- Nom de fournisseur
- Numéro de fournisseur
- Montant à transférer

**Formulaire du fournisseur** : Formulaire qu'un fournisseur remplit après avoir donné un service et dont le but est de vérifier que le montant que *ChocAn* lui verse à la fin de la semaine correspond bien au montant total des services qu'il a offert.

**Fournisseur** : Professionnel de la santé qui fournit un ou des services à des membres de *ChocAn*

**Liste des fournisseurs** : Liste de tous les fournisseurs faisant affaire avec *ChocAn* et leurs informations.

**Liste des membres** : Liste de tous les membres de *ChocAn* et leurs informations.

**Liste de service** : Liste de tous les services fournis à des membres par des fournisseurs *ChocAn* durant la semaine courante.

**Membre** : Personne abonnée mensuellement à *ChocAn* afin de recevoir des services.

**Nom de service** : Identifiant de maximum 20 caractères représentant un service fourni par un fournisseur.

**Numéro de fournisseur** : Numéro à 9 chiffres représentant un fournisseur

**Numéro de membre** : Numéro à 9 chiffres représentant un membre.

**Rapport de service** : Rapport de fournisseur ou rapport de membre.

**Rapport fournisseur** : Bilan des services fournis par un fournisseur durant la semaine.

- Nom du fournisseur (25 caractères)
- Numéro du fournisseur (9 chiffres)
- Adresse du fournisseur (25 caractères)
- Ville du fournisseur (14 caractères)
- Province du fournisseur (2 lettres)
- Code postal du fournisseur (5 chiffres)
- Pour chaque service fourni, les détails suivants sont requis :
  - Date du service (JJ-MM-AAAA)
  - Date et heure à laquelle les données étaient reçues par l'ordinateur (JJ-MM-AAAA HH:MM:SS)
  - Nom du membre (25 caractères)
  - Numéro du membre (9 chiffres)
  - Code du service (6 chiffres)
  - Montant à payer (jusqu'à 999.99\$).
- Résumé de la semaine :
  - Nombre de consultations avec les membres
  - Total des frais de la semaine

**Rapport membre** : Bilan des services reçus par un membre durant la semaine.

- Nom du membre (25 caractères)
- Numéro du membre (9 chiffres)
- Adresse du membre (25 caractères)

- Ville du membre (14 caractères)
- Province du membre (2 lettres)
- Code postal du membre (5 chiffres).
- Pour chaque service fourni, les détails suivants sont requis :
  - Date du service (JJ-MM-AAA)
  - Nom du fournisseur (25 caractères)
  - Nom du service (20 caractères).

**Rapport de synthèse** : Rapport hebdomadaire produit pour le gérant de *ChocAn* qui décrit la liste des fournisseurs qui doivent être payés.

- Liste des fournisseurs devant être payé, pour chacun :
  - Nombre de consultation
  - Total des frais
- Résumé de la semaine :
  - Nombre total de fournisseurs ayant fourni des services cette semaine-là
  - Nombre total de consultation
  - Frais total

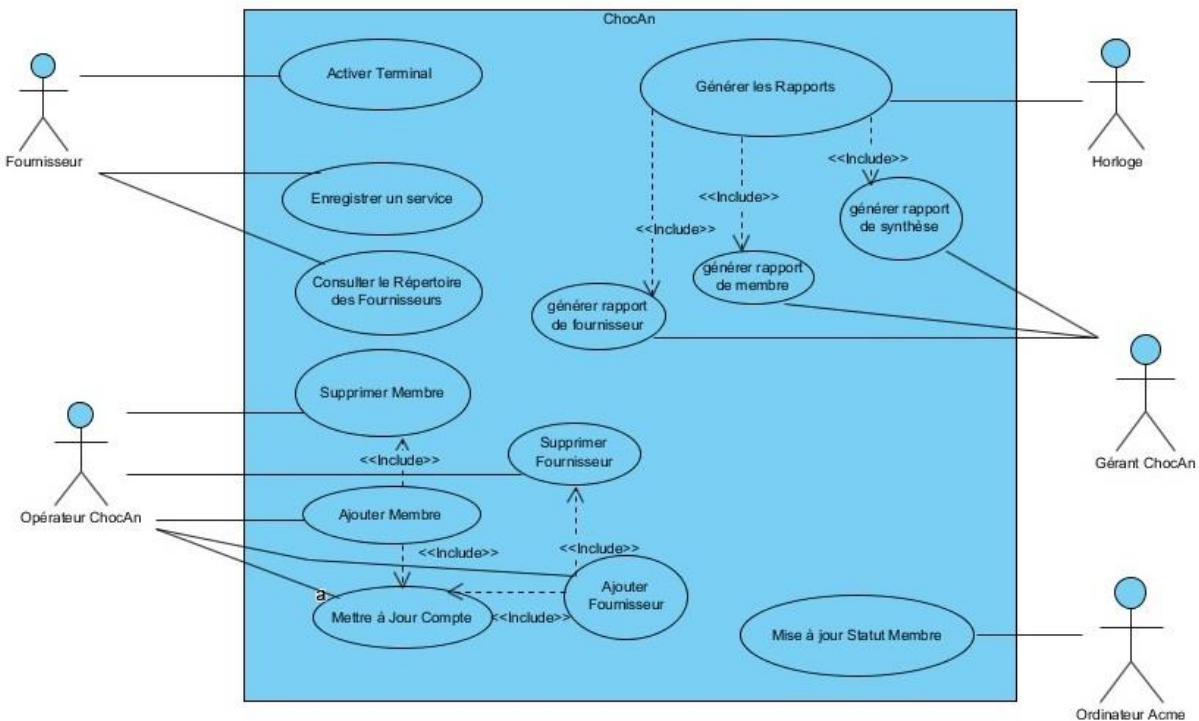
**Répertoire des fournisseurs** : Liste ordonnée alphabétiquement des noms des différents services, avec les codes et frais correspondant.

**Service** : Soins offerts à un membre par un fournisseur.

**Statut d'un membre** : État de l'abonnement d'un membre : Actif ou suspendu.

**TEF** : Abréviation de transfert électronique de fonds.

## Diagramme de cas d'utilisation



[Diagram ChocAn.vpp](#)

### Cas d'utilisation

#### Cas d'utilisation : Activer Terminal

**But :** Associé un terminal à un fournisseur.

**Préconditions :** Le fournisseur a un terminal

**Postconditions :** Le terminal est associé au fournisseur et connecté au système. Le système est prêt à recevoir des entrées du terminal.

**Acteurs :** Fournisseur

#### Scénario principal :

1. Le fournisseur allume le terminal.
2. Le système demande un numéro de fournisseur.
3. Le fournisseur entre son numéro de fournisseur.
4. Le système vérifie que le numéro est bien dans la liste des fournisseurs.
5. Le système retourne que le numéro est valide.
6. Le terminal est connecté.

#### Scénario alternatif :

- 4a. Le numéro n'est pas dans la liste des fournisseurs
  - 4.1 Le terminal affiche « numéro est invalide ».
  - 4.2 Retour à 2.

## **Cas d'utilisation : Enregistrer un service**

**But :** Comptabiliser un service qui sera fourni à un membre *ChocAn*.

**Préconditions :** Le terminal du fournisseur est activé. Un membre se présente avec sa carte de membre. La carte est glissée dans le terminal.

**Acteurs :** Fournisseur, terminal du fournisseur

Comme le fournisseur fait toute les interactions avec le logiciel via le terminal, nous considérons le fournisseur et son terminal comme l'acteur principal dans ce cas d'utilisation.

### **Scénario principal :**

1. Le système reçoit un numéro de membre.
2. Le système valide le numéro de membre
  - 2.1 'Valide' s'affiche sur le terminal.
3. Le système demande un code de service.
4. Le fournisseur entre un code de service
  - 4.1 Le terminal envoie le code de service au système.
5. Le système reçoit et trouve le code service dans le Répertoire des Fournisseurs.
  - 5.1 Le système envoie le nom du service correspondant au code.
  - 5.2 Le terminal affiche le nom.
6. Le système demande une confirmation que le nom correspond bien au service fourni.
  - 6.1 Le fournisseur entre oui ou non :
  - 6.2 Le fournisseur entre oui.
7. Le système demande la date à laquelle le service a été fourni.
8. Le fournisseur entre une date.
9. Le système demande si le fournisseur souhaite entrer un commentaire par rapport au service fourni :
  - 9.1 Le fournisseur entre oui ou non :
  - 9.2 Dans le cas oui :
    - 9.2.1 Le fournisseur entre le commentaire.
    - 9.2.2 Le système reçoit le commentaire.
    - 9.2.3 Continuer à l'étape 11.
  - 9.3 Dans le cas non :
    - 9.3.1 Continuer à l'étape 11.
11. Le système crée un enregistrement dans la liste de service.
12. Le système envoie le montant associé au service.
  - 12.1 Le terminal affiche le montant.
13. Retour à l'état initial.

### **Scénario alternatif :**

- 2a. Le numéro de membre est invalide.
  - 2a.1 Afficher 'Numéro Invalide'
  - 2a.2 Retour à l'état initial
- 2b. Le membre est suspendu
  - 2b.1 Afficher 'Membre Suspendu'
  - 2b.2 Afficher le message que le membre est suspendu à cause de frais non payé pour au moins un mois.
  - 2b.3 Retour à l'état initial

- 4a. Le code de service reçu par le terminal n'est pas valide
  - 4a.1 Le système affiche 'code de service non-existant'
  - 4a.2 Retour à 3.

- 6.2b Le fournisseur entre non.
  - 6.2b.1 Retour à l'étape 5.

Il faudrait peut-être aussi considérer un temps d'attente trop long d'une réponse du terminal. Par exemple, si aucune réponse n'est envoyée par le terminal après 1 minute, on annule le cas et on retourne à l'état initial. On considère aussi qu'il y a un bouton sur le terminal qui permet au fournisseur d'annuler l'enregistrement de service à tout moment lors du CU.

### **Cas d'utilisation : Consulter le Répertoire des Fournisseurs**

**But :** Envoyer le répertoire des fournisseurs à un fournisseur à sa demande.

**Préconditions :** Le terminal du fournisseur est activé.

**Acteurs :** Fournisseur

On considère qu'il y a un bouton sur le terminal du fournisseur qui lui permet d'envoyer une requête à tout moment lorsque le terminal est activé. On assume que le Répertoire des Fournisseurs est un fichier qui est mis-à-jour lorsque des services s'ajoute et s'enlève, ou bien que les tarifs associés aux services changent.

#### **Scénario principal :**

1. Le système reçoit une requête d'accès au Répertoire des Fournisseurs.
2. Le système trouve le fournisseur associé au terminal dans la liste des fournisseurs.
  - 2.1 Le système trouve l'adresse courriel du fournisseur.
3. Le système envoie par courriel en pièce jointe le Répertoire des Fournisseurs.

### **Cas d'utilisation : Générer les Rapports**

**But :** Le système produit les rapports hebdomadaires : rapports de fournisseur, rapports de membre, rapport de synthèse. Le système enregistre aussi les données TEF.

**Préconditions :** Vendredi minuit

**Acteurs :** Principal : Horloge

Récepteurs : *Acme*, Fournisseurs, Membres, Gérant *ChocAn*

**\*\*** Les récepteurs sont les acteurs qui ne sont pas actif, mais passif. Ils recevront les différents rapports produits automatiquement par le système.

#### **Scénario principal :**

1. Pour chaque fournisseur dans la liste de service :
  - 1.1 *accès CU générer rapport de fournisseur*
  - 1.2 Le système envoi un rapport à chacun des fournisseurs concernés.
2. Pour chaque membre dans la liste de service :
  - 2.1 *accès CU générer rapport de membre*

- 2.2 Le système envoie un rapport à chacun des membres concernés.
3. Le système fait l'enregistrement des données TEF à partir de la liste de service.
4. *accès CU générer rapport de synthèse*
  - 4.1 Le rapport de synthèse est envoyé au gérant.
5. Back up la liste de service de la semaine.
6. Met à zéro la liste pour la semaine suivante.
7. Retour à l'état initial.

**Scénario alternatif :**

- 1a. Liste de service est vide (pas de service fourni cette semaine)
  - 1a.1 Aller directement à l'étape 3.

On a cru pertinent de garder une copie de la liste de service de chaque semaine au cas où il y aurait des problèmes dans la création des rapports, d'où l'étape 5. On profite de ce cas d'utilisation pour remettre la liste de service à zéro pour la semaine à venir.

**Cas d'utilisation : générer un rapport de fournisseur**

**But :** Produire un rapport de fournisseur.

**Préconditions :** Le système a reçu la demande de générer un rapport de fournisseur avec un numéro de fournisseur.

**Acteurs :** Gérant *ChocAn* (ou Horloge)

**Scénario principal :**

1. Le système trouve les informations associées au numéro de fournisseur dans la liste des fournisseurs.
2. Le système trouve tous les services fournis par le fournisseur dans la liste de service.
3. Le système génère le fichier du rapport de fournisseur.
4. Sortie du cas d'utilisation.

**Scénario alternatif :**

- 1a. Le fournisseur n'est pas dans la liste des fournisseurs.
  - 1a.1 Le système avertit que le numéro de fournisseur n'est pas valide.
  - 1a.1 Sortie du cas d'utilisation.

Le scénario alternatif concerne le gérant *ChocAn* qui pourrait entrer un numéro de fournisseur invalide.

Dans le cas de l'horloge qui génère les rapports à chaque semaine, comme elle se base sur les fournisseurs déjà dans la liste de service, il est logiquement impossible d'arriver dans ce scénario alternatif car le fournisseur doit exister.

Si la liste est vide au moment de générer le rapport, le rapport est tout de même produit, mais ne contient que le minimum.

**\*\*Voir le glossaire pour le contenu du rapport.**



## **Cas d'utilisation : générer un rapport de membre**

**But :** Produire un rapport de membre.

**Préconditions :** Le système a reçu la demande de générer un rapport de membre avec un numéro de membre.

**Acteurs :** Gérant *ChocAn* (ou Horloge)

### **Scénario principal :**

1. Le système trouve les informations associées au numéro de membre dans la liste des membres.
2. Le système trouve tous les services reçus par le membre dans la liste de service.
3. Le système génère le fichier du rapport de membre.
4. Sortie du cas d'utilisation.

### **Scénario alternatif :**

- 1a. Le membre n'est pas dans la liste des membres.
  - 1a.1 Le système avertie que le numéro de membre n'est pas valide.
  - 1a.1 Sortie du cas d'utilisation.

Le scénario alternatif concerne le gérant *ChocAn* qui pourrait entrer un numéro de membre invalide. Dans le cas de l'horloge qui génère les rapports à chaque semaine, comme elle se base sur les membres déjà dans la liste de service, il est logiquement impossible d'arriver dans ce scénario alternatif car le membre doit exister.

Si la liste est vide au moment de générer le rapport, le rapport est tout de même produit, mais ne contient que le minimum.

**\*\*Voir le glossaire pour le contenu du rapport.**

## **Cas d'utilisation : générer un rapport de synthèse**

**But :** Produire un rapport de synthèse.

**Préconditions :** Le système a reçu la demande de générer un rapport de synthèse.

**Acteurs :** Gérant *ChocAn* (ou Horloge)

### **Scénario principal :**

1. Le système compile les informations de la liste de service.
3. Le système génère le fichier du rapport de synthèse.
3. Sortie du cas d'utilisation.

Si la liste est vide au moment de générer le rapport, le rapport est tout de même produit, mais ne contient que le minimum.

**\*\*Voir le glossaire pour le contenu du rapport.**

## **Cas d'utilisation : Supprimer Membre**

**But :** Supprimer un membre. Différent de suspendre un membre. Dans ce cas-ci, on veut supprimer complètement un membre, c'est-à-dire le retirer complètement de la liste de membre et effacer ses informations du Centre de données *ChocAn*.

**Préconditions :** Logiciel en mode interactif.

**Acteurs :** Opérateur *ChocAn*

**Scénario principal :**

1. Le système reçoit le numéro du membre à supprimer.
2. Le système trouve le numéro du membre dans la liste de membre.
  - 2.1 Le système retourne le nom du membre et demande s'il s'agit bien du membre que l'opérateur désire supprimer. (Oui / Non)
    - 2.1.1 Dans le cas oui : Continuer à 3.
    - 2.1.2 Dans le cas non : Continuer à 5.
3. Le système demande de confirmer si l'on veut supprimer définitivement le membre (Oui / Non)
  - 3.1 Dans le cas oui : Continuer à 4.
  - 3.2 Dans le cas non : Continuer à 5.
4. Le système supprime le membre de la liste de membre du centre de données.
5. Sortie du cas d'utilisation.

**Scénario alternatif :**

- 2a. Le numéro de membre ne se trouve pas dans la liste de membre.
  - 2a.1 Le système avertit que le numéro de membre est invalide.
  - 2a.2 Sortie du cas d'utilisation

**Cas d'utilisation : Ajouter Membre**

**But :** Ouvrir un compte pour une personne qui souhaite devenir membre *ChocAn*.

**Préconditions :** Un nouveau client se présente chez *ChocAn* et un opérateur lui ouvre un compte. Le système est en mode interactif.

**Postconditions :** Le nouveau membre est ajouté à la liste de membre.

**Acteurs :** Opérateur *ChocAn*

**Scénario principal :**

1. Le système reçoit une requête pour ajouter un membre.
2. Le système crée un nouveau membre.
  - 2.1 Le système assigne un numéro de membre disponible.
  - 2.2 Le membre est créé avec des champs d'informations vides.
  - 2.3 Le membre est enregistré dans la liste de membre.
3. Accès à *CU Mettre à Jour Compte* avec le numéro de membre tout juste créé.
4. Retour à l'état initial.

**Scénario alternatif :**

À tout moment durant le cas d'utilisation, l'ouverture de compte peut être annulée par l'opérateur. Dans un tel cas, le système arrête la procédure d'ajout de membre et le *CU Supprimer Membre* est appelé avec le numéro de membre qui vient d'être créé.

Une carte est fournie initialisée avec le numéro de membre est fournie au nouveau membre. Comme les frais d'adhésion sont gérés par *Acme*, on considère que les paiements se font directement à eux. Par exemple, un prélèvement pourrait être fait sur la carte de crédit du membre à chaque mois et *Acme* s'assure de mettre à jour le statut du membre selon les paiements reçus.

## **Cas d'utilisation : Mettre à Jour Compte**

**But :** Modifier les informations du compte d'un membre ou fournisseur.

**Préconditions :** À l'ouverture d'un compte. Si les informations courantes sont erronées et doivent être modifiées. Normalement suite à la demande du membre ou fournisseur.

**Acteurs :** Opérateur *ChocAn*

### **Scénario principal :**

1. Le système reçoit une requête pour modifier des informations et un numéro.
2. C'est un numéro de membre ou de fournisseur :
  - 2.1 Dans le cas d'un numéro de fournisseur :
    - 2.1.1 Le système trouve le fournisseur dans la liste de fournisseur.
    - 2.1.2 Continuer à 3.
  - 2.2 Dans le cas d'un numéro de membre:
    - 2.2.1 Le système trouve le membre dans la liste de membre.
    - 2.2.2 Continuer à 3.
3. Le système demande quel champ du compte l'opérateur souhaite modifier en rappelant les différents champs d'informations.
4. L'opérateur choisi le champ à modifier.
5. Le système demande la nouvelle valeur.
6. La nouvelle valeur est valide.
  - 6.1 Le système enregistre la nouvelle valeur. (Écrase l'ancienne)
7. Le système demande si l'opérateur souhaite modifier un autre champ.
  - 7.1 Dans le cas oui :
    - 7.1.1 Retour à 3.
  - 7.2 Dans le cas non :
    - 7.2.1 Continuer à 8.
8. Le système vérifie qu'il n'y a pas de champs vide.
9. Retour à l'état initial

### **Scénario alternatif :**

- 2a. Le numéro ne se trouve pas dans la liste de membre/fournisseur.
  - 2a.1 Le système avertie que le numéro est invalide.
  - 2a.2 Sortie du cas d'utilisation
- 6a. La valeur entrée n'est pas valide.
  - 6a.1 Le système avertie que la valeur n'est pas valide pour le champ et rappelle le format que doit avoir la valeur.
  - 6a.2 Retour à 5.
- 8a. Il y a des champs obligatoires vide.
  - 8a.1 Le système indique les champs qui doivent être remplis.
  - 8a.2 Retour à 3.

À tout moment l'opérateur peut vouloir quitter le cas d'utilisation. Dans ce cas, il va directement à l'étape 8 qui le laissera sortir seulement si les champs sont bien remplis.

Comme le compte d'un membre et le compte d'un fournisseur sont très semblables, nous avons décidé de généraliser le cas d'utilisation de mise à jour. Éventuellement, on peut imaginer qu'une même fonction pourrait modifier les champs de deux classes qui implémentent « compte ».

## **Cas d'utilisation : Supprimer Fournisseur**

**But :** Supprimer un fournisseur.

**Préconditions :** Logiciel en mode interactif. Le fournisseur ne veut plus faire affaire avec *ChocAn* ou vice-versa. On assume que les comptes sont réglés, c'est-à-dire que *ChocAn* ne doit rien au fournisseur en question.

**Acteurs :** Opérateur *ChocAn*

### **Scénario principal :**

1. Le système reçoit le numéro du fournisseur à supprimer.
2. Le système trouve le numéro du fournisseur dans la liste de fournisseur.
  - 2.1 Le système retourne le nom du fournisseur et demande s'il s'agit bien du fournisseur que l'opérateur désire supprimer. (Oui / Non)
    - 2.1.1 Dans le cas oui : Continuer à 3.
    - 2.1.2 Dans le cas non : Continuer à 5.
3. Le système demande de confirmer si l'on veut supprimer définitivement le fournisseur (Oui / Non)
  - 3.1 Dans le cas oui : Continuer à 4.
  - 3.2 Dans le cas non : Continuer à 5.
4. Le système supprime le fournisseur de la liste de fournisseur du centre de données.
5. Sortie du cas d'utilisation.

### **Scénario alternatif :**

- 2a. Le numéro de fournisseur ne se trouve pas dans la liste de fournisseur.
  - 2a.1 Le système avertit que le numéro de fournisseur est invalide.
  - 2a.2 Sortie du cas d'utilisation.

## **Cas d'utilisation : Ajouter Fournisseur**

**But :** Ouvrir un compte pour un spécialiste de la santé voulant fournir ses services aux membres *ChocAn*.

**Préconditions :** Un spécialiste de la santé se présente chez *ChocAn* et un opérateur lui ouvre un compte. Le système est en mode interactif.

**Postconditions :** Le nouveau fournisseur est ajouté à la liste de fournisseur.

**Acteurs :** Opérateur *ChocAn*

### **Scénario principal :**

1. Le système reçoit une requête pour ajouter un fournisseur.
2. Le système crée un nouveau membre.
  - 2.1 Le système assigne un numéro de fournisseur disponible.
  - 2.2 Le fournisseur est créé avec des champs d'informations vides.
  - 2.3 Le fournisseur est enregistré dans la liste de fournisseur.
3. Accès à *CU Mettre à Jour Compte* avec le numéro de fournisseur tout juste créé.
4. Retour à l'état initial.

**Scénario alternatif :**

À tout moment durant le cas d'utilisation, l'ouverture de compte peut être annulé par l'opérateur. Dans un tel cas, le système arrête la procédure d'ajout de fournisseur et le *CU Supprimer fournisseur* est appelé avec le numéro de fournisseur qui vient d'être créé.

**Cas d'utilisation : Mise à jour Statut Membre**

**But :** Mettre à jour des statuts de membre par rapport au frais d'adhésion. Le statut peut être mis à actif ou suspendu. Cela dépend si le membre a fait son paiement ou pas (géré du côté d'*Acme*).

**Préconditions :** Chaque soir à 21h.

**Acteurs :** Ordinateur *Acme*

Il y a un acteur horloge dans ce cas-ci, mais il est du côté de l'ordinateur d'*Acme*. Donc à 21h l'ordinateur d'*Acme* utilise le logiciel du centre de données pour faire ses modifications au statu des membres. Un membre qui est suspendu est toujours dans le système, mais ne peut pas avoir accès aux services. Le statut du membre est un champ dans le compte du membre. Comme on veut qu'*Acme* ne puisse changer que le champ statut, on passe par un cas d'utilisation isolé au lieu de *CU Mettre à Jour Compte*.

**Scénario principal :**

1. Le système reçoit une requête pour modifier le statut d'un membre et un numéro de membre.
2. Le système trouve le membre dans la liste de membre.
3. Le système avertie du statut courant du membre et demande s'il doit être modifié.
4. L'ordinateur d'*Acme* indique si le statut doit être changé.
  - 4.1 Dans le cas oui :
    - 4.1.1 Le système change le statut.
    - 4.1.2 Continuer à 5.
  - 4.2 Dans le cas non :
    - 4.2.1 Continuer à 5.
5. Sortie du cas d'utilisation.

**Scénario alternatif :**

- 2a. Le numéro de membre ne se trouve pas dans la liste de membre.
  - 2a.1 Le système avertie que le numéro de membre est invalide.
  - 2a.2 Sortie du cas d'utilisation

## Risques

- Risque Sévère : On n'a rien trouvé qui puisse compromettre le fonctionnement du logiciel.
- Risque Important :
  - La structure de donnée choisie pour stocker l'information dans le centre de données doit être optimale pour certaines opérations. Risque de commencer avec une mauvaise structure qui soit difficile à changer une fois l'implantation commencée.
  - On doit sécuriser les informations de membre et de fournisseurs pour éviter la fraude.
  - Risque sur la performance du logiciel si trop d'acteurs accèdent au centre de données en même temps. Le logiciel doit performer assez rapidement pour qu'il n'y ait pas trop de trafic. On doit gérer le trafic si différent CU accède aux mêmes données en même temps.
- Risque Éloigné :
  - Possibilité de pouvoir retourner en arrière si un opérateur a commis une erreur : ex. supprimer le mauvais membre. Il se pourrait bien que ChocAn demande cette fonction. Prévoir à l'avance.
- Obstacle Mineur :
  - Si ChocAn veut grossir ses opérations et ouvrir ses services ailleurs dans le monde, le format du code postal pourrait changer. Il faut que le format soit facilement adaptable.

## Exigences Non-Fonctionnelles

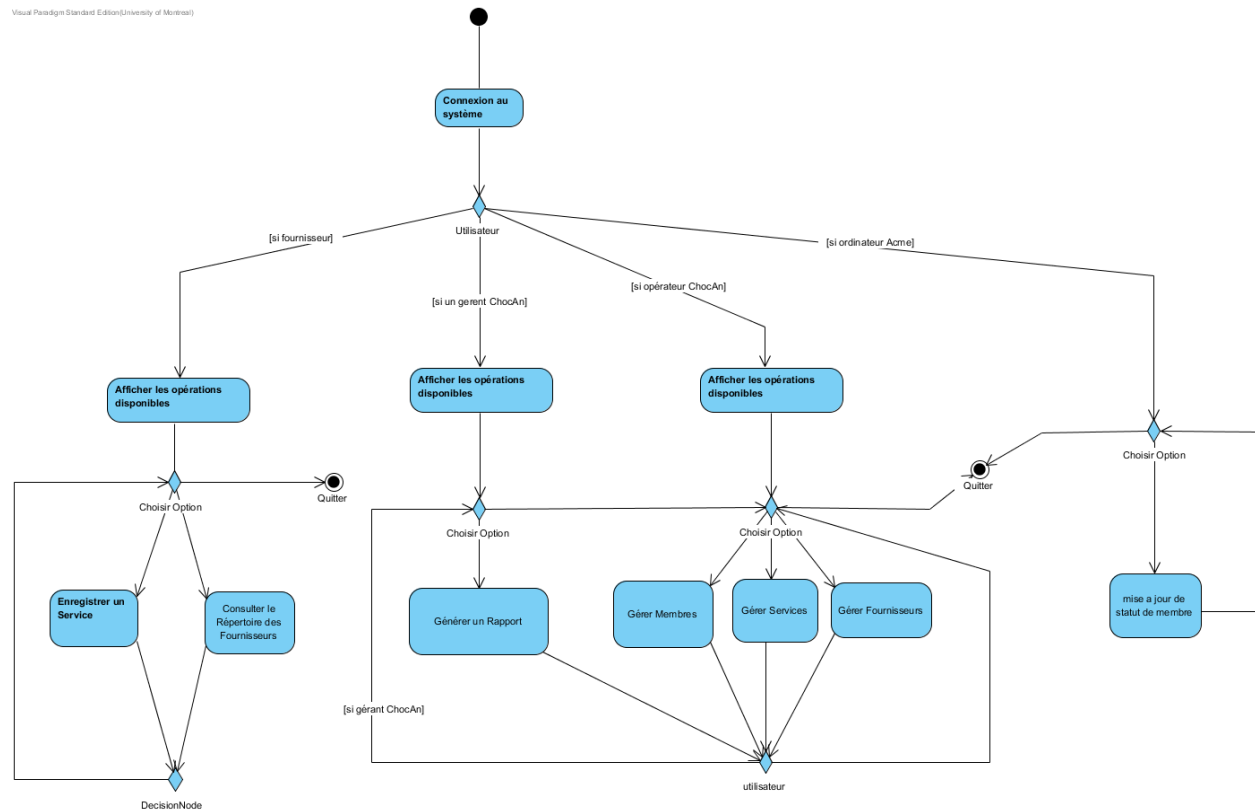
- La nomenclature des fichiers générés. Le nom d'un rapport de membre doit commencer par le nom du membre suivi de la date du rapport.
- L'information contenue dans le rapport du fournisseur, doit être dans le même ordre que dans le formulaire du fournisseur afin de faciliter la vérification.
- Distinguer les numéros de membres des numéros de fournisseurs. Les deux numéros sont composés de 9 chiffres. Peut-être qu'on pourrait utiliser le premier chiffre pour faire la distinction.
- Les interactions entre le terminal du fournisseur et le centre de données doivent se faire dans un temps raisonnable. (Demander à ChocAn le délais raisonnable)
- Le système doit être opérationnel chaque soir à 21h et les vendredis à minuit, pour les événements temporels.

Si on se fie au format du code postal, le logiciel sera implémenté aux États-Unis. Un besoin non-fonctionnelle serait que le logiciel soit implanté en anglais et espagnol.

# WORKFLOW ANALYSE ET CONCEPTION

## Diagramme d'activité UML

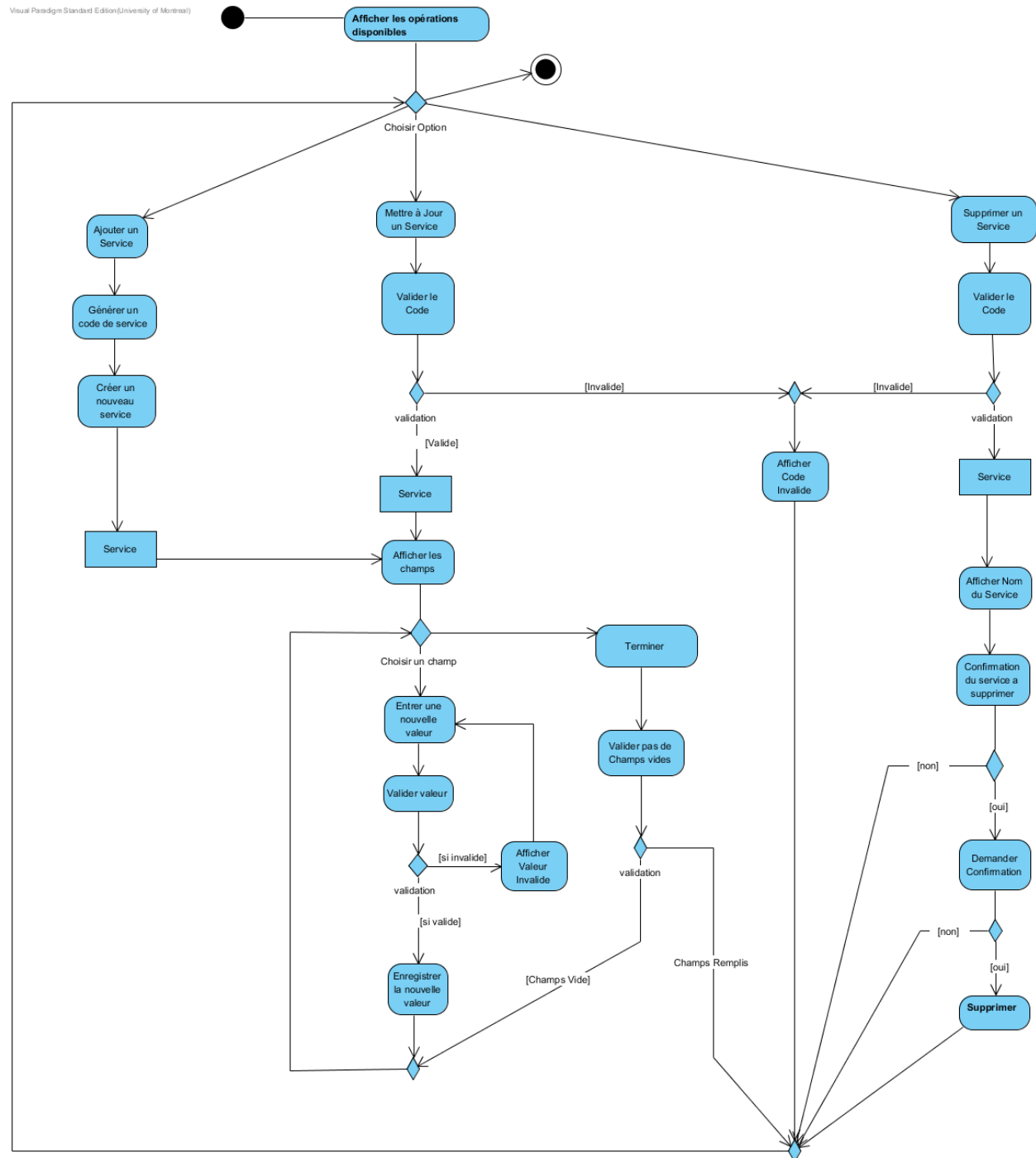
### Diagramme Principal



[Activity Diagram - ChocAn.png](#)

Une fois connecté au système, il y a plusieurs possibles activités selon l'acteur impliqué.

## Diagramme Gérer les Services



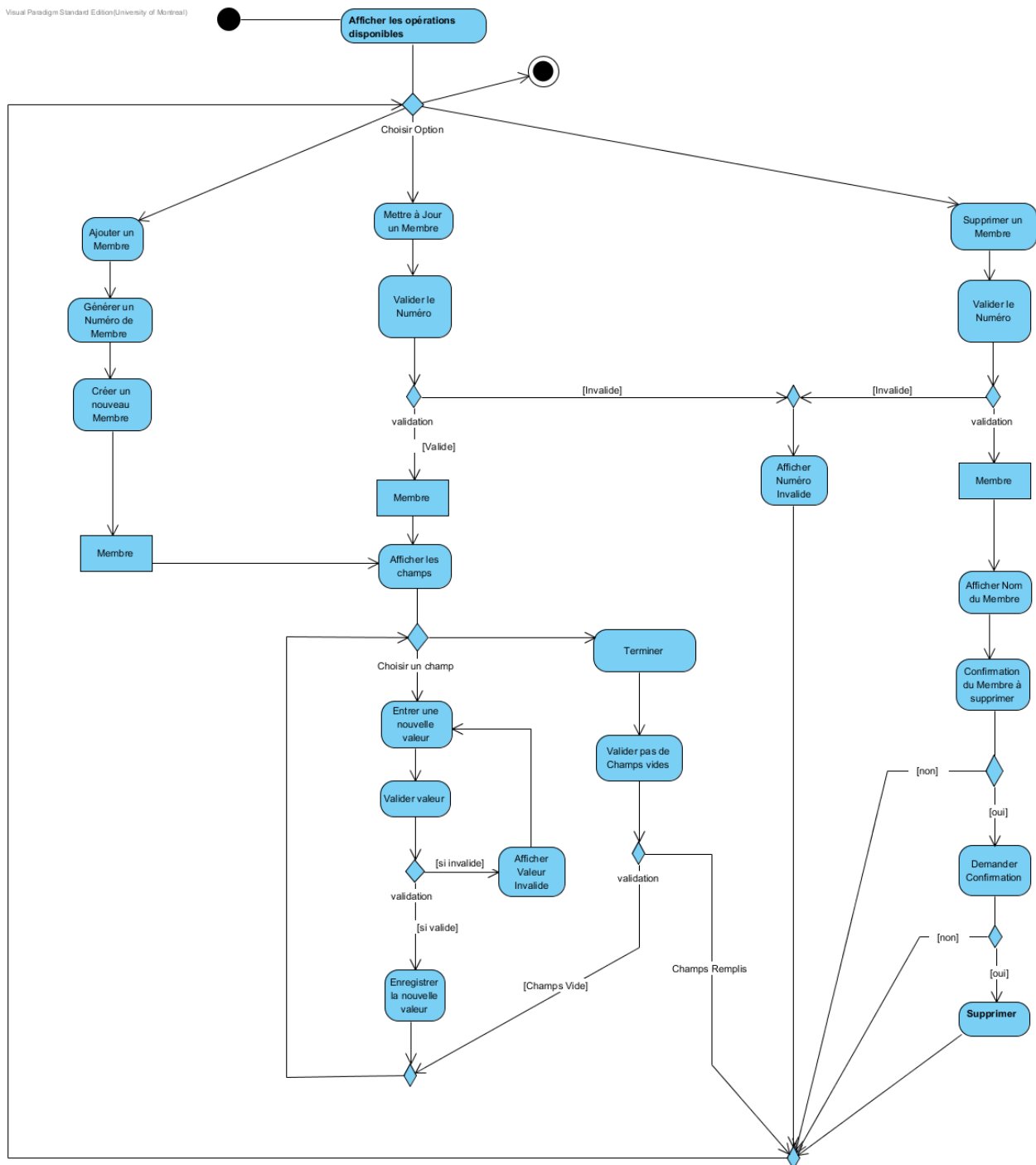
Activity Diagram - Gérer Services.png

Les champs qui peuvent être modifiés dans un services sont : code, frais et nom. Il faut vérifier que la nouvelle valeur entrée dans un champs est valide selon le champ avant de modifié le service. On ne peut pas sortir de l'activité Mise à jour d'un service tant que les champs ne sont pas tous remplis.



## Diagramme Gérer les Membres

Visual Paradigm Standard Edition (University of Montreal)

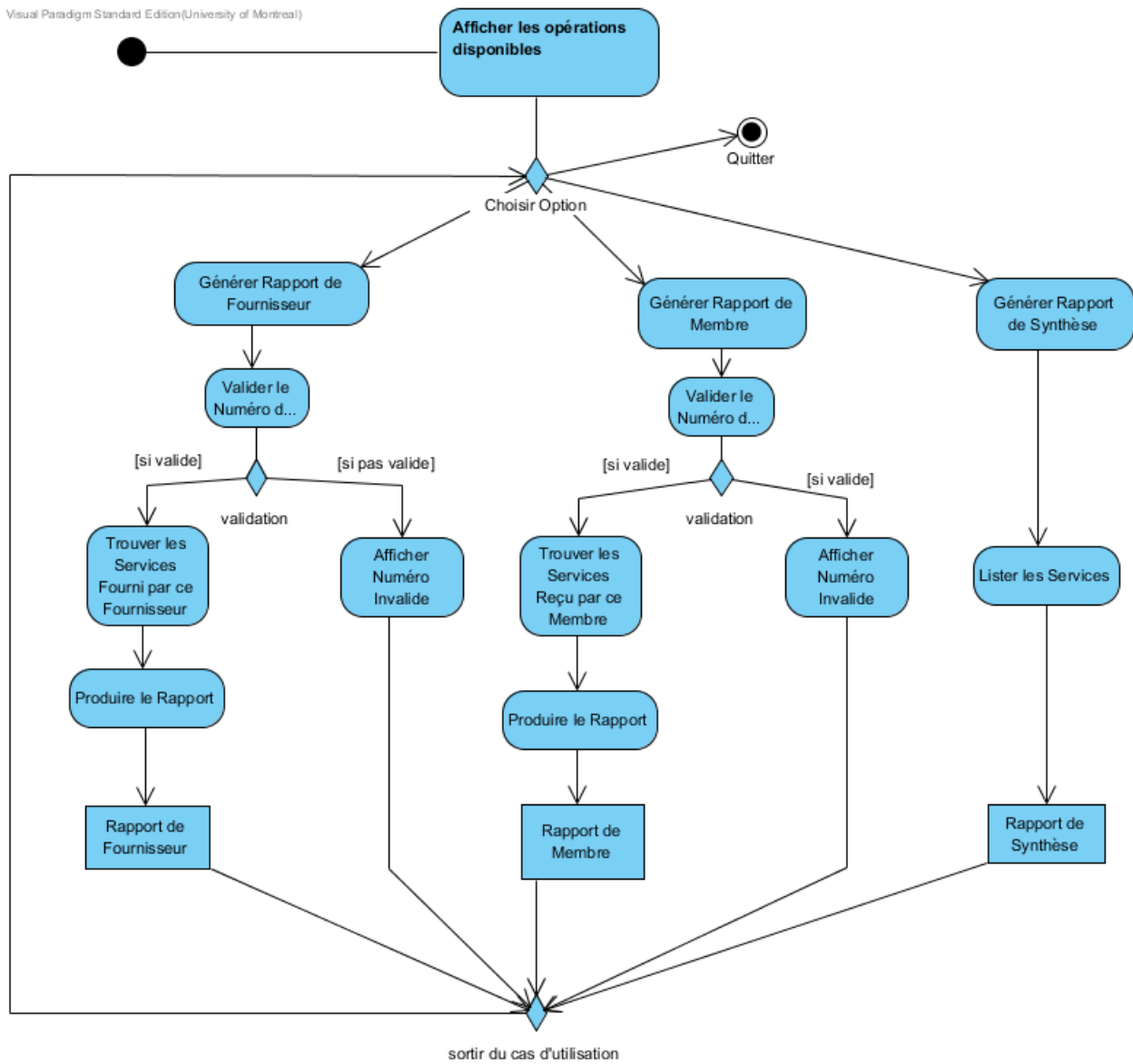


Activity Diagram - Gérer Membres.png

Les champs d'un membre qui peuvent être modifiés sont : code, frais et nom. Il faut vérifier que la nouvelle valeur entrée dans un champs est valide selon le champ avant de modifié le service. On ne peut pas sortir de l'activité Mise à jour d'un service tant que les champs ne sont pas tous remplis.

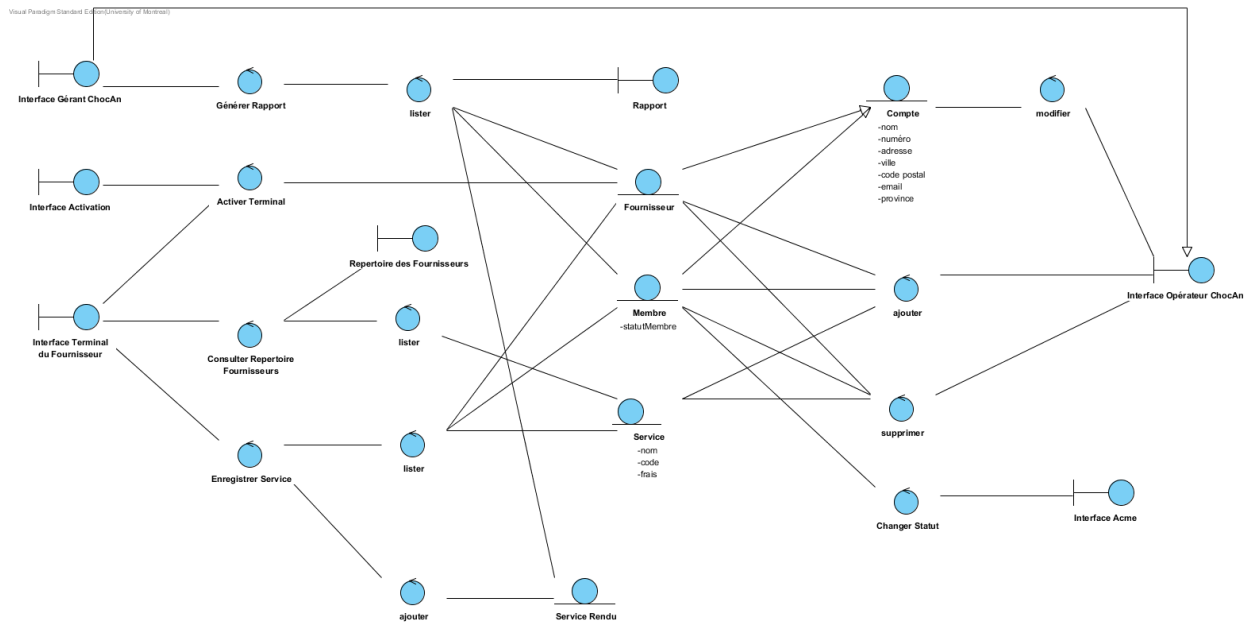
## Diagramme Produire un Rapport

Visual Paradigm Standard Edition (University of Montreal)



[Activity Diagram - Produire Rapport.png](#)

## Diagramme de Classe Participante



[Diagramme Classe Participante.png](#)

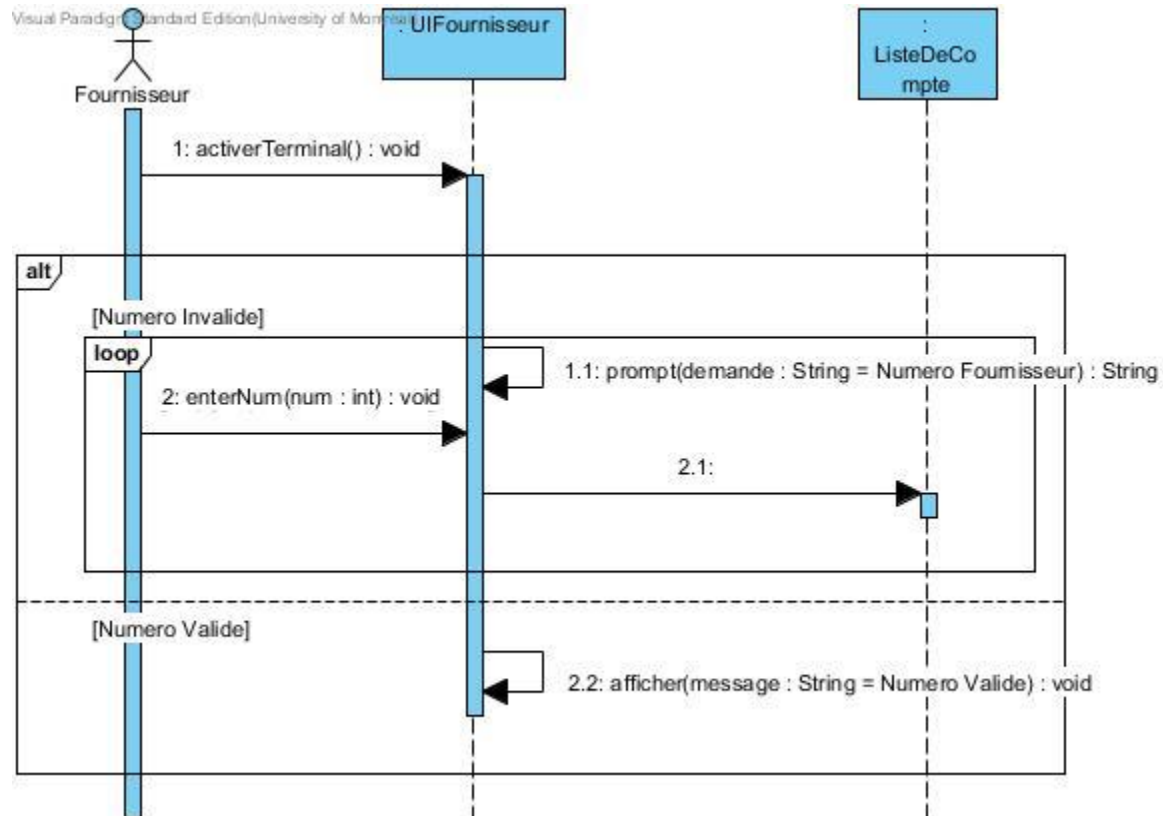
Comme notre diagramme de classe est plus proche de l'implémentation, nous avons décidé de garder le diagramme de classe participante plus simple afin d'avoir une vision simplifiée des interactions entre les classes. Dans le diagramme de classe il y a des classes de listes qui, au final, servent surtout à gérer/contrôler les entités membres, services, fournisseur et service rendu au travers de différentes méthodes. Nous ne les avons donc pas inclus ici.

Source: *University of Tennessee & Southern Baptist Convention*. (n.d.). Retrieved from <http://www.southwestern.edu>



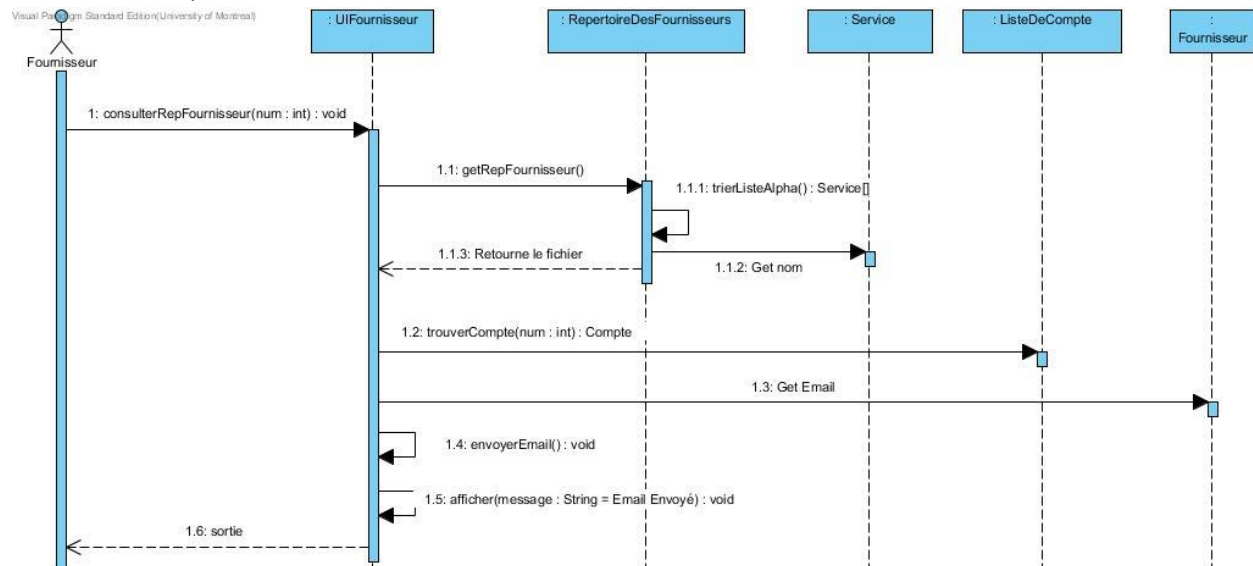
## Diagrammes de Séquence

### Activer Terminal



[Sequence Diagram - Activer Terminal.png](#)

### Consulter Répertoire des Fournisseurs



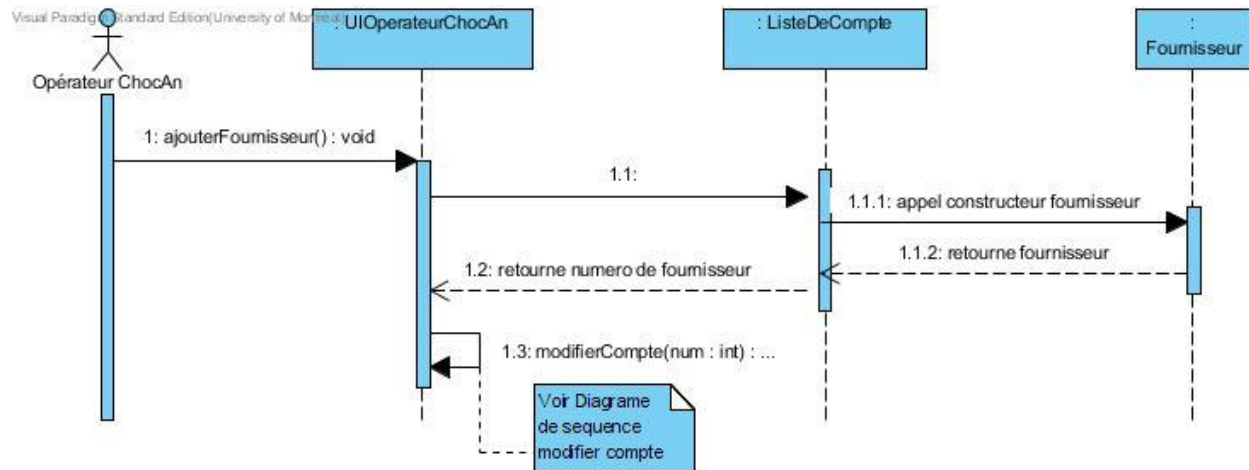
[Sequence Diagram - Consulter Répertoire des Fournisseurs.png](#)

Visual Programming Standard Edition (University of Montreal)



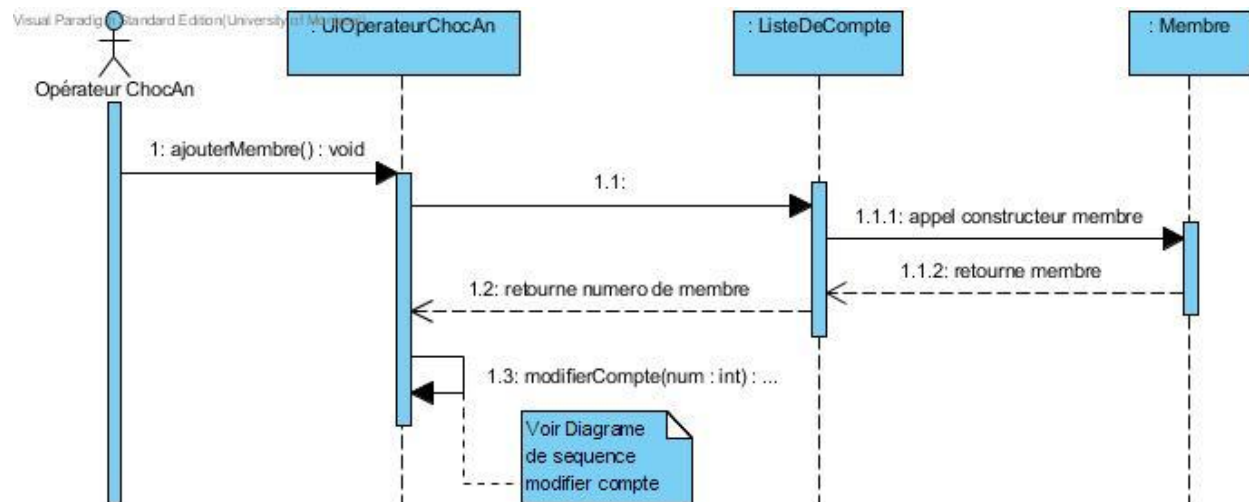
[Sequence Diagram - Enregistrer Service.png](#)

## Ajouter Fournisseur



[Sequence Diagram - Ajouter Fournisseur.png](#)

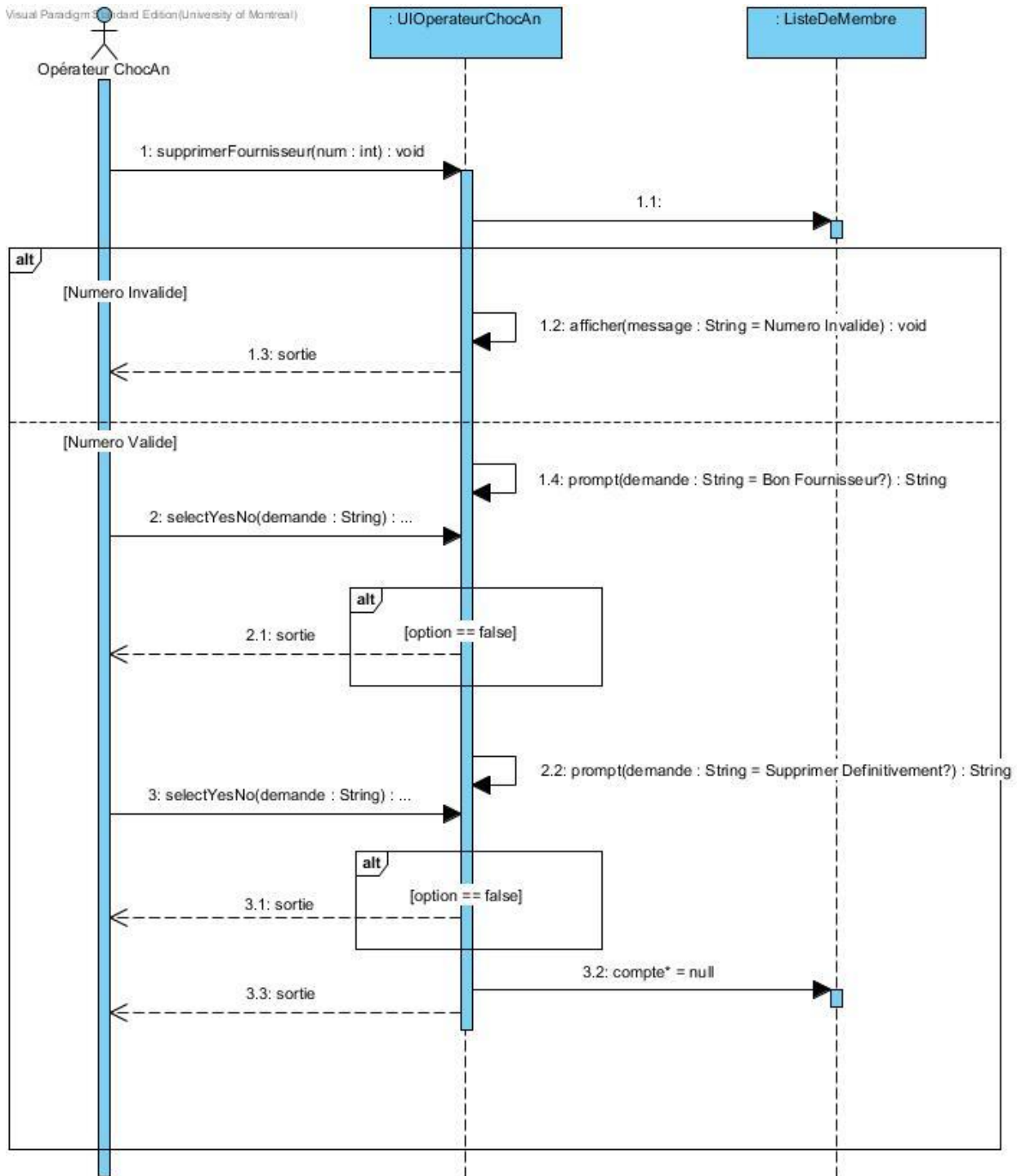
## Ajouter Membre



[Sequence Diagram - Ajouter Membre.png](#)

## Supprimer Fournisseur

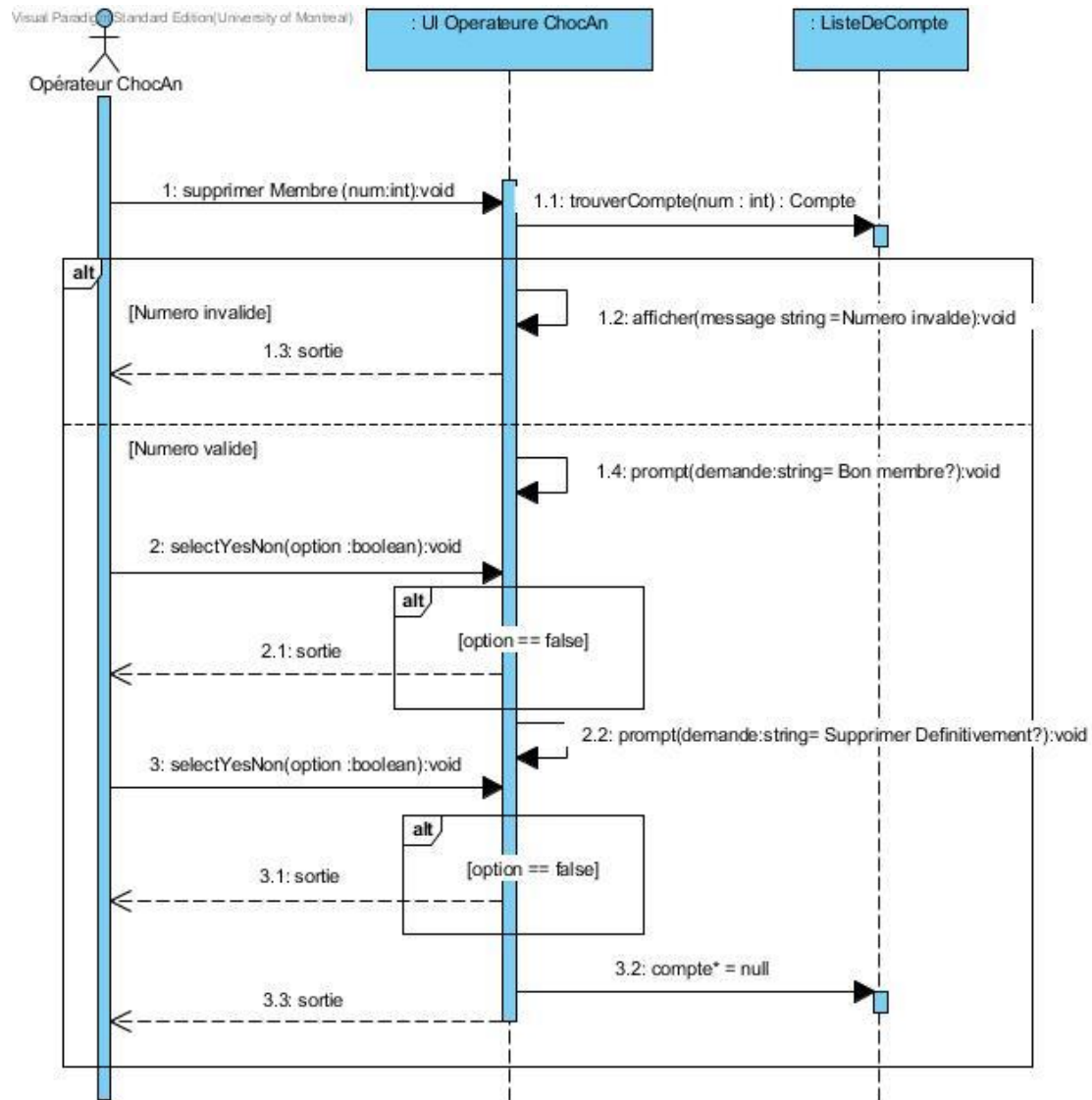
Visual Paradigm Standard Edition (University of Montreal)



[Sequence Diagram - Supprimer Fournisseur.png](#)

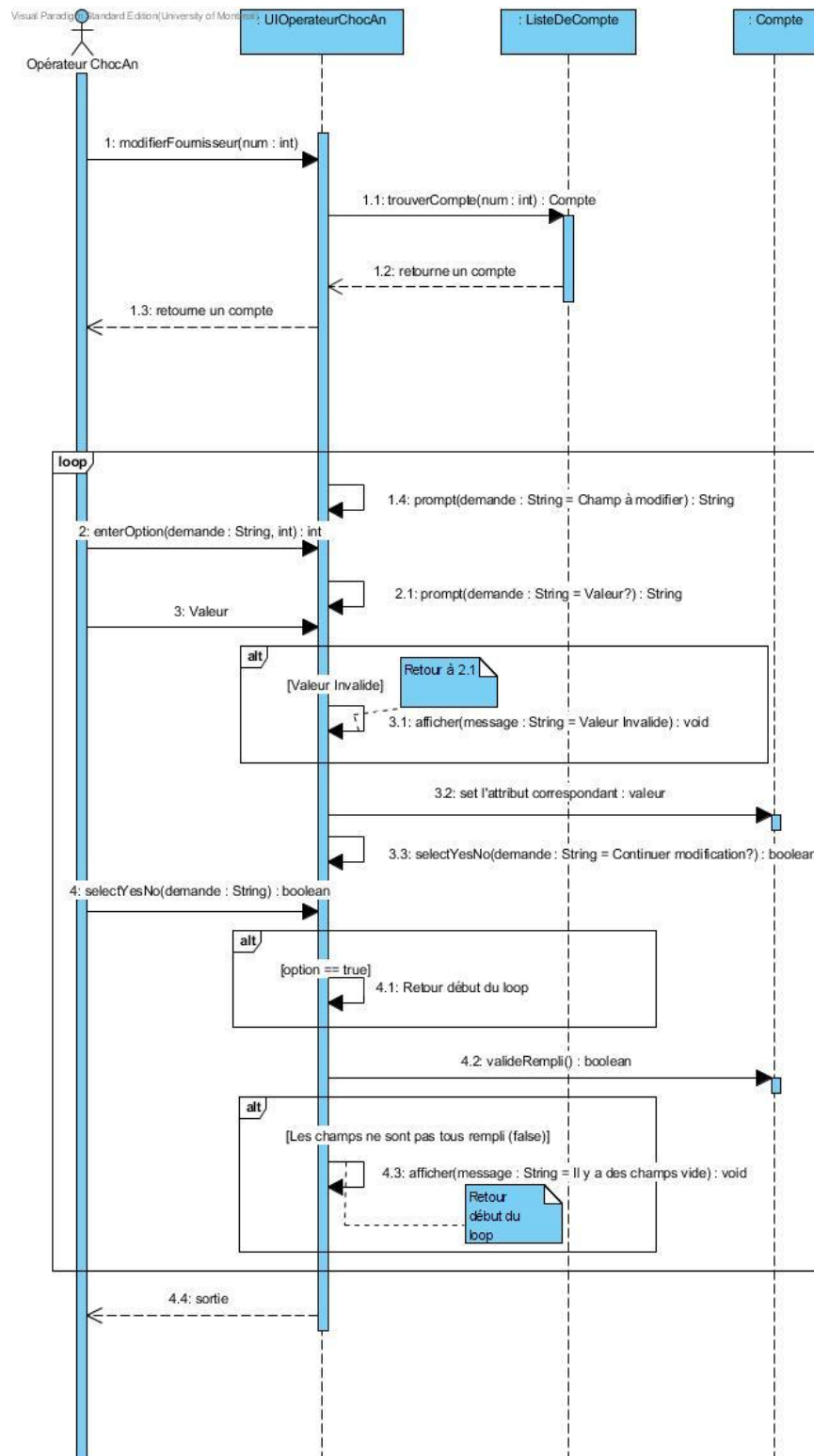


## Supprimer Membre



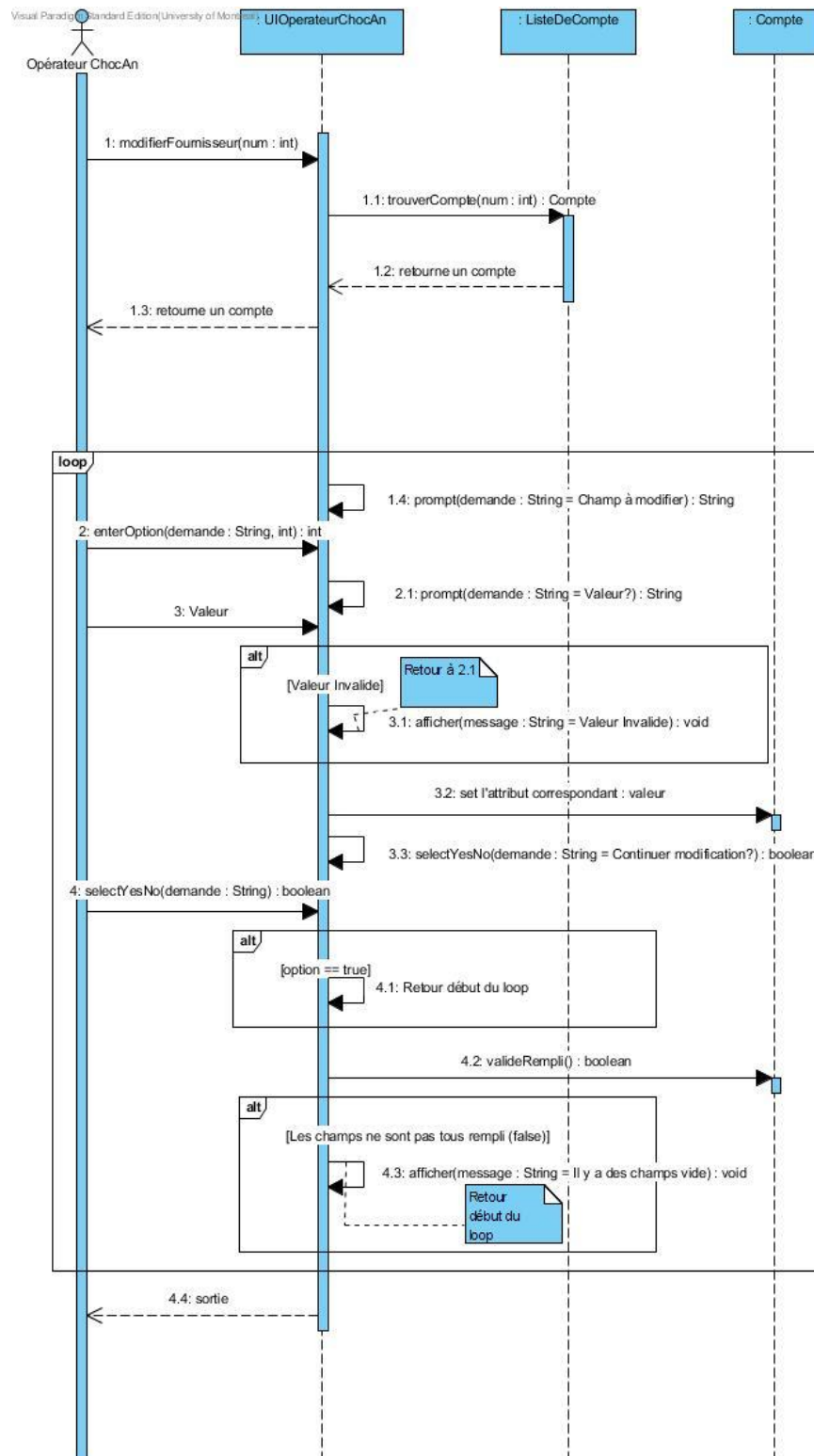
[Sequence Diagram - Supprimer Membre.png](#)

## Modifier Fournisseur



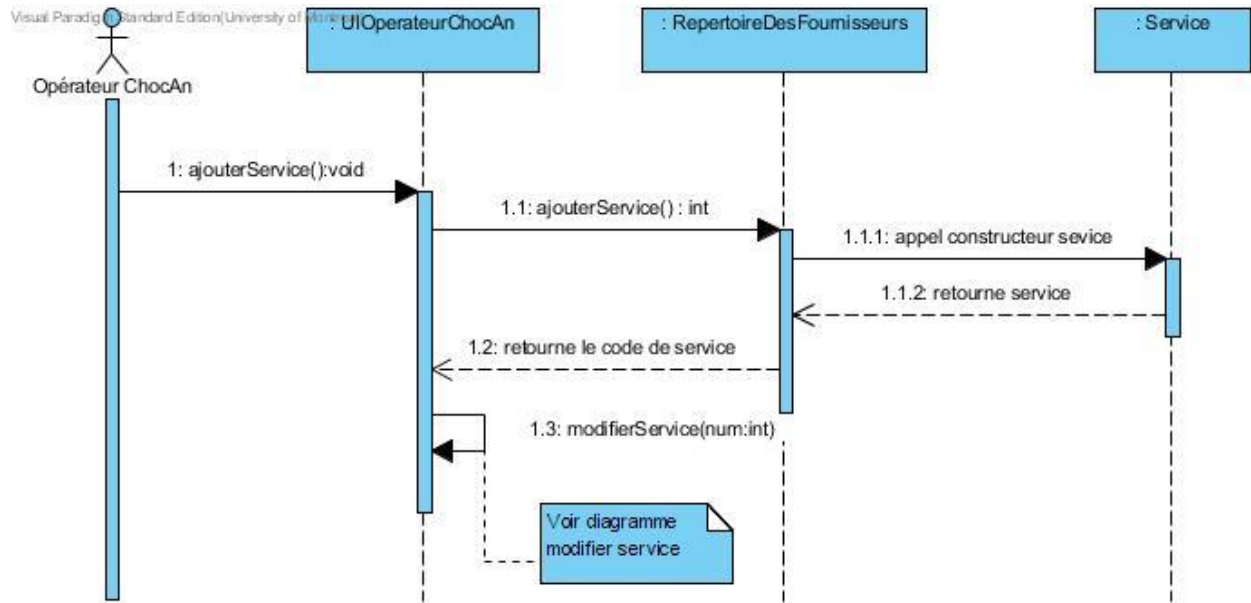
Sequence Diagram - Modifier Fournisseur.jpg

## Modifier Membre



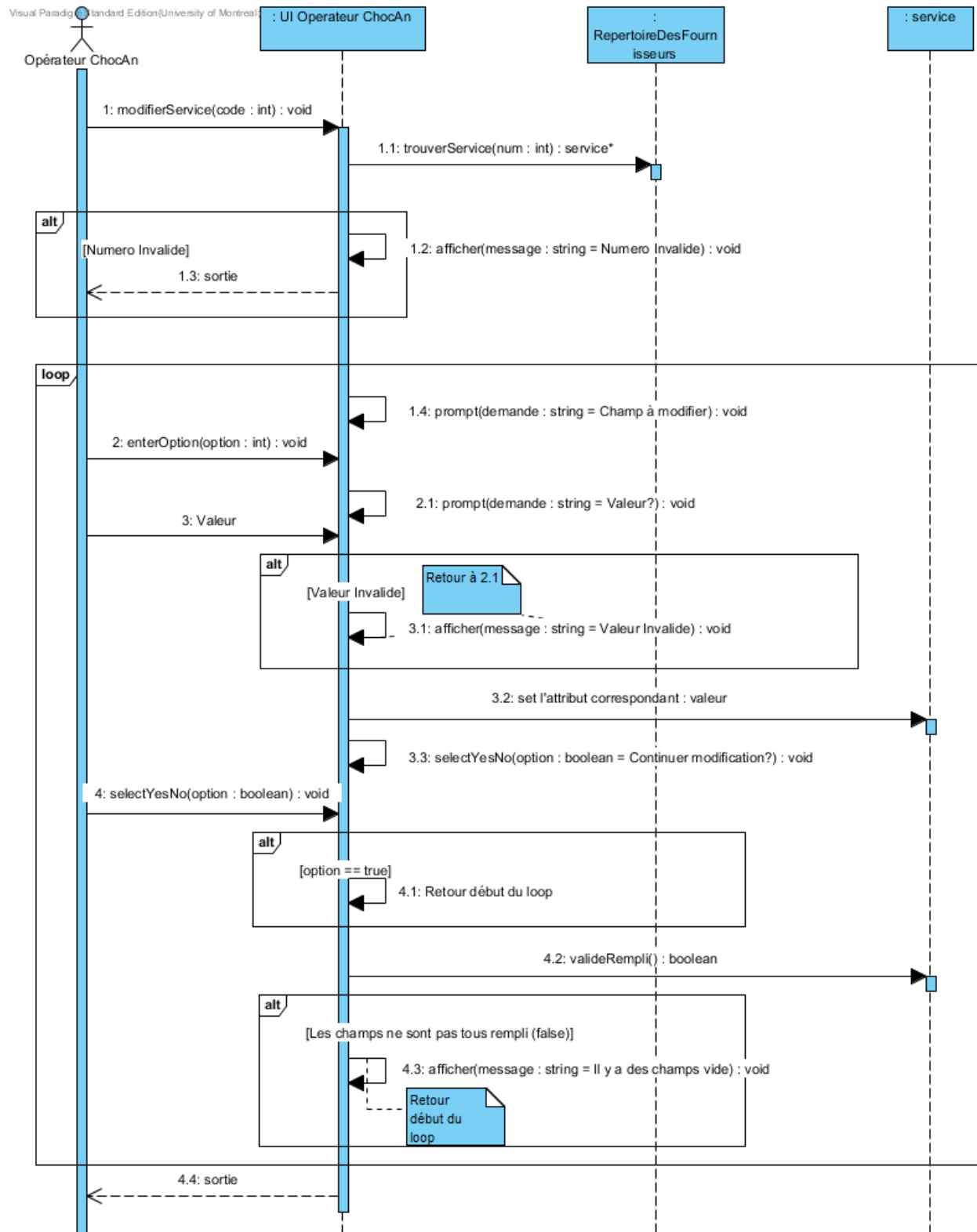
Sequence Diagram – Modifier Membre.jpg

## Ajouter Service



[Sequence Diagram - Ajouter Service.png](#)

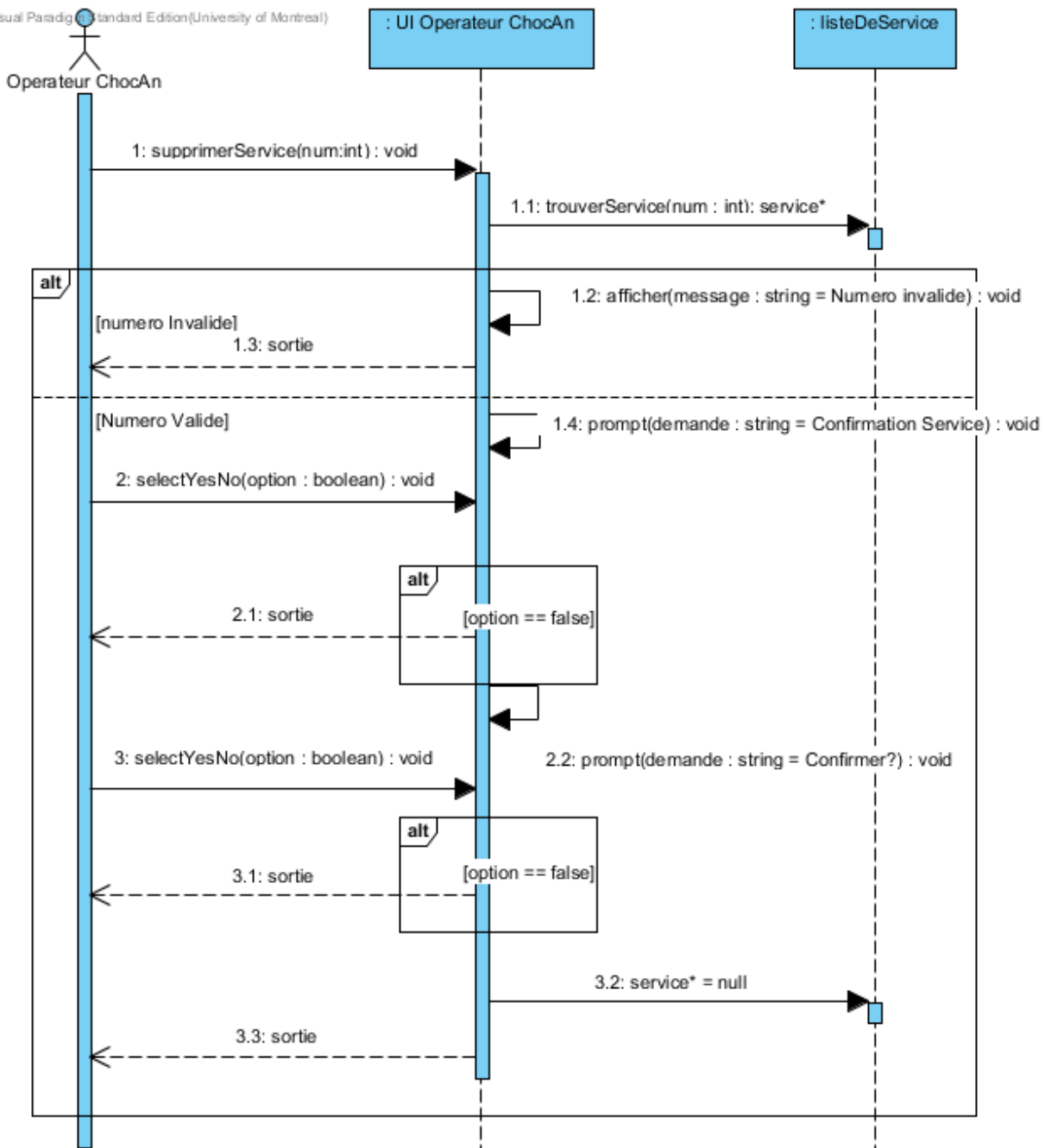
## Modifier Service



Sequence Diagram - Modifier Service.png

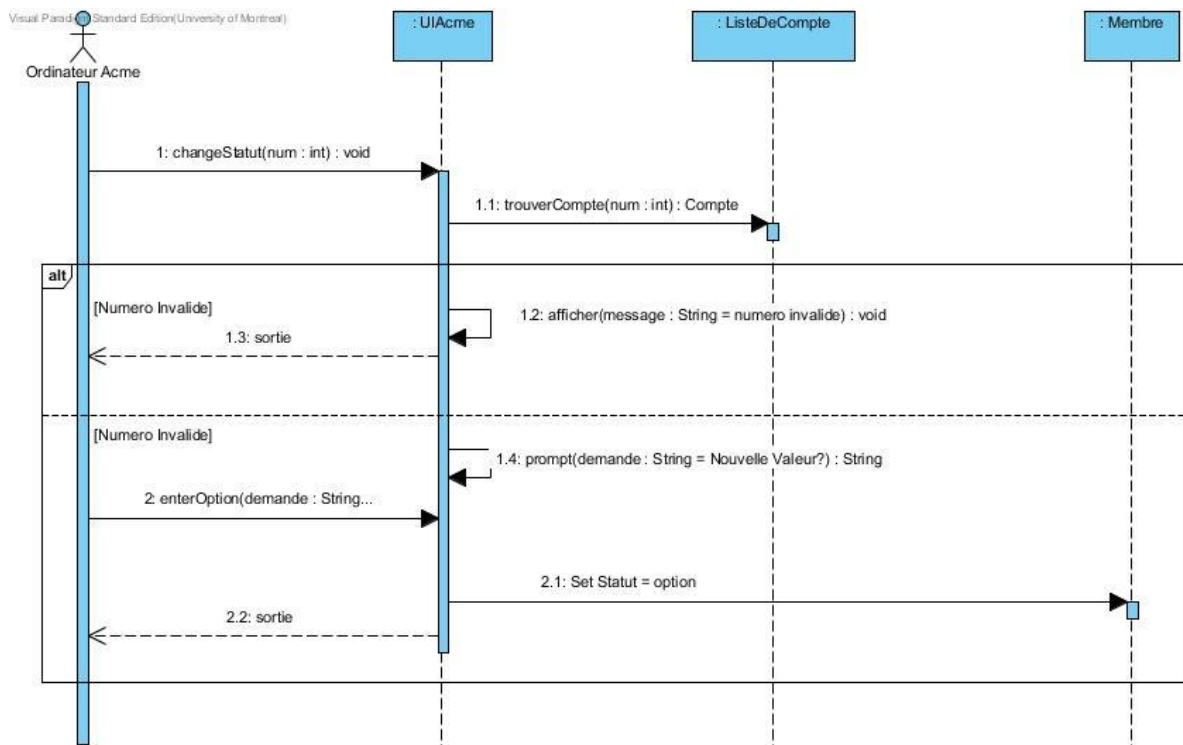
## Supprimer Service

Visual Paradigm Standard Edition (University of Montreal)



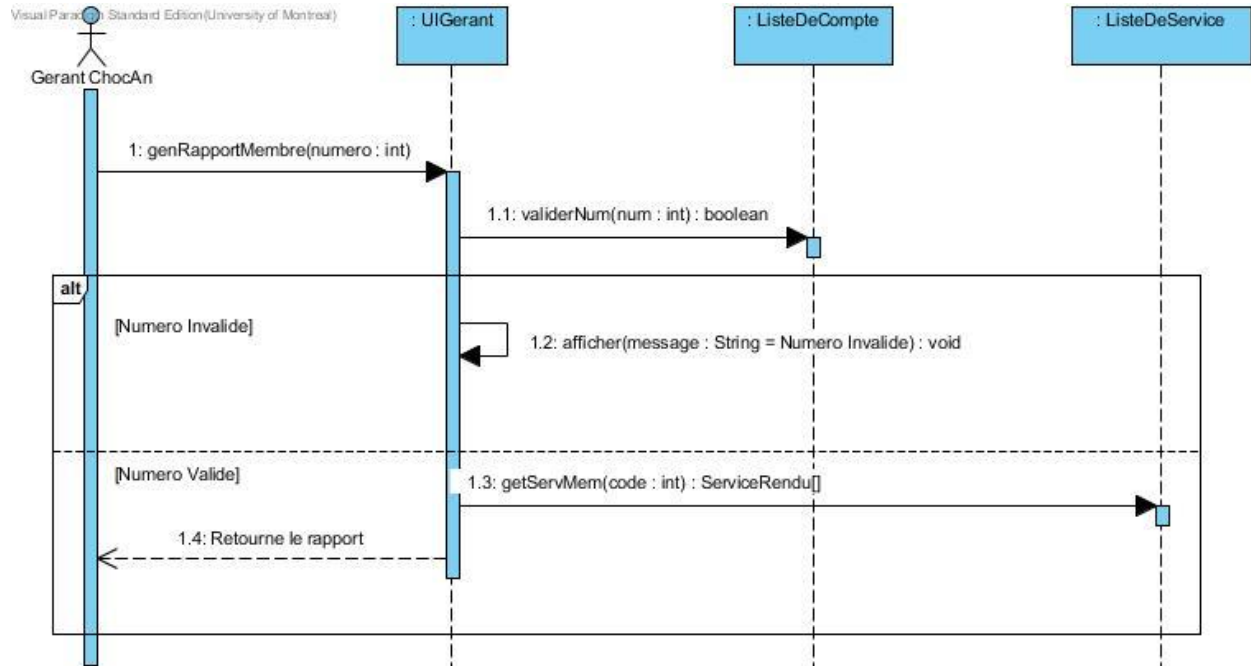
[Sequence Diagram - Supprimer Service.png](#)

## Mise À Jour Statut Membre



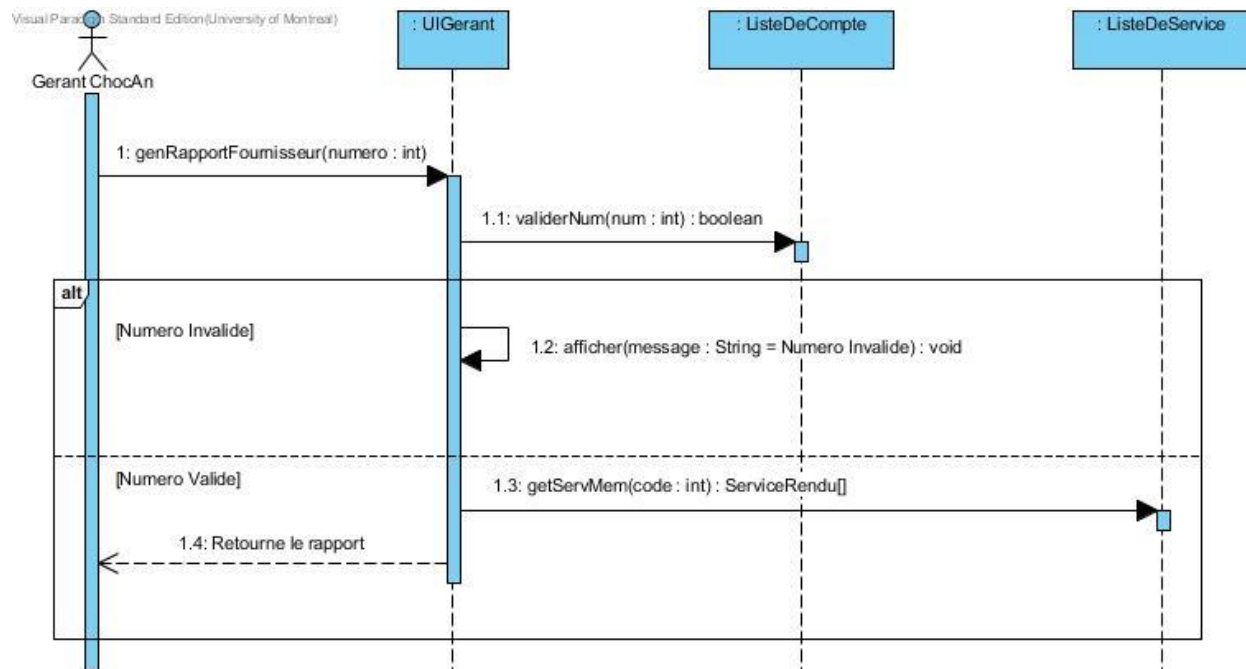
[Sequence Diagram - Mise À Jour Statut Membre.png](#)

## Générer Rapport de Membre



[Sequence Diagram - Generer Rapport Membre.png](#)

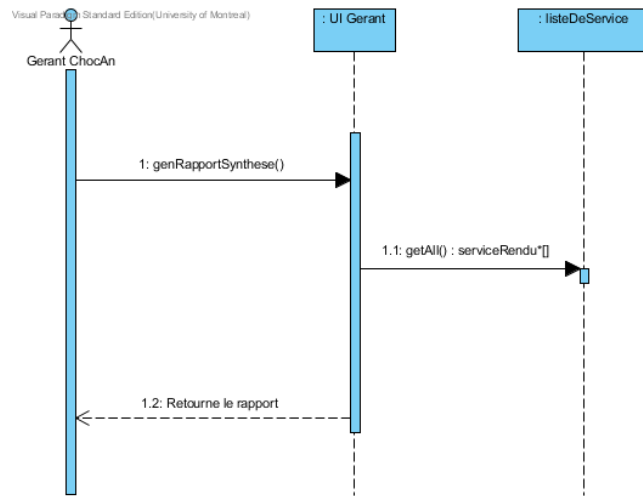
## Générer Rapport de Fournisseur



[Sequence Diagram - Générer Rapport Fournisseur.png](#)

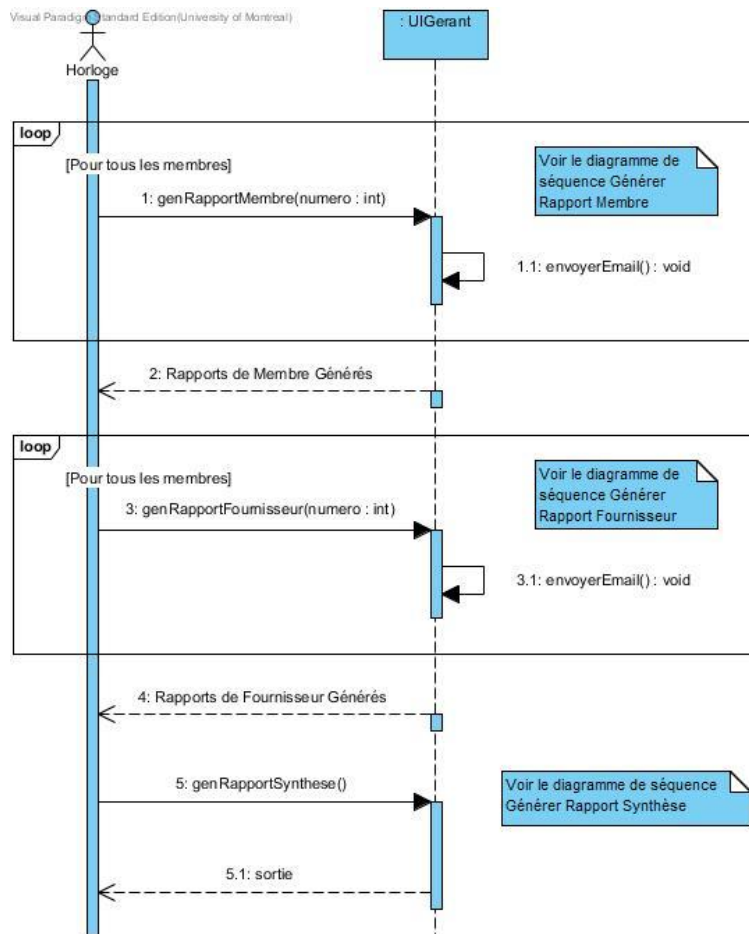


## Générer Rapport de Synthèse



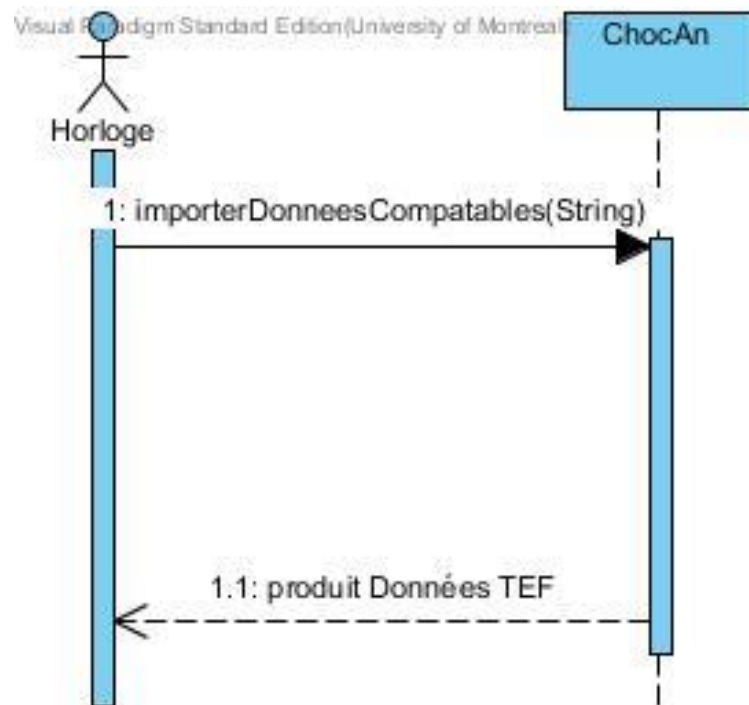
Sequence Diagram - Générer Rapport Synthèse.png

## Générer les Rapports



Sequence Diagram - Générer les Rapports.png

TEF



[Diagramme de sequence - Produire Données TEF.jpg](#)

## WORKFLOW IMPLÉMENTATION, TEST, LIVRAISON

Avant de débiter l'implémentation, nous avons dû regarder ce que chacune des équipes avait fait lors du workflow de conception et d'analyse. Nous avons décidé de nous baser sur celui de l'équipe 05, mais nous avons pris quelques éléments manquant comme générer les données TEF de celui de l'équipe 11. Ensuite nous avons décidé de diviser les packages par aspect de la façon suivante : UI, entités Compte et entités Service. Nous avons ensuite distribué les tâches en quatre : UI pour les opérateurs ChocAn, UI du terminal et classe abstraite UI, les classes touchant au Services et les classes touchant au Comptes. Bien entendu il y avait plusieurs zones grises où tout le monde a collaboré.

Pour débiter l'implémentation, nous avons d'abord essayé de suivre le plus fidèlement possible le diagramme de classe que nous avons fait durant le workflow d'Analyse et de Conception. Au fur et à mesure, nous avons réalisé que certaines classes étaient inutiles et que d'autres requéraient plus de fonctions que nous avions prévu.

Nous avons décidé de supprimer ListeDeMembre et ListeDeFournisseur qui héritaient de ListeDeCompte, et qui finalement n'apportait rien de plus que cette dernière.

Nous avons décidé dès le début de faire une classe abstraite UI de laquelle chaque UI hériterait. Cela nous a permis de faire des fonctions précises de prompt pour chaque type d'input qui pourrait être requis d'un utilisateur. Comme chaque fonction s'assure de retourner le bon type et le bon format, on pouvait écrire les opérations d'UI (cas d'utilisation) sans se soucier de vérifier les inputs de l'utilisateur à chaque fois.

## Manuel d'utilisateur ChocAn

**\*\*** Pour que le programme ne commence pas vide, au lancement nous créons quelques Membres, Fournisseurs, Services et ServiceRendu.

**\*\*\*** Nous avons des problèmes pour générer un le .jar exécutable. Pourtant le programme fonctionnait très bien dans la console Eclipse. Finalement, en dupliquant chacune des classes avec le même contenu, nous avons réussi à régler le bug... C'est pourquoi chaque classe est nommé version 2...

Guide pratique pour une utilisation efficiente du logiciel de gestion pour l'organisation ChocAn.

Le présent guide fournit les outils de référence pour maximiser l'utilisation du logiciel, notamment en spécifiant les options offertes aux personnes qui le manipulent pour qu'elles puissent en tirer tout le potentiel.

Le logiciel ChocAn a été conçu pour fonctionner sur des lignes de commande, ces composants et fonctionnalités sont les suivantes :

Un menu de démarrage qui permet à l'utilisateur d'entrer selon son statut dans l'organisation ; soit Acme, un Fournisseur, un gérant ou un opérateur ainsi qu'une option de sortie.  
Des menus secondaires qui affichent les options offertes à l'utilisateur, pour Acme les options sont :

- L'option Changer statut du membre : Permet à Acme de changer le statut du membre en introduisant le code dudit membre.

Pour un Fournisseur, un menu principal à lui dont les options sont :

- Activer Terminal.
- Sortir.

Ainsi qu'un menu secondaire affichant les options suivantes :

- Glisser une carte de membre dans le terminal.
- Consulter le répertoire des fournisseurs : Option qui génère un rapport qui est placé dans un fichier .txt dont le chemin d'accès est affiché par la suite.
- Sortir : pour revenir au menu précédent.

Pour le Gérant les options sont :

- Ajouter un membre : Permet l'insertion au système d'un nouveau membre en insérant ses données. Le système affiche par la suite les informations du membre créé.
- Ajouter fournisseur : Permet l'insertion d'un nouveau fournisseur au système en insérant ses données. Le système affiche par la suite les informations du fournisseur créé.
- Ajouter un service : Permet l'insertion d'un nouveau service, les informations demandées sont : nom du service et frais du service.
- Mettre à jour le dossier d'un membre : Permet de changer des informations sauf son numéro d'identification auprès de ChocAn.
- Mettre à jour le dossier d'un fournisseur : Permet de changer des informations du fournisseur sauf son numéro d'identification auprès de ChocAn.

- Mettre à jour un service : Permet de changer des informations d'un service sauf son numéro d'identification auprès de ChocAn.
- Supprimer un membre : Permet de supprimer un membre en demandant une confirmation de la suppression avant de l'effectuer.
- Supprimer un fournisseur : Permet de supprimer un fournisseur en demandant une confirmation de la suppression avant de l'effectuer.
- Suppression d'un service : Permet de supprimer un service, ne demande pas de confirmation.
- Générer un rapport de membre : Permet de générer un rapport qui contient les services reçus par ce membre.
- Générer un rapport de fournisseur : Permet de générer un rapport qui contient les services offerts par ce fournisseur.
- Générer un rapport de Synthèse : Permet de générer un rapport de synthèse.
- Revenir au menu précédent.

Pour un Opérateur, un menu principal dont les options sont les mêmes que pour le Gérant sauf celles de générer les rapports.

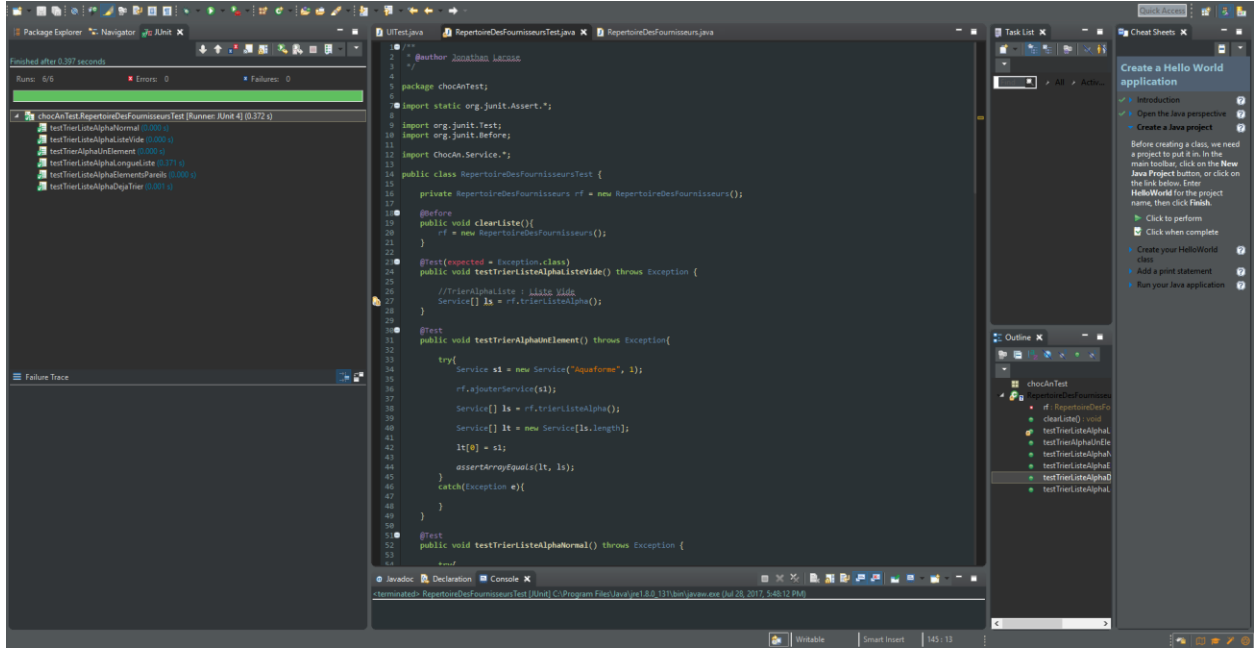
## JAVADOC

\*\* Nous n'avons pas réussi à générer en UTF-8 avec les accents.

## TESTS UNITAIRES

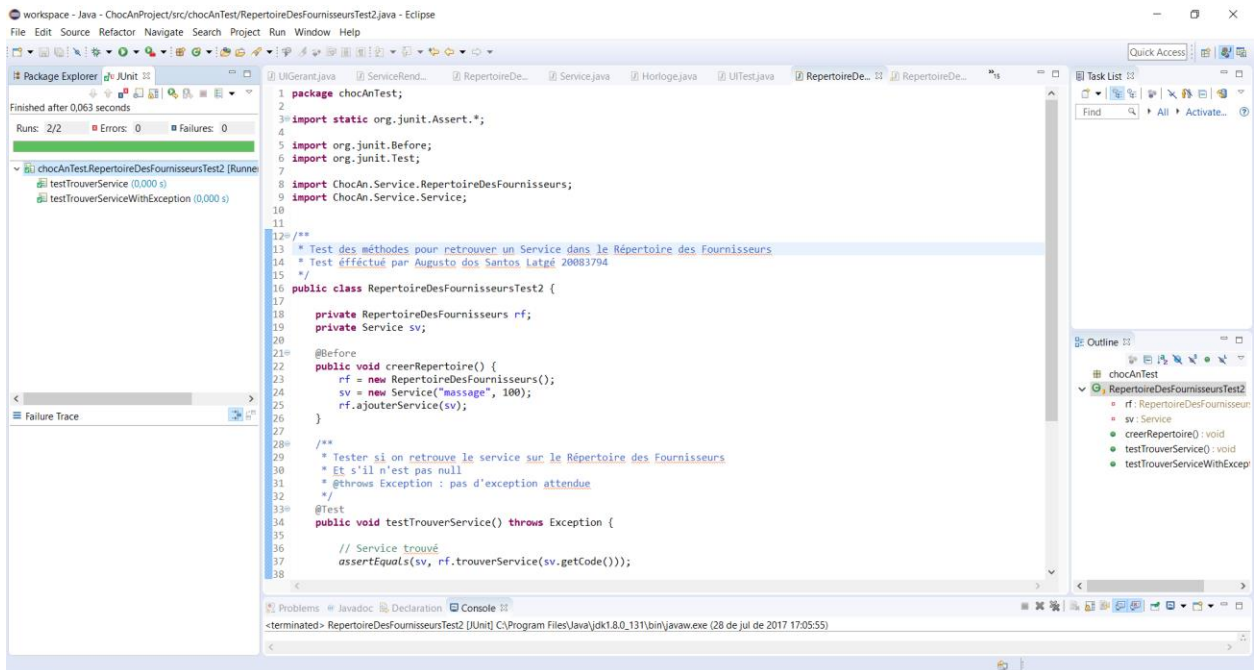
RepertoireDesFournisseurs : trierListeAlpha() par Jonathan Larose

### Méthode qui trie la liste des services disponibles par ordre alphabétique (quicksort)



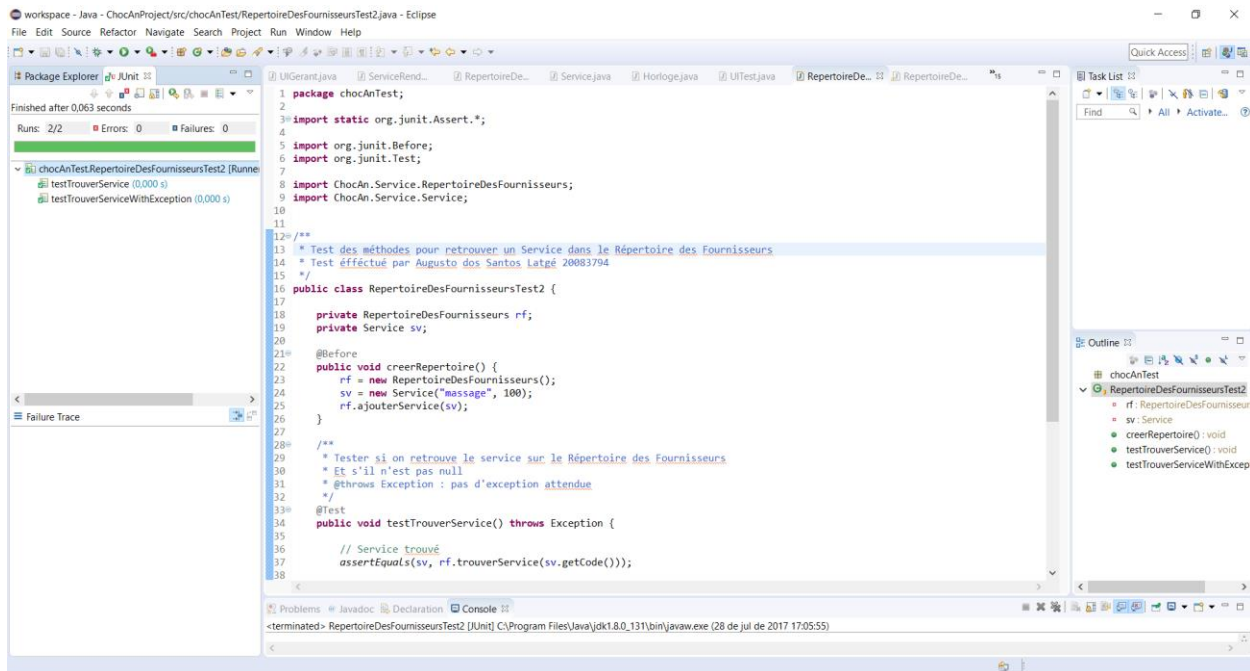
[RepertoireDesFournisseurTest\\_trierListeAlpha.PNG](#)

RepertoireDesFournisseurs : trouverService() par Augusto dos Santos Latgé



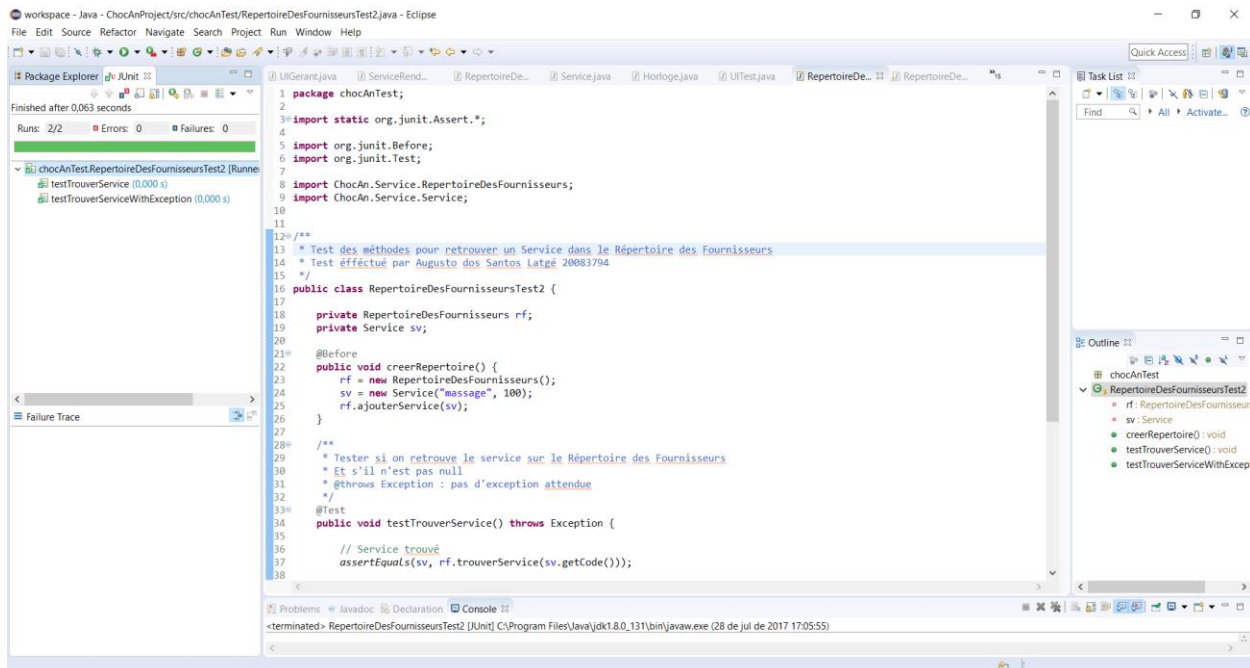
[RepertoireDesFournisseursTest2-trouverService.PNG](#)

## ListeDeCompte : AjouterCompte() par Dennis Orozco Martinez



## ListeDeCompte AjouterCompte.png

## UI : promptOuiNon par Aliona Cupcea



## UITest-promptOuiNon.PNG

## Tous les tests :

The screenshot shows the Eclipse IDE interface for a Java project named 'ChocAnProject'. The Package Explorer on the left lists the following test classes under 'chocAnTest':

- chocAnTestRepertoireDesFournisseursTest2 (Runner: JUnit 4)
- testTrouverService (0.001 s)
- testTrouverServiceWinException (0.000 s)
- chocAnTestUITest (Runner: JUnit 4) (0.000 s)
- testPromptOuNon (0.000 s)
- chocAnServiceListeDeServiceTest (Runner: JUnit 4) (0.000 s)
- ajouterServiceRendu (0.000 s)
- chocAnTestRepertoireDesFournisseursTest (Runner: JUnit 4)
- testTrierListeAlphaNormal (0.000 s)
- testTrierListeAlphaListeVide (0.000 s)
- testTrierListeAlphaUnElement (0.000 s)
- testTrierListeAlphaLongueListe (0.393 s)
- testTrierListeAlphaElementsPareils (0.000 s)
- testTrierListeAlphaDejaTrier (0.000 s)
- chocAnTestListeDeCompteTest (Runner: JUnit 4) (0.002 s)
- testAjoutCompteMembre (0.002 s)
- testAjoutCompteFournisseur (0.000 s)
- testDoublageListeDeCompte2 (0.000 s)

The main editor displays the source code of 'ListeDeCompteTest.java':

```
1 package chocAnTest;
2 import static org.junit.Assert.*;
3 import org.junit.Test;
4 import chocAn.Compte.ListeDeCompte2;
5 import chocAn.Compte.Fournisseur.*;
6 import chocAn.Compte.Membre.*;
7
8 /**
9  * Test des méthodes ajouter membre de la liste de membres
10  * Test effectué par Dennis Orozco Martinez 20031060
11  */
12 public class ListeDeCompteTest {
13
14     @Test
15     public void testAjoutCompteMembre() {
16
17         ListeDeCompte2 l = new ListeDeCompte2(); //Création d'une liste de comptes
18         assertTrue(l.getnombreCompte() == 0); // aucun élément dans la liste
19
20         // Création d'un nouveau compte de membre
21         Membre2 a = new Membre2(null, null, null, null, null, null);
22
23         l.ajouterCompte(a); //Ajout du compte du membre dans la liste de comptes
24         assertEquals(1, l.getnombreCompte()); //un élément dans la liste de comptes
25     }
26
27     @Test
28     public void testAjoutCompteFournisseur(){
29
30         ListeDeCompte2 l = new ListeDeCompte2();
31         //création d'un compte de fournisseur
32         Fournisseur2 b = new Fournisseur2(null, null, null, null, null, null);
33
34         l.ajouterCompte(b); //Ajout d'un compte de fournisseur dans la liste de comptes
35         assertEquals(1, l.getnombreCompte()); //un élément dans la liste de comptes
36     }
37 }
```

The console at the bottom shows the output of the tests, indicating they passed:

```
<terminated> chocAnProject [JUnit] C:\Program Files\Java\jdk1.8.0_131\bin\java.exe (28 de jul de 2017 22:35:34)
Entrez oui ou non :
1. Oui.
2. Non.
```

An 'Updates Available' dialog is visible in the bottom right corner, stating: 'Updates are available for your software. Click to review and install updates. Set up Reminder options'.



# Collaboration

## statistique.PNG

July 16, 2017 - July 29, 2017

To select a time span double click near its start point and drag to its end point.

Display in Days Weeks Months

