

SPÉCIFICATIONS DES EXIGENCES D'UN LOGICIEL

(Adapté de la norme IEEE 830-1993)

SGBA : SYSTÈME DE GESTION D'UNE BANQUE ALIMENTAIRE

**VERSION DU DOCUMENT :
1.0.0**

PRÉSENTÉ PAR :

**Mathieu Branchaud
Martin Gaboury
Frédéric Legendre
Paul Lessard
Benoît Meilleur**

**DANS LE CADRE DU COURS INM5151
ATELIER D'ANALYSE ET DE MODÉLISATION**

**UNIVERSITÉ DU QUÉBEC À MONTRÉAL
12 MARS 2007**

Table des matières

Table des matières.....	2
Table des illustrations.....	6
1. Introduction.....	7
1.1 Objectifs.....	7
1.2 Portée.....	7
1.3 Définitions, acronymes et abréviations.....	9
1.4 Documents de référence.....	12
1.5 Aperçu du document.....	13
2. Description générale du logiciel.....	14
2.1 Perspective du produit.....	14
2.1.1 Autres parties du système.....	15
2.1.2 Utilisateurs (usagers manipulant le système).....	16
2.1.3 Composantes matérielles.....	16
2.1.4 Composantes logicielles.....	16
2.1.5 Télécommunications.....	16
2.1.6 Contraintes de mémoire primaire et secondaire.....	17
2.1.7 Interfaces aux opérateurs.....	17
2.1.8 Adaptation du site d'installation.....	18
2.2 Vue d'ensemble des fonctions du produit.....	19
2.3 Description des utilisateurs.....	22
2.4 Contraintes d'ordre général.....	23
2.4.1 Règlements des organismes gouvernementaux, syndicaux	23
2.4.2 Limitations du matériel.....	23
2.4.3 Interfaces à d'autres applications.....	23
2.4.4 Opérations en parallèle.....	24
2.4.5 Fonctions d'audit.....	24
2.4.6 Fonctions de contrôle.....	24
2.4.7 Exigences de langages de programmation	24
2.4.8 Protocoles de communication.....	25
2.4.9 Exigences de fiabilité.....	25
2.4.10 Aspects critiques de l'application.....	25
2.4.11 Considérations de sûreté et de sécurité.....	25
2.5 Hypothèses et dépendances.....	27
2.6 Répartition des exigences.....	28
3. Description détaillée.....	30
3.1 Interfaces externes.....	30
3.1.1 Ajouter un dossier d'utilisateur.....	30
3.1.2 Mettre à jour un dossier d'utilisateur.....	31
3.1.3 Supprimer un dossier d'utilisateur.....	32
3.1.4 Réinscrire un utilisateur.....	33
3.1.5 Rechercher un dossier d'utilisateur.....	34
3.1.6 Afficher un dossier d'utilisateur.....	35
3.1.7 Rembourser un crédit.....	36
3.1.8 Enregistrer la visite d'un utilisateur.....	37
3.1.9 Changer le statut des utilisateurs pour « à réinscrire ».....	38

3.1.10 Enregistrer un crédit à un utilisateur.....	39
3.1.11 Consulter un compte d'utilisateur.....	40
3.1.12 Créer un compte d'utilisateur.....	41
3.1.13 Mettre à jour un compte d'utilisateur.....	42
3.1.14 Supprimer un compte d'utilisateur.....	43
3.1.15 Gérer le niveau de permission d'un usager.....	44
3.1.16 Obliger l'utilisateur à changer son mot de passe.....	45
3.1.17 Créer un nouvel atelier.....	46
3.1.18 Mettre à jour un atelier.....	47
3.1.19 Supprimer un atelier.....	48
3.1.20 Créer un groupe-cours.....	49
3.1.21 Mettre à jour un groupe-cours.....	50
3.1.22 Supprimer un groupe-cours.....	51
3.1.23 Inscrire un utilisateur à un atelier.....	52
3.1.24 Supprimer l'inscription d'un utilisateur à un atelier.....	53
3.1.25 Saisir le résultat d'un utilisateur à un atelier.....	54
3.1.26 Obtenir la liste des inscriptions à un atelier.....	55
3.1.27 Créer une activité.....	56
3.1.28 Mettre à jour une activité.....	57
3.1.29 Supprimer une activité.....	58
3.1.30 Inscrire un utilisateur à une activité.....	59
3.1.31 Supprimer l'inscription d'un utilisateur à une activité.....	60
3.1.32 Obtenir la liste des inscriptions à une activité.....	61
3.1.33 Générer un rapport.....	62
3.1.34 Afficher les statistiques du système.....	63
3.1.35 Authentifier un usager.....	64
3.2 Spécifications fonctionnelles.....	65
3.2.1 Diagrammes de séquence.....	65
3.2.1.1 Ajouter un dossier d'utilisateur.....	65
3.2.1.2 Mettre à jour un dossier d'utilisateur.....	66
3.2.1.3 Supprimer un dossier d'utilisateur.....	67
3.2.1.5 Rechercher un dossier d'utilisateur.....	69
3.2.1.6 Afficher un dossier d'utilisateur.....	70
3.2.1.7 Rembourser un crédit.....	71
3.2.1.8 Enregistrer la visite d'un utilisateur.....	72
3.2.1.9 Changer le statut des utilisateurs pour « à réinscrire ».....	74
3.2.1.10 Enregistrer un crédit à un utilisateur.....	75
3.2.1.11 Consulter un compte d'utilisateur.....	76
3.2.1.12 Créer un compte d'utilisateur.....	77
3.2.1.13 Mettre à jour un compte d'utilisateur.....	78
3.2.1.14 Supprimer un compte d'utilisateur.....	79
3.2.1.15 Gérer le niveau de permission d'un usager.....	80
3.2.1.16 Obliger l'utilisateur à changer son mot de passe.....	81
3.2.1.17 Créer un nouvel atelier.....	82
3.2.1.18 Mettre à jour un atelier.....	83
3.2.1.21 Mettre à jour un groupe-cours.....	86
3.2.1.22 Supprimer un groupe-cours.....	87
3.2.1.23 Inscrire un utilisateur à un atelier.....	88

3.2.1.24	Supprimer l'inscription d'un utilisateur à un atelier.....	89
3.2.1.25	Saisir le résultat d'un utilisateur à un atelier.....	90
3.2.1.26	Obtenir la liste des inscriptions à un atelier.....	91
3.2.1.27	Créer une activité.....	92
3.2.1.28	Mettre à jour une activité.....	93
3.2.1.29	Supprimer une activité.....	94
3.2.1.30	Inscrire un utilisateur à une activité.....	95
3.2.1.31	Supprimer l'inscription d'un utilisateur à une activité.....	96
3.2.1.32	Obtenir la liste des inscriptions à une activité.....	97
3.2.1.33	Générer un rapport.....	98
3.2.1.34	Afficher les statistiques du système.....	99
3.2.1.35	Authentifier un usager.....	100
3.2.2	Contrats.....	101
3.2.2.1	CO01 Ajouter un dossier d'utilisateur.....	101
3.2.2.2	CO02 Mettre à jour un dossier d'utilisateur.....	101
3.2.2.3	CO03 Supprimer un dossier d'utilisateur.....	101
3.2.2.4	CO04 Réinscrire un utilisateur.....	101
3.2.2.5	CO05 Rechercher un dossier d'utilisateur.....	102
3.2.2.6	CO06 Afficher un dossier d'utilisateur.....	102
3.2.2.7	CO07 Rembourser un crédit.....	102
3.2.2.8	CO08 Enregistrer la visite d'un utilisateur.....	102
3.2.2.9	CO09 Changer le statut des utilisateurs pour « à réinscrire ».....	103
3.2.2.10	CO10 Enregistrer un crédit à un utilisateur.....	103
3.2.2.11	CO11 Consulter un compte d'usager.....	103
3.2.2.12	CO12 Créer un compte d'usager.....	104
3.2.2.13	CO13 Mettre à jour un compte d'usager.....	104
3.2.2.14	CO14 Supprimer un compte d'usager.....	104
3.2.2.15	CO15 Gérer le niveau de permission d'un usager.....	104
3.2.2.16	CO16 Obliger l'usager à changer son mot de passe.....	105
3.2.2.17	CO17 Créer un nouvel atelier.....	105
3.2.2.18	CO18 Mettre à jour un atelier.....	105
3.2.2.19	CO19 Supprimer un atelier.....	105
3.2.2.20	CO20 Créer un groupe-cours.....	106
3.2.2.21	CO21 Mettre à jour un groupe-cours.....	106
3.2.2.22	CO22 Supprimer un groupe-cours.....	106
3.2.2.23	CO23 Inscrire un utilisateur à un atelier.....	106
3.2.2.24	CO24 Supprimer l'inscription d'un utilisateur à un atelier.....	107
3.2.2.25	CO25 Saisir le résultat d'un utilisateur à un atelier.....	107
3.2.2.26	CO26 Obtenir la liste des inscriptions à un atelier.....	107
3.2.2.27	CO27 Créer une activité.....	107
3.2.2.28	CO28 Mettre à jour une activité.....	108
3.2.2.29	CO29 Supprimer une activité.....	108
3.2.2.30	CO30 Inscrire un utilisateur à une activité.....	108
3.2.2.31	CO31 Supprimer l'inscription d'un utilisateur à une activité.....	108
3.2.2.32	CO32 Obtenir la liste des inscriptions à une activité.....	109
3.2.2.33	CO33 Générer un rapport.....	109
3.2.2.34	CO34 Afficher les statistiques du système.....	109
3.2.2.35	CO35 Authentifier un usager.....	109

3.3 Exigences d'opérations, de communications et de performance.....	110
3.3.1 Nombre de terminaux à supporter.....	110
3.3.2 Nombre d'utilisateurs simultanés.....	110
3.3.3 Quantité et types d'information à traiter.....	110
3.3.4 Nombre de transactions à traiter par unité de temps sous conditions normales et en période de pointe.....	110
3.4 Exigences logiques de bases de données.....	111
3.4.1 Types d'informations utilisées par les différentes fonctions.....	111
3.4.2 Fréquence d'utilisation.....	111
3.4.3 Capacité d'accès.....	111
3.4.4 Entités et relations des données.....	112
3.4.5 Contraintes d'intégrité.....	116
3.4.6 Exigences de rétention de données.....	116
3.5 Contraintes de conception.....	117
3.5.1 Contraintes de matériel.....	117
3.5.2 Format des rapports.....	117
3.5.3 Nomenclature des données.....	118
3.5.4 Procédures de comptabilité.....	118
3.5.5 Exigences d'audit, traçabilité des traitements.....	118
3.6 Exigences non-fonctionnelles.....	119
3.6.1 Fiabilité.....	119
3.6.2 Disponibilité.....	119
3.6.3 Sécurité.....	119
3.6.4 Rendement.....	119
3.6.5 Portabilité.....	119
4. Informations complémentaires.....	120
4.1 Annexe : aperçu de l'interface.....	120
4.1.1 Menu principal (Personnel administratif).....	120
4.1.2 Gestion des utilisateurs – Recherche et création d'un dossier.....	121
4.1.3 Modification d'un utilisateur (aperçu).....	122
4.2 Exemple de script Oracle XE.....	124

Table des illustrations

- Figure 1** : Répartition des tâches selon le niveau de permission (p. 15)
- Figure 2** : Modèle relationnel de la base de données du SGBA (p. 113)
- Figure 3** : Menu principal du personnel administratif (ébauche) (p. 122)
- Figure 4** : Ébauche du menu de gestion des utilisateurs (p. 123)
- Figure 5** : Ébauche de la création d'un nouvel utilisateur (p. 124)
- Figure 6** : Ébauche de l'enregistrement des visites (p. 125)

1. Introduction

Le présent document a pour mission de donner suite au projet du système de gestion d'une banque alimentaire (SGBA), dont le concept a été déposé précédemment au moyen du document IEEE 1362 (concept of operations). Nous exposerons ici les spécifications générales et détaillées de la solution qui devra être conçue et développée dans le cadre de ce projet, ce que nous ferons en décrivant formellement son comportement ainsi que les charges qu'il devra être en mesure de supporter. Le document couvrira également les contraintes à respecter ainsi que l'environnement cible.

1.1 Objectifs

Le document de spécifications des exigences d'un logiciel vise à préciser dans une langue claire les exigences et la portée du logiciel en développement, ce que nous appliquerons ici à notre projet particulier, le SGBA. L'ensemble des fonctionnalités et utilisations seront décrites et décortiquées afin d'en exprimer le fonctionnement détaillé.

Bien que davantage technique que le document de concept des opérations, il vise néanmoins l'ensemble du personnel impliqué dans le développement du logiciel, particulièrement les architectes, analystes, programmeurs et testeurs, sans négliger toutefois les gestionnaires de l'organisme d'aide alimentaire bénéficiaire, en particulier pour ce qui a trait à la description générale du logiciel.

1.2 Portée

L'informatisation des activités d'affaires des grandes entreprises n'est pas nouvelle. Cependant, les coûts associés à une intégration des services de plus petits organismes, particulièrement dans un domaine à but non lucratif, ont souvent rendu la transition de ces dernières vers une exploitation laissant de plus en plus de place à l'automatisation difficile ou même impossible, celles-ci ne possédant d'ordinaire que de modestes moyens et s'avérant dans l'incapacité de disposer du personnel technique nécessaire à un développement logiciel sur mesure ou à acquérir une solution commerciale. Parmi ces organismes, les banques alimentaires, en première ligne de la lutte à la pauvreté, consacrant presque tous leurs moyens financiers à leur cause, ne disposent pas des moyens nécessaires à l'acquisition ou au développement d'une solution informatisée, même si paradoxalement cette dernière leur permettrait, au final, de consacrer davantage à leurs bénéficiaires, en raison des économies en temps provoquées par le logiciel.

Le système de gestion de banque alimentaire (SGBA) a pour mission de répondre aux exigences de base des activités associées aux utilisateurs d'un petit organisme de distribution alimentaire offrant également des ateliers éducatifs à ses bénéficiaires et

désirant posséder un outil simple qui permettra également la génération automatisée de rapports.

Le SGBA vise la couverture des tâches cléricales suivantes :

- La gestion de dossiers d'utilisateurs (création, modification, suppression) ainsi que de la composition de leurs ménages et le suivi de l'état des dossiers au moyen de statuts,
- L'enregistrement des visites d'utilisateurs, ainsi que des crédits monétaires qui leurs sont éventuellement associés,
- La gestion des ateliers éducatifs, l'inscription des utilisateurs à celles-ci ainsi que la saisie des résultats,
- La gestion d'activités facultatives et l'inscription d'utilisateurs à ces dernières.

Les tâches de gestion suivantes seront également couvertes :

- La génération de statistiques diverses (visites par mois, par ville, etc.),
- La production de rapports administratifs.

Le système devra aussi être paramétrable et supporter la création de comptes d'utilisateurs pour le personnel en fonction de ses permissions d'accès.

Cependant, le logiciel ne couvrira aucune tâche d'inventaire ou de comptabilité. Il ne traitera pas avec la moindre application externe et ne partagera aucune donnée avec l'extérieur. Il ne sera pas accessible en dehors du contexte de l'organisme, ces derniers ne souhaitant pas qu'il soit même imaginable qu'un intrus de l'extérieur puisse parvenir à accéder à pareilles données hautement confidentielles.

Le SGBA s'intégrera dans un environnement peu ou pas informatisé. La plupart des organismes pouvant être intéressés par son exploitation fonctionnent par tables papiers ou utilisent des feuilles simples sur Excel. Par définition, il sera implanté pour remplacer cette situation actuelle, en migrant cette société vers un environnement d'exploitation informatisé, ce qui réduira considérablement les erreurs dues aux manipulations de dossiers papier tout en accroissant de manière significative l'efficacité d'une organisation ne disposant que d'un personnel réduit et le ratio nombre de dossiers traités par heure travaillée.

1.3 Définitions, acronymes et abréviations

Terme employé dans le document	Définition
Actif (utilisateur)	Statut d'un utilisateur ayant réussi tous les ateliers exigés par l'organisme et effectuant des visites régulièrement (sa dernière visite remonte à moins de 2 ans)
Activité	Cours d'apprentissage ou présentation ludique, une activité est un événement unique dans le temps auquel un nombre défini d'utilisateurs peut participer. Une activité n'est pas évaluée et les participations ne sont pas conservées dans le temps.
Administrateur	Niveau de permission d'un usager, qui a accès aux fonctions de paramétrage de l'application au niveau des règles d'affaires. Est également le seul niveau de permission permettant de créer des comptes d'utilisateurs et de leur attribuer leurs permissions.
À réinscrire	Statut annuel d'un utilisateur, qu'il soit conditionnel ou actif, afin de vérifier la rectitude de ses coordonnées personnelles. La date à laquelle le statut des utilisateurs devient « à réinscrire » est paramétrable.
Atelier	Cours magistral ou démonstrations pratiques, les activités sont des enseignements obligatoires dans la démarche d'un utilisateur auprès de l'organisme. Un utilisateur réussissant le dernier atelier non complété voit son statut changer pour « actif ».
Conditionnel (utilisateur)	Statut d'un nouvel utilisateur n'ayant pas encore complété ses ateliers. Il n'est pas limité outre mesure pour ce qui est des droits de visite, ce qui n'empêche pas l'organisme de limiter leurs privilèges comme il l'entend.

Crédit	Prêt d'argent accordé à l'utilisateur qui n'est pas en mesure d'apporter sa contribution à une visite alimentaire (si l'organisme l'exige). Un nombre trop élevé de crédits bloque l'enregistrement d'une visite. Le nombre maximal de crédits est paramétrable par l'administrateur.
Cryptage / Chiffrement	Techniques visant à limiter la compréhension d'un document aux seuls destinataires et destinateurs. Dans le contexte du SGBA, fonctions de chiffrement de certaines données sensibles afin de prévenir la fuite d'informations en cas de vol d'équipement de stockage.
Gestionnaire	Niveau de permission d'utilisateur donnant accès aux modules de statistiques et de rapports.
Groupe-cours	Dans le contexte d'un atelier, présentation de ce dernier (exactement comme dans une institution d'études postsecondaires) par un membre du personnel éducatif, à un lieu et à un moment donné. Un groupe-cours comporte un nombre de places défini à l'avance.
Inactif (utilisateur)	Statut d'un utilisateur qui n'a pas effectué de visite depuis un nombre d'années paramétrable par l'administrateur. Un utilisateur dont le statut est inactif n'apparaît pas dans les listes d'utilisateurs destinées aux visites, et ne peut enregistrer de visites.
Lien familial	Se dit d'une personne à charge de l'utilisateur. Peut être un époux, enfant ou autre vivant sous le même toit et devant bénéficier indirectement des services de l'organisme. Les liens sont très utiles pour les statistiques et les demandes de subventions.
Permission (usager)	La permission d'un usager est sa catégorie, c'est-à-dire les fonctions du logiciel auxquelles il a accès, par rapport à son rôle dans l'organisme d'aide alimentaire.

Personnel administratif	Le personnel administratif est la majorité des employés de l'organisme, ceux qui manipulent directement le système en gérant les dossiers d'utilisateurs, les activités et les ateliers. Ils enregistrent également les visites. La permission « personnel administratif » donne accès aux fonctions requises pour les tâches précédemment citées.
Personnel éducatif	À l'instar du personnel administratif, le personnel éducatif est une permission d'utilisateur. Un membre du personnel éducatif est la plupart du temps un bénévole qui enseigne un groupe-cours lié à un atelier. Il a accès aux fonctions associées à cette tâche.
SGBA	Système de gestion d'une banque alimentaire, le projet décrit dans ce document.
SGBD	Système de gestion de bases de données
Statut (utilisateur)	Le statut d'un utilisateur est une représentation de son niveau d'intégration dans l'organisme, une expression de sa démarche d'aide alimentaire. Il représente ses « droits » de visite dans l'organisme et l'état de son dossier.
Usager	Un compte d'utilisateur au sein du système, accordé à un membre du personnel de l'organisme ou à un bénévole. Il possède un niveau de permission et est distingué par un nom de compte qui lui est propre ainsi qu'un mot de passe.
Utilisateur	Un terme neutre pour désigner les bénéficiaires venant solliciter une aide alimentaire par l'organisme.
Visite	La présence d'un utilisateur lors d'une distribution d'aliments. L'utilisateur a le droit d'effectuer un nombre paramétrable de visites pour une plage de temps donnée. Si l'organisme applique la règle associée, une visite peut coûter une certaine somme d'argent, mais on peut lui faire crédit si nécessaire (les crédits étant gérés par le système).

1.4 Documents de référence

Équipe du projet SGBA – *Le concept des opérations*, 2007
Larman Craig, *Applying UML and Patterns 3^e édition*, Prentice Hall, 2005
IEEE-830 – *Spécifications des exigences d'un logiciel (résumé)*

Règles régissant la conservation d'informations personnelles :
<http://e-com.ic.gc.ca/epic/site/ecic-ceac.nsf/fr/gv00244f.html>

Plusieurs documents de spécifications des exigences d'un logiciel mis à la disposition de l'équipe ont été consultés à titre d'exemples.

1.5 Aperçu du document

Le document de spécifications d'exigences logicielles appliquées au SGBA est divisé en quatre chapitres distincts auxquels viennent s'ajouter un nombre réduit d'annexes.

Le chapitre un est une introduction au document même, à ses objectifs, puis présente le projet d'un SGBA, les acronymies et définitions de temps utilisés au sein du document et une liste de documents de références utilisés lors de la rédaction.

Le chapitre deux décrit de manière générale le projet SGBA et met la table pour les précisions qui seront offertes au chapitre trois.

Le chapitre trois présente les exigences et les spécifications détaillées associées au système (cas d'utilisation, diagrammes de séquence et contrats) et les exigences non fonctionnelles à un niveau très précis.

Le chapitre quatre s'avérera un aperçu des interfaces développées dans le cadre du projet dans un stade préliminaire.

2. Description générale du logiciel

2.1 Perspective du produit

La motivation profonde au développement d'un système de gestion d'une banque alimentaire est de tenter d'offrir l'occasion aux sociétés oeuvrant dans ces milieux et ne disposant pas d'une solution logicielle permettant l'analyse, la manipulation et la gestion des informations confidentielles que ces dernières doivent préserver afin de permettre d'atteindre leurs objectifs de posséder un outil à la fois simple à utiliser et hautement paramétrable, pour s'adapter à leurs situations particulières.

Le SGBA aura pour tâche de substituer à un conglomérat plus ou moins archaïque de pratiques héritées d'un manque flagrant de moyens financiers et de compétences techniques dans le domaine informatique d'un organisme de charité disposant de ressources fort limitées un logiciel permettant l'accomplissement de la majorité des opérations concernant directement le service à la clientèle, ce qui simplifiera considérablement l'accomplissement de leurs traitements courants.

D'un point de vue purement opérationnel, l'exploitation du SGBA accélérera considérablement le traitement des tâches quotidiennes tout en réduisant les erreurs provoquées par les manipulations manuelles en éliminant les supports papiers et en uniformisant les informations stockées et les moyens d'y accéder. Cette information actuellement conservée sur des supports hétérogènes sera centralisée pour en simplifier les procédures de sauvegardes et d'audit.

Une fois complètement implanté au sein d'un organisme typique, le SGBA englobera les opérations liées aux tâches administratives de l'ensemble des travailleurs, salariés, gestionnaires et bénévoles, mais uniquement en ce qui concerne le service à la clientèle direct (la distribution contrôlée et réglementée d'aliments). Il s'agit d'une solution logicielle utilisée uniquement à l'interne, et qui ne disposera d'aucun moyen de communiquer avec l'extérieur (son fonctionnement ne nécessitant pas d'informations en provenance d'un autre système).

Le SGBA permettra aux membres du personnel administratif de l'organisme bénéficiaire de saisir les dossiers de ses utilisateurs, d'assurer leur mise à jour par un système de statuts et de les inscrire aux activités et ateliers éducatifs qu'ils jugent nécessaires à leur réinsertion sociale. Le système disposera d'un moyen fiable de demandes de visites alimentaires qui sera validé au moyen de règles d'affaires paramétrables, permettant de justifier adéquatement un refus, lorsque cela sera nécessaire. Les membres du personnel éducatif, quant à eux, seront en mesure de créer des groupes où ils enseignent le contenu d'un atelier, auxquels les utilisateurs pourront s'inscrire. Le personnel éducatif sera en mesure d'imprimer les listes d'inscriptions associées à ses groupes, puis de saisir les résultats de ses élèves. La nature du nouveau système permettra la parallélisation de toutes ces tâches, ce qui était impossible par le passé en raison de la nature statique et des moyens de manipulation des dossiers papiers.

En plus de ces fonctions courantes, les gestionnaires de l'organisme disposeront d'un module de statistiques et de rapports paramétrables. L'ensemble du logiciel sera bâti pour faciliter le stockage d'informations nécessaires à la production de rapports destinés aux organismes subventionnaires (e.g. nombre de visites accordés à des utilisateurs masculins subvenant aux besoins d'une famille ayant plus de 3 enfants mineurs). Les gestionnaires disposeront également d'un audit sur l'ensemble des transactions, qui permettra de lier facilement une transaction à un usager du système.

De nombreuses options et paramètres permettront à l'administrateur de créer les dossiers d'utilisateurs du système, et surtout d'adapter le logiciel aux besoins particuliers de son organisme. Les règles d'affaires seront paramétrables et éventuellement extensibles.

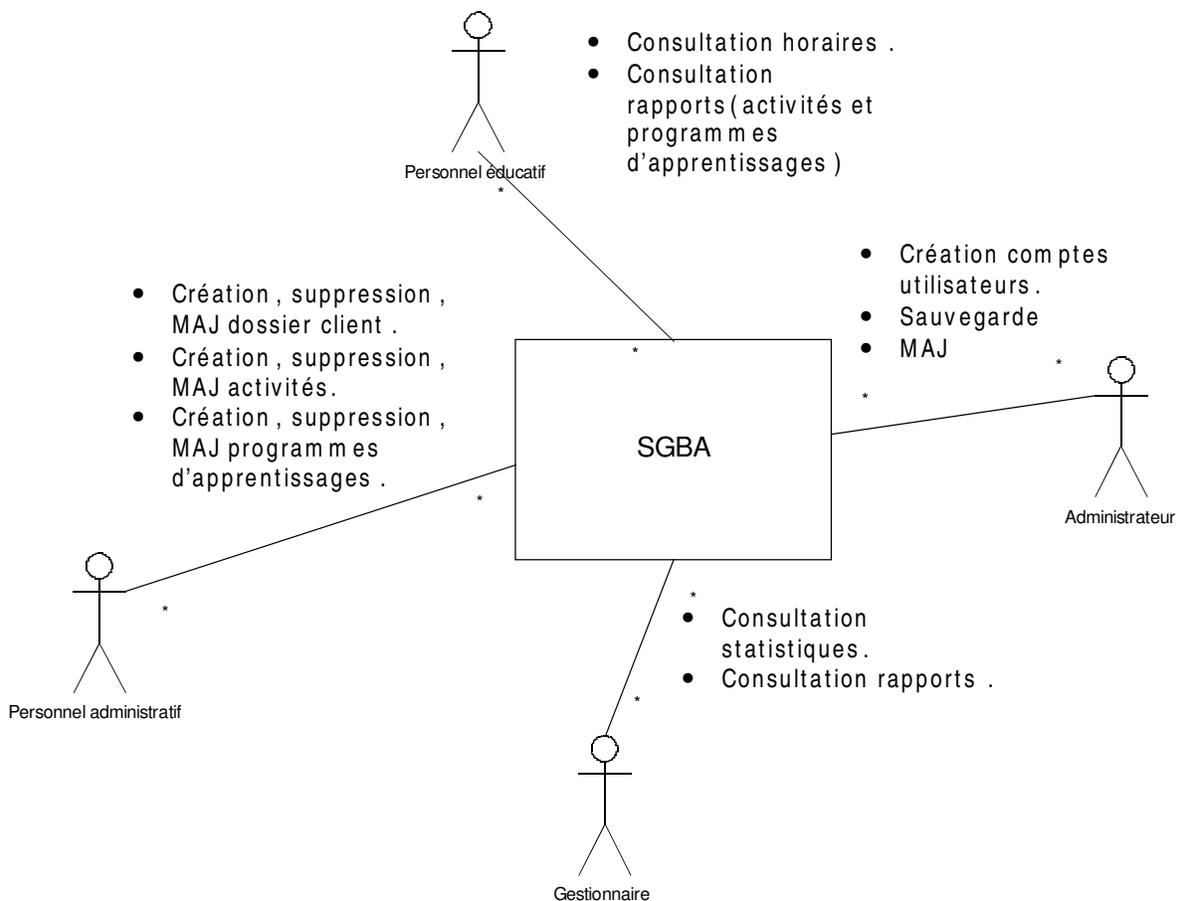


Figure 1 : Répartition des tâches selon le niveau de permission

2.1.1 Autres parties du système

Bien que cela ne représente pas une autre partie à strictement parler, mais plutôt plusieurs composantes formant un tout, l'ensemble du système ne sera pas déployé sur une seule machine, puisque le stockage de données sera séparé de la présentation et du traitement.

2.1.2 Utilisateurs (usagers manipulant le système)

Le SGBA disposera de quatre niveaux de « permissions », qui représentent les accès permis à quatre différentes catégories d'utilisateurs, soient le personnel administratif, le personnel éducatif, les gestionnaires et l'administrateur. Le niveau de permission d'un utilisateur déterminera l'ensemble des interfaces auxquelles il aura accès. Ces distinctions seront décrites plus précisément à la section 2.3.

2.1.3 Composantes matérielles

La précarité des composantes matérielles disponibles influencera directement le déploiement de celui-ci, et doit être considéré lors des étapes préalables, notamment en ce qui a trait à la sécurité nécessaire, en raison de la nature hautement confidentielle des renseignements stockés et des risques physiques des locaux où le matériel sera entreposé. Nous prévoyons que la solution d'une installation totalement interne sera retenue, mais cette solution nécessitera la présence d'au moins deux ordinateurs personnels, dont l'un servira uniquement au stockage des informations (hébergement de la base de données). Ces machines devront disposer de la capacité de communiquer (réseau privé), et la machine de stockage devra être suffisamment puissante pour répondre aux requêtes dans un délai acceptable.

Cette approche aura comme avantage d'isoler complètement de la vue d'autrui (même d'un hébergeur) les informations qu'il stockera. Il sera toutefois nécessaire de mettre en place une certaine solution de rechange en cas de panne et surtout une procédure rigoureuse de sauvegarde des données.

2.1.4 Composantes logicielles

Les composantes logicielles auxquelles le SGBA s'interfacera dépendront du système d'exploitation utilisé, étant donné la nature portable du langage dans lequel le logiciel sera développé. Le système d'exploitation devra supporter le SGBD Oracle qui contiendra les données, ainsi que les protocoles réseaux nécessaires à son interconnexion avec les clients SGBA. Les autres logiciels éventuellement installés sur les stations clientes ne devraient pas entrer en conflit avec le SGBA (anti-virus, anti-spyware, suite bureautique, navigateur Web, etc.) ni communiquer avec lui.

2.1.5 Télécommunications

Bien que nécessitant l'utilisation d'un réseau local, même un faible débit¹ s'avérerait suffisant à une exploitation efficace du SGBA, puisque les distances parcourues par les

¹ En date de mars 2007, un faible débit pour un réseau local serait de 10 mégabits par seconde.

signaux seront minimales. Une attention particulière devra être portée à la sécurité du réseau si celui-ci est de type sans-fil, le cryptage s'imposant d'emblée. Le SGBA n'impose pas de contrainte particulière sur la configuration des équipements d'interconnexion.

2.1.6 Contraintes de mémoire primaire et secondaire

Les contraintes de mémoire imposées par le SGBA ne sont pas énormes, étant proportionnelles à la taille de la base de données de l'organisme bénéficiaire. Nous n'estimons pas que celle-ci puisse nécessiter une quantité importante de mémoire physique, la disposition RAID 1 pouvant cependant être utile d'un point de vue de la redondance des informations stockées.

La mémoire physique des machines clientes n'a pas besoin d'être importante, et variera en fonction du système d'exploitation utilisé et de la machine virtuelle Java employée. Face à la capacité physique des disques durs actuels², la faible taille de la partie client du SGBA (que nous estimons au pire à quelques mégaoctets) ne représente aucune contrainte considérable.

La configuration des machines clientes ne nécessite pas de sauvegarde physique. Au plus, une procédure fixe (« logbook ») permettra de reconfigurer rapidement et simplement ces stations de travail en cas de remplacement. Ce document ne devra évidemment pas contenir les mots de passe permettant d'accéder à la base de données. Par contre, les fichiers de configuration (règles d'affaires) devront être sauvegardés à la manière que nous décrivons plus bas pour la base de données.

Il en va tout autrement pour ce qui a trait aux sauvegardes de la base de données, qui devront être rigoureusement respectées. Nous recommandons l'instauration d'une procédure de sauvegarde complète bimensuelle, en rotation sur 3 jeux de rubans, en plus de sauvegardes différentielles après chaque jour de distribution alimentaire (ce qui peut varier en fonction de l'organisme, d'une à trois fois par semaine ou encore davantage). Si les moyens financiers ne permettent pas de disposer d'autant de jeux de supports de sauvegarde, on pourrait réduire la fréquence des sauvegardes différentielles et augmenter celle des sauvegardes complètes, puisque la base de données est petite et que le système ne sera pas consulté en dehors des heures d'ouverture de l'organisme.

La question de la sécurité des copies de sauvegarde ne doit pas être négligée. L'organisme devrait entreposer au moins un jeu récent à l'externe quelques fois par année. De plus, tous les rubans entreposés à l'interne devront être isolés des lieux fréquentés et conservés sous clef.

2.1.7 Interfaces aux opérateurs

² Mars 2007

Les fonctions de support et d'archivage ne s'avérant pas directement manipulés par le SGBA, que ce soit directement ou indirectement, celui-ci n'admet pas leur existence du point de vue du présent document.

2.1.8 Adaptation du site d'installation

Les adaptations nécessaires au site d'installation concerneront essentiellement la disposition physique des stations de travail qui devront être placées de manière à pouvoir servir lors des distributions alimentaires, afin d'optimiser le contact entre les utilisateurs et le personnel de l'organisme. Éventuellement, des travaux électriques pourraient s'avérer nécessaires à leur disposition, et l'on profiterait de l'occasion pour filer les locaux au format Ethernet, afin de permettre aux stations de rejoindre le serveur de base de données. Celui-ci pourrait être installé dans un local isolé. Étant donné la pauvreté des utilisateurs et les risques que celle-ci entraîne, la sécurité physique des locaux devra être adaptée pour éviter le vol des équipements, particulièrement le serveur et l'entrepôt des sauvegardes.

2.2 Vue d'ensemble des fonctions du produit

Le SGBA, une fois développé, devra être en mesure d'offrir les fonctionnalités suivantes :

1. **Ajouter un dossier d'utilisateur** : permet à un membre du personnel administratif de créer un nouveau dossier d'utilisateur qui souhaite bénéficier des services de l'organisme bénéficiaire.
2. **Mettre à jour un dossier d'utilisateur** : assure la capacité de mise à jour des dossiers d'utilisateurs en ce qui concerne les informations personnelles stockées par le SGBA.
3. **Supprimer un dossier d'utilisateur** : permet de supprimer un dossier d'utilisateur. Nous ne croyons pas que cette fonction soit utilisée très souvent en ce qui a trait à la non utilisation des dossiers – elle relèverait essentiellement d'une erreur concernant un dossier, ou dans le but d'éliminer un doublon.
4. **Réinscrire un utilisateur** : permet de régulariser le statut d'un utilisateur inactif ou marqué comme « à réinscrire ».
5. **Rechercher un dossier d'utilisateur** : assure la capacité de repérer un dossier d'utilisateur en fonction de critères de recherche.
6. **Afficher un dossier d'utilisateur** : affiche les informations appropriées concernant un dossier d'utilisateur en particulier.
7. **Rembourser un crédit** : permet de rembourser un crédit préalablement accordé lors d'une visite antérieure.
8. **Enregistrer la visite d'un utilisateur** : permet d'enregistrer la collecte de sacs de nourriture (visite alimentaire) d'un utilisateur en temps réel – ce qui permet d'en assurer la légalité par rapport aux règles d'affaires imposées par l'organisme.
9. **Changer le statut des utilisateurs pour « à réinscrire »** : tâche automatique et temporelle, qui permet de forcer la réinscription annuelle des utilisateurs, si elle est activée.
10. **Enregistrer un crédit à un utilisateur** : invoquée dans le cadre d'une visite, cette fonction du logiciel permet de vérifier s'il est possible d'accorder un crédit à un utilisateur, et à l'enregistrer si c'est le cas.
11. **Consulter un compte d'utilisateur** : affiche les informations associées à un compte d'utilisateur (qui permet d'accéder au système, par rapport à un dossier d'utilisateur, qui a recours aux services de l'organisme sans jamais interagir directement avec le SGBA).

12. **Créer un compte d'utilisateur** : permet de créer un nouveau compte d'utilisateur par l'administrateur du système, en sélectionnant du même coup son niveau de permission.
13. **Mettre à jour un compte d'utilisateur** : permet de changer les informations associées à un compte d'utilisateur, y compris le niveau de permission, le nom d'utilisateur et le mot de passe. Dans le cas où ce dernier serait modifié, il devrait répondre aux mêmes exigences qu'à la fonction 16.
14. **Supprimer un compte d'utilisateur** : permet de supprimer le compte d'un usager.
15. **Gérer le niveau de permission d'un usager** : permet spécifiquement d'accorder un niveau ou l'autre de permission à un usager.
16. **Obliger l'utilisateur à changer son mot de passe** : afin de minimiser les risques liés à une fuite de mot de passe, l'utilisateur est obligé de changer régulièrement de mot de passe – et de ne pas réutiliser un même mot de passe que les 10 derniers. Cette fonction permet d'imposer cette règle.
17. **Créer un nouvel atelier** : permet de créer un nouvel atelier obligatoire qui sera imposé aux utilisateurs de l'organisme bénéficiaire.
18. **Mettre à jour un atelier** : permet de mettre à jour les informations associées à un atelier existant.
19. **Supprimer un atelier** : permet de supprimer un atelier, effaçant du même geste tous les groupes-cours et inscriptions qui y étaient associées.
20. **Créer un groupe-cours** : permet de créer un groupe-cours, qui sera une « représentation » d'un atelier, enseigné par un membre du personnel éducatif. Celui-ci est d'ailleurs autorisé à le créer pour lui-même.
21. **Mettre à jour un groupe-cours** : permet de modifier les informations liées à un groupe-cours, mais pas de saisir les renseignements liés aux utilisateurs ayant suivi l'atelier.
22. **Supprimer un groupe-cours** : permet de supprimer un groupe-cours en annulant l'ensemble des inscriptions qui y étaient liées.
23. **Inscrire un utilisateur à un atelier** : permet d'inscrire un utilisateur à un groupe-cours d'atelier – si cela est possible.
24. **Supprimer l'inscription d'un utilisateur à un atelier** : permet d'annuler l'inscription d'un utilisateur à un groupe-cours d'atelier.
25. **Saisir le résultat d'un utilisateur à un atelier** : permet au membre du personnel éducatif ayant dirigé le groupe-cours de saisir les résultats associés

aux utilisateurs qui y ont assistés. Le résultat sera stocké sous une notation de type lettre.

26. **Obtenir la liste des inscriptions à un atelier** : permet d'obtenir (et de manière implicite, d'imprimer) la liste des utilisateurs inscrits à un atelier, formatée de manière à faciliter le suivi lors de l'atelier même et comprenant une zone permettant de marquer d'un crochet la présence de l'utilisateur et la note accordée.
27. **Créer une activité** : permet de créer une nouvelle activité, ce qui peut être fait par un membre du personnel administratif ou éducatif, si ce dernier organise l'activité en question.
28. **Mettre à jour une activité** : permet de modifier les renseignements associés à une activité.
29. **Supprimer une activité** : permet de supprimer une activité ainsi que les inscriptions des utilisateurs y étant associés.
30. **Inscrire un utilisateur à une activité** : permet d'inscrire un utilisateur à une activité. Dans la version initiale du logiciel, seule une contrainte de places disponibles régit cette inscription – il n'est cependant pas impossible d'imaginer que cela puisse être détaillé lors d'une mise à jour ultérieure.
31. **Supprimer l'inscription d'un utilisateur à une activité** : permet de supprimer la participation d'un utilisateur à une activité.
32. **Obtenir la liste des inscriptions à une activité** : à l'instar de l'obtention de la liste d'inscriptions d'un groupe-cours d'atelier, permet d'obtenir la liste des utilisateurs inscrits à une activité. Il n'est cependant pas nécessaire de conserver une trace des présences ou un résultat quelconque.
33. **Générer un rapport** : module permettant de générer un nombre fini de rapports paramétrables ; il n'est accessible qu'aux usagers de niveau gestionnaire.
34. **Afficher les statistiques du système** : accessibles aux gestionnaires comme les rapports, les statistiques du système sont des informations quotidiennes de gestion générées par le logiciel. Contrairement aux rapports, elles ne sont pas paramétrables par un système de filtres.

2.3 Description des utilisateurs

La répartition des charges selon les groupes d'utilisateurs dans le contexte du SGBA est faite en fonction des opérations que chacun aura à effectuer ; cependant, ces dernières s'avèrent directement influencées par l'éducation préalable et l'expérience du personnel approprié. Ces usagers se distinguent par les « niveaux » de permissions qui sont :

- Le personnel administratif, qui sera essentiellement composé de travailleurs sociaux et de secrétaires à l'emploi de l'organisme de distribution alimentaire. Ces gens sont à la base du contact entre l'organisme et la clientèle, ce qui implique diverses qualités humaines et des compétences linguistiques, en raison du contexte multiculturel des bénéficiaires de ces services. Nous supposons que la plupart de ces employés ayant déjà manipulé un microordinateur, ils ne nécessiteront qu'un apprentissage des interfaces et possibilités du logiciel, en particulier pour les secrétaires. Une formation à l'interne s'avérera donc nécessaire, tout comme l'introduction à certains termes connexes au système (e.g. ce qu'est un réseau) afin de pouvoir le résumer aux utilisateurs curieux de voir l'employer manipuler le logiciel devant eux. Ils devront bénéficier d'une période d'entraînement sur des données de test pour assurer une performance correcte face à de vrais utilisateurs.
- Le personnel éducatif pourra être constitué d'employés, mais sera surtout formé de bénévoles, qui désirent faire bénéficier les utilisateurs de l'organisme de leurs connaissances particulières en organisant et animant un atelier ou une activité. Ils devront évidemment maîtriser suffisamment leur domaine « d'expertise » pour pouvoir l'enseigner clairement et simplement à une clientèle hétéroclite. Leurs origines professionnelles et études seront fort diverses. À l'instar du personnel administratif, ils pourront bénéficier d'une formation à l'interne, bien qu'ils ne manipulent que peu de fonctions et ne souffrent pas du stress d'exercer leur art aux vues et sues des utilisateurs.
- Les gestionnaires seront les professionnels chargés de gérer l'organisme et de faire les démarches appropriées auprès de compagnies et d'organismes subventionnaires. Leurs formations professionnelles dépassent les limites de ce document. Ils devront être formés aux modules de rapports et de statistiques, et surtout à la manière dont le générateur de rapports fonctionnera.
- Finalement, l'administrateur sera le seul membre du personnel (ou bénévole) de l'organisme de distribution alimentaire dont le rôle nécessitera des connaissances réelles en informatique, car il sera en charge de la configuration du SGBA, en plus de l'installation physique de l'équipement. Éventuellement, il pourra personnaliser les règles d'affaires du logiciel au moyen d'extensions. L'administrateur sera appelé à agir en cas de problème avec l'équipement, ainsi que pour s'occuper des sauvegardes de la base de données.

2.4 Contraintes d'ordre général

2.4.1 Règlements des organismes gouvernementaux, syndicaux ...

Les aspects légaux devant être considérés dans le cadre du SGBA seront essentiellement de l'ordre de la conservation d'informations personnelles sur un support électronique, tant de la nature des données collectées que des mesures mises en place pour assurer en assurer la protection contre le vol et la contrefaçon.

« Si la collecte de renseignements personnels est nécessaire, elle doit être réduite au minimum. L'utilisation ou la divulgation de renseignements personnels devrait également être réduite au minimum. Les renseignements personnels ne devraient être recueillis, utilisés ou divulgués qu'avec le consentement éclairé de la personne physique. »³

Un autre aspect légal devant être exploré concernera les mesures de chiffrement des données, qui devront respecter les normes en vigueur selon la législation canadienne.

2.4.2 Limitations du matériel

Les limitations du matériel ne représentent guère une contrainte importante au SGBA, en raison de sa petite taille relative, d'autant plus qu'il s'agit d'une application distribuée (c'est-à-dire des clients se connectant à une base de donnée centrale) prévue pour opérer chacun de ses éléments à une courte distance l'un de l'autre. Cependant, il n'est pas impossible qu'un organisme décide d'entreposer ses données à distance, ce qui ajoutera une limitation du point de vue de l'accès Internet absents d'un contexte de réseau local.

En une phrase, bien que prévu pour être exécuté sur un réseau local ou métropolitain, et malgré le fait que ces réseaux puissent en général fournir un débit important, il serait néanmoins nécessaire de conserver à l'esprit que la distance entre les clients et le serveur représente une limitation inhérente au principe même qui régit la répartition des couches du SGBA.

Un autre aspect qu'il sera nécessaire d'envisager sera l'état même du matériel qui sera utilisé pour faire fonctionner les différentes entités composant le système. En effet, l'état des finances de la majorité des organismes bénéficiaires potentiels ne permettant guère d'envisager l'achat d'équipement informationnel, la dépendance de ces derniers envers les dons du public réduisent les chances de bénéficier d'une plateforme permettant de réduire les risques de pannes matérielles.

2.4.3 Interfaces à d'autres applications

³ <http://e-com.ic.gc.ca/epic/site/ecic-ceac.nsf/fr/gv00244f.html>

Les contraintes du SGBA par rapport à d'autres applications sont pour pratiquement inexistantes, étant donné que celui-ci ne partage aucune information avec aucun autre logiciel. Nous pouvons cependant noter la connexion que celui-ci devra établir vers la base de données Oracle, ce qui imposera l'utilisation du pilote JDBC pour standardiser les échanges entre cet entrepôt et le système que nous désirons développer.

2.4.4 Opérations en parallèle

Le SGBA doit être conçu de manière à permettre l'accès simultané de plusieurs clients vers un même entrepôt de données, tant au niveau de l'authentification que de la vaste majorité des opérations, ce qui ne va pas sans imposer de contraintes au niveau du développement (e.g. verrous sur la base de données au niveau des dossiers en traitement, limiter l'accès à certaines opérations jugées trop chargées pour les heures d'ouverture de l'organisme).

Le système possédera également une composante de sécurité qui fermera automatiquement les sessions inactives d'utilisateurs connectés depuis trop longtemps, et signalera à l'administrateur les situations jugées suspectes (par exemple, des tentatives d'ouvrir une session si un même usager est déjà connecté) et potentiellement hasardeuses.

2.4.5 Fonctions d'audit

L'ensemble des opérations effectuées et affectant directement les données seront stockées, et ce autant pour ce qui a trait à la consultation que la manipulation des informations, ce qui permettra de détecter les risques de sécurité liés au vol interne de données confidentielles ou autres délits comparables.

2.4.6 Fonctions de contrôle

Les contraintes au niveau des fonctions de contrôle concernent essentiellement les connexions courantes à la base de données, qui se font au moyen des outils intégrés au SGBD. Il serait intéressant de munir la machine sur laquelle l'entrepôt de données est installé d'un registre d'adresses IP et MAC des ordinateurs y ayant accédés. Cependant, étant donné la nature locale du logiciel, cela pourrait s'avérer inutile, surtout advenant le cas où les machines exploitant le SGBA ne disposaient pas d'un accès à Internet.

2.4.7 Exigences de langages de programmation

Le langage que nous avons choisi pour développer le SGBA est Java, version 1.5 et ultérieures, en raison de sa fiabilité, du nombre sans cesse croissant d'applications et de solutions logicielles disponibles sur cette plateforme, de sa portabilité (qui permettrait

à l'organisme bénéficiaire de l'installer sous tout système d'exploitation disposant d'une machine virtuelle Java supportant Java 1.5), mais surtout en raison du nombre de développeurs aptes à utiliser cette technologie (i.e. aptes à supporter l'organisme en question advenant le cas où celui-ci désirerait un développement particulier). La gratuité de cette plateforme est également d'un intérêt certain.

L'utilisation d'une base de données Oracle XE « Express Edition » (version limitée, mais gratuite) nous apporte également un SGBD éprouvé, et cela sans aucun frais, pour lequel une large documentation existe et s'avère facile à consulter.

Bien qu'il sera nécessaire de posséder certaines compétences informatiques pour en effectuer l'installation et l'administration, nous croyons que l'utilisation d'outils standards tels que ceux que nous avons choisis faciliteront l'intégration du logiciel, peu importe le système d'exploitation choisi, et que les connaissances requises seront à la portée du technicien en informatique typique, disposant d'une documentation offerte à tous sur Internet par leurs fournisseurs⁴.

2.4.8 Protocoles de communication

Si la base de données est installée sur le même site que l'ensemble des stations clientes, aucun accès Internet ne sera nécessaire ; que ce soit le cas ou non, le réseau devra supporter le protocole TCP/IP, sur lequel le protocole OCI (Oracle Call Interface) sera utilisé, mais cette utilisation sera totalement transparente, en raison de l'utilisation par le composant de persistance des données du pilote JDBC pour l'accès à la base de données.

2.4.9 Exigences de fiabilité

Le système devra être totalement opérationnel lors des heures d'ouverture de l'organisme bénéficiaire, puisque celui-ci se trouvera dans une situation de dépendance relative par rapport au logiciel qu'il utilise. Il n'apparaît pas nécessaire que le SGBA doive fonctionner en permanence, en raison de l'absence d'une composante Web et des besoins d'entretien liés à son exploitation.

2.4.10 Aspects critiques de l'application

Hormis la connectivité réseau entre les composantes, aucun aspect n'apparaît critique dans le cadre du SGBA : l'impossibilité de se connecter à la base de données entraîne une impossibilité totale d'exploiter le logiciel.

2.4.11 Considérations de sûreté et de sécurité

⁴ En l'occurrence, Sun Microsystems et Oracle.

Les contraintes de sécurité imposées dans le cadre du SGBA reposeront essentiellement sur l'authentification des usagers responsables des opérations effectuées sur le système, et le respect scrupuleux de règles élémentaires liées à leurs comptes (e.g. ne pas laisser autrui effectuer des opérations en leur nom, ou divulguer leur mot de passe à quiconque). Le meilleur système de sécurité disponible dans les circonstances ne pourrait pas contrer pareilles pratiques, advenant le cas où elles seraient monnaie courante là où le SGBA sera implanté.

La sécurité devra également s'appliquer aux équipements informatiques, comme nous l'avons déjà expliqué précédemment.

2.5 Hypothèses et dépendances

Il va de soi que le passage d'un système papier et/ou partiellement informatisé au SGBA aura un impact immédiat sur les procédures et habitudes des employés et bénévoles oeuvrant au sein d'un organisme bénéficiaire. La transition à effectuer sera fortement dépendante de l'organisme, en particulier de la rigueur de sa répartition des tâches, étant donné que le SGBA sera très strict au niveau de cette dernière. Une période de formation des employés devra être aménagée, et il ne serait pas superflu de faire cohabiter l'ancien système et le SGBA en parallèle pour un laps de temps à déterminer, avant de cesser complètement les opérations précédentes, ne serait-ce que pour valider l'adaptation du système à l'organisme, la validité de la configuration des règles d'affaires et la capacité des usagers à manipuler le système avec une aisance raisonnable.

Un autre aspect à considérer sera la transition de l'ancien système vers le nouveau, qui pourrait nécessiter un temps proportionnel au nombre d'utilisateurs actuels et au support précédent (il sera plus rapide de transférer les données d'une source Excel que de les saisir entièrement à la main ou par reconnaissance optique de caractères). Dans le cas où le volume de données à convertir sera trop grand pour envisager une transition manuelle, il serait imaginable de mettre au point un module capable d'extraire les données de leur source actuelle et de les convertir en un format reconnu par le SGBA. Cependant, pareil module serait par définition indépendant du système, puisqu'il ne servirait qu'à une seule tâche, celle-ci étant totalement détachée du cadre normal de l'exploitation de notre logiciel.

L'utilisation du langage Java procure une indépendance du logiciel par rapport à une plateforme particulière, ce qui permet d'imaginer un déploiement sur plusieurs stations clientes, obtenues gratuitement ou à prix modique, dont les systèmes d'exploitations s'avèreraient pour le moins hétérogènes. Ce déploiement se ferait sans heurs, en autant qu'une version de la machine virtuelle Java puisse être installée sur l'appareil en question et que celui-ci soit capable d'effectuer des échanges réseaux au moyen du protocole TCP/IP. Au moment de rédiger ces lignes, l'immense majorité des ordinateurs personnels sont très facilement aptes à remplir ces exigences.

2.6 Répartition des exigences

La première version du SGBA ne comprendra pas une implantation de l'ensemble des fonctions que nous lui souhaitons, qui devraient faire l'objet d'un développement ultérieur :

- La version suivante contiendra une implantation d'un moteur de règles basé sur une machine virtuelle, dont la tâche sera d'interpréter des informations au format XML afin d'alimenter le gestionnaire de sécurité avec les renseignements nécessaires à autoriser ou refuser en temps réel les opérations effectuées sur le SGBA, en fonction des paramètres spécifiés dans les fichiers de règles. Il serait également souhaitable d'être en mesure de mettre à la disposition de l'administrateur le moyen de définir ses propres règles à travers un langage balisé simple, ce qui implique une grande généralité dans l'architecture du logiciel, en particulier pour ce qui relève du domaine des composantes de sécurité.
- Un mécanisme de fonctionnement local, en cas de panne d'accès à la base de données centrale, pourrait aussi faire l'objet d'un développement ultérieur. Déjà compris dans certaines applications commerciales⁵, cette sécurité s'avérera d'un grand secours lors de pareilles circonstances, permettant aux clients de fonctionner de manière autonome, puis de mettre à jour la centrale lorsque le fonctionnement de la liaison avec cette dernière est rétabli.
- La génération des rapports ne sera pas entièrement développée dans la première version, c'est-à-dire que malgré le fait que l'ensemble des rapports souhaitables dans l'immédiat seront accessibles, la création de rapports personnalisés (en choisissant les sources de données, en filtrant sur certains critères) à partir de rien ne sera pas possible. Cette capacité devra être déléguée à une version ultérieure, car elle ne s'avère pas indispensable et nous estimons qu'elle retarderait considérablement la livraison du module de statistiques et de rapports, en raison de sa grande complexité.
- L'accès aux informations d'audit n'est pas prévu pour la première version. Cette composante s'avérant fortement désirable dans une perspective d'exploitation sécurisée du logiciel, elle devrait faire l'objet du plus proche développement ultérieur. La version initiale du SGBA, toutefois, emmagasinerait l'ensemble de ces données, qui ne pourront malgré tout être consultées qu'à partir de l'interface propre au SGBD, ce qui ne devrait pas dépasser les compétences techniques de l'administrateur système typique.
- Il serait fort souhaitable de permettre l'affichage des interfaces usagers en plusieurs langues, en raison du contexte socioculturel dans lequel il sera exploité. La première version, toutefois, ne sera disponible qu'en français, ce qui ne

⁵ Le progiciel Sirsi Unicorn permet à une bibliothèque de prêt de fonctionner de manière autonome, avec des fonctionnalités réduites, en cas de perte de contact prolongée avec le système central.

représente pas un handicap en cas d'exploitation au Québec, celle-ci y étant la langue de travail.

3. Description détaillée

3.1 Interfaces externes

3.1.1 Ajouter un dossier d'utilisateur

ID :	UC1		
Nom :	Ajouter un dossier d'utilisateur		
Acteurs :	Personnel administratif		
Objectif :	Permettre de créer et d'ajouter un nouveau dossier d'utilisateur à la liste des dossiers.		
Résumé :	Suite à une démarche de ce dernier auprès de l'organisme d'aide alimentaire, un membre du personnel administratif souhaite créer un nouveau dossier d'utilisateur pour lui permettre de bénéficier des services de la banque.		
Type :	Importance (primaire), Modélisation (essentielle)		
Références :	CO01		
Pré-condition :	L'acteur est authentifié et a le droit de créer un dossier d'utilisateur.		
Post-condition :	Le nouveau dossier d'utilisateur est ajouté à la liste des dossiers.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
1	Sollicite la création d'un nouvel utilisateur.		
		2	Présente l'écran de saisie d'un nouvel utilisateur
3	Remplit les champs appropriés de l'écran de saisie		
4	Demande l'enregistrement du nouvel utilisateur		
		5	Valide les informations saisies
		6	Crée un nouveau dossier d'utilisateur dans la BD
		7	Confirme le succès de l'enregistrement du nouveau dossier
Exceptions :	E5. a. Un champ obligatoire est manquant b. Le format d'un champ en particulier est incorrect c. Un dossier existe déjà pour ce nom d'utilisateur, cette adresse et cette date de naissance E8. a. Il n'y a plus suffisamment d'espace disque pour créer le dossier		

3.1.2 Mettre à jour un dossier d'utilisateur

ID :	UC2		
Nom :	Mettre à jour un dossier d'utilisateur		
Acteurs :	Personnel administratif		
Objectif :	Permettre de mettre à jour un dossier d'utilisateur existant.		
Résumé :	Un membre du personnel administratif souhaite mettre à jour le dossier d'un utilisateur pour refléter un changement qui lui a été signalé.		
Type :	Importance (primaire), Modélisation (essentielle)		
Références :	CO02, Rechercher un utilisateur		
Pré-condition :	L'acteur est authentifié et a le droit de modifier un dossier d'utilisateur. Le dossier d'utilisateur à modifier existe.		
Post-condition :	Le dossier d'utilisateur est mis à jour dans la liste des dossiers.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Sollicite la modification d'un dossier d'utilisateur.		
		<u>2</u>	Présente l'écran de saisie de recherche de dossier d'utilisateur
<u>3</u>	Saisit les critères appropriés de recherche		
		<u>4</u>	Limite les utilisateurs aux critères sélectionnés
<u>5</u>	Sélectionne un dossier d'utilisateur à modifier		
		<u>6</u>	Affiche le dossier de l'utilisateur tel qu'enregistré dans la BD
<u>7</u>	Modifie les informations d'un ou de plusieurs champs		
<u>8</u>	Demande d'enregistrer les modifications apportées au dossier		
		<u>9</u>	Valide les informations saisies
		<u>10</u>	Enregistre les modifications au dossier dans la BD
Exceptions :	E4 a. Aucun dossier d'utilisateur n'existe dans la liste E9. a. Un champ obligatoire est manquant b. Le format d'un champ en particulier est incorrect		

3.1.3 Supprimer un dossier d'utilisateur

ID :	UC3		
Nom :	Supprimer un dossier d'utilisateur		
Acteurs :	Personnel administratif		
Objectif :	Permettre de supprimer un dossier d'utilisateur existant.		
Résumé :	Un membre du personnel administratif souhaite supprimer un dossier d'utilisateur sans laisser de trace dans le système.		
Type :	Importance (secondaire), Modélisation (essentielle)		
Références :	CO03, Rechercher un utilisateur		
Pré-condition :	L'acteur est authentifié et a le droit de supprimer un dossier d'utilisateur. Le dossier d'utilisateur à supprimer existe.		
Post-condition :	Le dossier d'utilisateur est mis à jour dans la liste des dossiers. Les inscriptions de l'utilisateur ne figurent plus à la liste des groupes-cours d'activités. Les inscriptions de l'utilisateur ne figurent plus aux activités. L'historique des visites de l'utilisateur est supprimé. L'historique des crédits de l'utilisateur est supprimé.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
1	Sollicite la suppression d'un dossier d'utilisateur.		
		2	Présente l'écran de saisie de recherche de dossier d'utilisateur
3	Saisit les critères appropriés de recherche		
		4	Limite les utilisateurs aux critères sélectionnés
5	Sélectionne un dossier d'utilisateur à supprimer		
		6	Affiche le dossier de l'utilisateur tel qu'enregistré dans la BD
7	Demande de supprimer le dossier		
		8	Demande une confirmation de la suppression du dossier
9	Confirme la suppression du dossier de l'utilisateur		
		10	Supprime les entrées liées à l'utilisateur dans la BD
Exceptions :	E4 a. Aucun dossier d'utilisateur n'existe dans la liste E9. a. Annule la suppression du dossier* (*) Dans ce cas, aucune modification n'est apportée aux entrées liées au dossier de l'utilisateur dans la BD		

3.1.4 Réinscrire un utilisateur

ID :	UC4		
Nom :	Réinscrire un utilisateur		
Acteurs :	Personnel administratif		
Objectif :	Permettre de réinscrire un utilisateur dont le dossier est devenu inactif ou marqué comme étant "à réinscrire".		
Résumé :	Un membre du personnel administratif souhaite enregistrer la réinscription d'un utilisateur rendue nécessaire par son désir de recourir à nouveau aux services de l'organisme alimentaire deux ans après sa dernière visite (inactif) ou par la nécessité de vérifier ses coordonnées (réinscription annuelle).		
Type :	Importance (secondaire), Modélisation (essentielle)		
Références :	CO04, Rechercher un utilisateur, Modifier un utilisateur		
Pré-condition :	L'acteur est authentifié et a le droit d'enregistrer une réinscription d'utilisateur Le dossier d'utilisateur à réinscrire existe.		
Post-condition :	Le dossier d'utilisateur est mis à jour dans la liste des dossiers.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
1	Sollicite la réinscription d'un dossier d'utilisateur.		
		2	Présente l'écran de saisie de recherche de dossier d'utilisateur
3	Saisit les critères appropriés de recherche		
		4	Limite les utilisateurs aux critères sélectionnés
5	Sélectionne un dossier d'utilisateur à modifier		
		6	Affiche le dossier de l'utilisateur tel qu'enregistré dans la BD
7	Vérifie si les renseignements de l'utilisateur sont à jour		
8	Demande d'enregistrer la réinscription		
		9	Détermine le statut d'utilisateur à enregistrer
		10	Enregistre les modifications au dossier dans la BD
Exceptions :	E4 a. Aucun dossier d'utilisateur n'existe dans la liste E6. a. L'utilisateur n'a pas besoin d'être réinscrit		

3.1.5 Rechercher un dossier d'utilisateur

ID :	UC5		
Nom :	Rechercher un dossier d'utilisateur		
Acteurs :	Personnel administratif		
Objectif :	Permettre de rechercher un dossier d'utilisateur en vertu de critères prédéfinis.		
Résumé :	Un membre du personnel administratif souhaite rechercher un dossier d'utilisateur afin de le consulter.		
Type :	Importance (primaire), Modélisation (essentielle)		
Références :	CO05		
Pré-condition :	L'acteur est authentifié et a le droit de rechercher un dossier d'utilisateur.		
Post-condition :			
Interactions :	(Ce cas d'utilisation peut être invoqué indépendamment, mais il sera davantage sollicité dans le cadre d'autres cas que par lui-même.)		
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Sollicite la recherche d'un dossier d'utilisateur.		
		<u>2</u>	Présente l'écran de saisie de recherche de dossier d'utilisateur
<u>3</u>	Saisit les critères appropriés de recherche		
		<u>4</u>	Limite les utilisateurs aux critères sélectionnés
<u>5</u>	Sélectionne un dossier d'utilisateur à traiter		
		<u>6</u>	Traite le dossier de l'utilisateur sélectionné selon les circonstances
Exceptions :	E3 a. Une valeur incorrecte est saisie dans un champ de recherche E4. a. Aucun dossier d'utilisateur n'existe E6. a. Aucun traitement n'est à effectuer (le cas d'utilisation est appelé seul)		

3.1.6 Afficher un dossier d'utilisateur

ID :	UC6		
Nom :	Afficher un dossier d'utilisateur		
Acteurs :	Personnel administratif		
Objectif :	Permettre d'afficher un dossier d'utilisateur existant en lecture seule.		
Résumé :	Un membre du personnel administratif souhaite consulter le dossier d'un utilisateur, sans pouvoir le modifier.		
Type :	Importance (primaire), Modélisation (essentielle)		
Références :	CO06, Rechercher un utilisateur		
Pré-condition :	L'acteur est authentifié et a le droit d'afficher un dossier d'utilisateur. Le dossier d'utilisateur à afficher existe.		
Post-condition :			
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
1	Sollicite la modification d'un dossier d'utilisateur.		
		2	Présente l'écran de saisie de recherche de dossier d'utilisateur
3	Saisit les critères appropriés de recherche		
		4	Limite les utilisateurs aux critères sélectionnés
5	Sélectionne un dossier d'utilisateur à modifier		
		6	Affiche le dossier de l'utilisateur tel qu'enregistré dans la BD
Exceptions :	E4 a. Aucun dossier d'utilisateur n'existe dans la liste		

3.1.7 Rembourser un crédit

ID :	UC7		
Nom :	Rembourser un crédit		
Acteurs :	Personnel administratif		
Objectif :	Permettre de retirer un crédit figurant à un dossier d'utilisateur.		
Résumé :	Un membre du personnel administratif souhaite enregistrer le remboursement d'un ou crédit préalablement accordé à un utilisateur.		
Type :	Importance (primaire), Modélisation (essentielle)		
Références :	CO07, Rechercher un utilisateur		
Pré-condition :	L'acteur est authentifié et a le droit d'enregistrer un remboursement de crédit Le dossier d'utilisateur à modifier existe. Le dossier d'utilisateur contient au moins un crédit à rembourser.		
Post-condition :	Le dossier d'utilisateur est mis à jour dans la liste des dossiers.		
Interactions :			
	<u>Acteurs</u>		<u>Systeme</u>
1	Sollicite la modification d'un dossier d'utilisateur.		
		2	Présente l'écran de saisie de recherche de dossier d'utilisateur
3	Saisit les critères appropriés de recherche		
		4	Limite les utilisateurs aux critères sélectionnés
5	Sélectionne un dossier d'utilisateur à modifier		
		6	Affiche le dossier de l'utilisateur tel qu'enregistré dans la BD
7	Choisit le nombre de crédits à rembourser		
8	Demande d'enregistrer le remboursement		
		9	Valide le nombre de remboursements saisis
		10	Enregistre les modifications au dossier dans la BD
Exceptions :	E4 a. Aucun dossier d'utilisateur n'existe dans la liste E6. a. Le dossier ne contient aucun crédit à rembourser E9. a. Le nombre de crédits saisis est invalide		

3.1.8 Enregistrer la visite d'un utilisateur

ID :	UC8	
Nom :	Enregistrer la visite d'un utilisateur	
Acteurs :	Personnel Administratif et Utilisateur	
Objectif :	Permettre au membre du personnel administratif d'enregistrer une visite d'un utilisateur.	
Résumé :	Un utilisateur se présente dans le cadre d'une distribution alimentaire, un membre du personnel administratif de la banque alimentaire enregistre la visite à des fins de suivi et de statistiques.	
Type :	Importante (primaire), Modélisation (essentielle)	
Références :	CO08; Invoque : Enregistrer crédit utilisateur, rembourser crédit utilisateur.	
Pré-condition :	L'acteur est authentifié et a le droit d'enregistrer une visite d'un utilisateur. Le dossier de l'utilisateur est accessible et en consultation.	
Post-condition :	L'historique des visites de l'utilisateur est à jour.	
Interactions :		
	<u>Acteurs</u>	<u>Systeme</u>
<u>1</u>	Demande d'enregistrer la visite d'un utilisateur.	
		<u>2</u> Vérifie si le statut de l'utilisateur lui permet d'enregistrer une visite.
		<u>3</u> Vérifie si l'utilisateur n'a pas excédé les demandes d'aide permises pour la période courante.
		<u>4</u> Demande confirmation d'enregistrement de la visite.
<u>5</u>	Approuve l'enregistrement de la visite.	
		<u>6</u> Enregistre la visite dans la BD.
Exceptions :	<p>E2.</p> <p>a. Le statut de l'utilisateur ne lui permet pas d'enregistrer une visite*. I. Un message avertit l'acteur que le statut de l'utilisateur ne lui permet pas d'enregistrer une visite et suggère de régulariser son statut avec le use-case réinscrire dossier utilisateur.</p> <p>E3.</p> <p>a. L'utilisateur excède le nombre de demandes qui lui sont permises*.</p> <p>E5.</p> <p>a. Annule l'enregistrement de la visite*.</p> <p>(*) Dans ces cas, aucune modification n'est apportée à l'historique des visites de l'utilisateur dans la BD</p>	

3.1.9 Changer le statut des utilisateurs pour « à réinscrire »

ID :	UC9		
Nom :	Changer le statut des utilisateurs pour « à réinscrire »		
Acteur :	Temporel (indirectement, utilisateurs)		
Objectif :	Le système change le statut de tous les utilisateurs pour « à réinscrire », afin de forcer une vérification générale des coordonnées.		
Résumé :	À l'atteinte d'un moment défini par l'administrateur, le système change le statut des utilisateurs pour « à réinscrire » afin de confirmer les adresses de tous les utilisateurs.		
Type :	Importance (secondaire), Modélisation (sans impact)		
Références :	CO09		
Pré-condition :	Le système atteint la date définie pour le changement de statut.		
Post-condition :	Le statut de tous les utilisateurs conditionnels et actifs sont changés pour à réinscrire.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
		1	Détecte que la date système a atteint la date définie pour le changement de statut.
		2	Enregistre le changement de statut pour tous les dossiers conditionnels et actifs.
Exceptions :			

3.1.10 Enregistrer un crédit à un utilisateur

ID :	UC10		
Nom :	Enregistrer un crédit à un utilisateur		
Acteurs :	Personnel administratif et utilisateur		
Objectif :	Permettre à un membre du personnel administratif d'enregistrer un crédit demandé par l'utilisateur.		
Résumé :	Un utilisateur se présente pour une visite, un membre du personnel administratif tente avec succès d'enregistrer une visite, mais l'utilisateur n'est pas en mesure de fournir sa contribution en argent à la visite, le membre du personnel administratif tente donc de lui faire crédit.		
Type :	Importance (primaire), Modélisation (désirable)		
Références :	CO10		
Pré-condition :	L'acteur est authentifié et a le droit d'enregistrer un crédit. Le dossier d'utilisateur à qui enregistrer un crédit existe et se réfère à une visite.		
Post-condition :	La liste des crédits de l'utilisateur est à jour.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
1	Demande d'enregistrer un crédit.		
		2	Vérifie si l'utilisateur a le droit d'obtenir un crédit.
		3	Enregistre le crédit de l'utilisateur dans la BD.
Exceptions :	E2. a. Le maximum de crédits permis est déjà atteint. I. Un message avertit que l'utilisateur n'a plus de crédits disponibles et cesse le use-case.		

3.1.11 Consulter un compte d'utilisateur

ID :	UC11		
Nom :	Consulter un compte d'utilisateur		
Acteur :	Administrateur		
Objectif :	Permettre à l'administrateur du système de consulter un compte d'utilisateur.		
Résumé :	L'administrateur veut consulter un compte d'utilisateur pour une raison quelconque.		
Type :	Importance (primaire), Modélisation (essentielle)		
Références :	CO11; Invoqué par Mettre à jour un d'utilisateur .		
Pré-condition :	L'administrateur du système est authentifié.		
Post-condition :			
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Soumet le numéro d'employé d'un usager.		
		<u>2</u>	Valide le numéro d'employé et vérifie l'existence de l'utilisateur.
		<u>3</u>	Charge le dossier de l'utilisateur à partir de la BD.
<u>Exceptions :</u>	<p>E2.</p> <ul style="list-style-type: none"> a. Le numéro d'employé est invalide. <ul style="list-style-type: none"> I. Un message avertit que le numéro d'employé est invalide retourne à l'étape 1. b. Le numéro d'employé est valide mais l'utilisateur n'existe pas. <ul style="list-style-type: none"> I. Le système affiche un message : voulez-vous créer un nouvel usager ? L'administrateur accepte, le système invoque le use-case Créer un compte d'utilisateur. II. L'administrateur refuse, le système retourne à E1. 		

3.1.12 Créer un compte d'utilisateur

ID :	UC12		
Nom :	Créer un compte d'utilisateur		
Acteur :	Administrateur		
Objectif :	Permettre à l'administrateur du système d'enregistrer un nouvel utilisateur.		
Résumé :	Un nouvel utilisateur est autorisé par l'organisme à utiliser le système. L'administrateur s'informe des coordonnées de l'utilisateur et lui crée un compte dans le système.		
Type :	Importance (primaire), Modélisation (essentielle)		
Références :	CO12		
Pré-condition :	L'administrateur du système est authentifié. On supposera que les données concernant l'utilisateur ont déjà été authentifiées et sont considérées fiables		
Post-condition :	Le nouvel utilisateur est enregistré dans le système.		
Interactions :			
	Acteurs		Système
1	Soumet le numéro d'employé du nouvel utilisateur.		
		2	Valide le numéro d'employé et vérifie l'existence du dossier de l'utilisateur.
		3	Crée un nouveau compte d'utilisateur.
4	Saisit les coordonnées de l'utilisateur.		
		5	Valide les coordonnées.
6	Saisit le nom d'utilisateur le mot de passe.		
		7	Valide la forme du nom d'utilisateur « login » et du mot de passe.
		8	Demande la confirmation de création du nouveau compte.
9	Confirme la création du nouvel utilisateur.		
		10	Enregistre le nouveau compte d'utilisateur dans la BD.
Exceptions :	E2. a. L'utilisateur existe déjà. a. Invoquer le use-case mettre à jour un compte d'utilisateur. b. Le numéro d'employé est invalide. I. Un message avertit que le numéro d'employé est invalide et retourne à l'étape 1.		

3.1.13 Mettre à jour un compte d'utilisateur

ID :	UC13		
Nom :	Mettre à jour un compte d'utilisateur		
Acteur :	Usager et administrateur		
Objectif :	Permettre à l'administrateur du système de modifier un usager.		
Résumé :	Un usager se présente à l'administrateur pour modifier ses coordonnées personnelles ou pour changer son mot de passe.		
Type :	Importance (secondaire), Modélisation (essentielle)		
Références :	CO13 ; Invoque Consulter un compte d'utilisateur .		
Pré-condition :	L'administrateur du système est déjà authentifié. Le compte de l'utilisateur existe.		
Post-condition :	Le compte de l'utilisateur est à jour dans le système.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Demande la mise à jour d'un compte d'utilisateur.		
		<u>2</u>	Invoque Consulter un compte d'utilisateur et affiche l'état actuel du compte.
<u>3</u>	Demande l'édition des coordonnées de l'utilisateur.		
		<u>4</u>	Valide les coordonnées saisies.
1. <u>5</u>	Saisit les nouvelles coordonnées.		
2. <u>6</u>	Modifier le mot de passe d'un usager.		
3.		<u>7</u>	Valide le mot de passe et demande la confirmation d'enregistrement des modifications.
4. <u>8</u>	Confirme qu'il désire enregistrer les modifications.		
<u>5.</u>		<u>9</u>	Enregistre dans la BD les modifications apportées au compte d'utilisateur.
Exceptions :	E3. a. Aucun usager n'est retourné par l'invocation. I. Fin du use-case avec message significatif.		

3.1.14 Supprimer un compte d'utilisateur

ID :	UC14		
Nom :	Supprimer un compte d'utilisateur		
Acteur :	Administrateur		
Objectif :	Permettre à l'administrateur du système de supprimer un compte d'utilisateur.		
Résumé :	L'administrateur veut supprimer un usager qui ne travaille plus pour la banque alimentaire ou qui n'a plus besoin d'accéder au système.		
Type :	Importante (primaire), Modélisation (désirable)		
Références :	CO14 ; Invoque : Consulter un compte d'utilisateur .		
Pré-condition :	L'administrateur du système est authentifié. Le dossier de l'utilisateur est déjà en consultation.		
Post-condition :	Le dossier de l'utilisateur est désactivé dans la BD.		
Interactions :			
	Acteurs		Système
1	Demande la suppression du compte d'utilisateur.		
		2	Demande confirmation pour supprimer l'utilisateur.
3	Confirme la suppression.		
		4	Marque le compte d'utilisateur comme étant supprimé.
Exceptions :	E3. a) L'administrateur décide de ne pas supprimer le compte d'utilisateur. I. Le système annule l'opération ; retour au use-case « Consulter un compte d'utilisateur ».		

3.1.15 Gérer le niveau de permission d'un usager

ID :	UC15		
Nom :	Gérer le niveau de permission d'un usager		
Acteur :	Usager et administrateur		
Objectif :	Permettre à l'administrateur du système de gérer les permissions des usagers.		
Résumé :	L'administrateur assigne le niveau de permission associé à un compte d'utilisateur à partir de la liste déterminée dans le système.		
Type :	Importance (primaire), Modélisation (essentiel)		
Références :	CO15 ; Invoque : Consulter un compte d'utilisateur .		
Pré-condition :	L'administrateur du système est déjà authentifié. Le compte de l'utilisateur est déjà en consultation.		
Post-condition :	Le niveau de permission de l'utilisateur est à jour.		
Interactions :			
	Acteurs		Système
1	Demande l'édition de la permission d'un usager.		
		2	Présente la liste des permissions possibles.
3	Sélectionne le niveau de permission à attribuer.		
4	Demande l'enregistrement des modifications.		
		5	Enregistre le changement de permission au niveau de la BD.
Exceptions :	E4. a. Annulation des modifications		

3.1.16 Obliger l'utilisateur à changer son mot de passe

ID :	UC16		
Nom :	Obliger l'utilisateur à changer son mot de passe		
Acteur :	Usager (tous niveaux de permissions)		
Objectif :	Un usager doit changer son mot de passe à tous les 4 mois. L'utilisateur doit saisir son ancien mot de passe suivi de son nouveau. Le système doit vérifier que son nouveau mot de passe n'est pas un mot de passe trop simple et qu'il ne fait pas partie des 10 derniers mots de passe utilisés par l'utilisateur.		
Résumé :	Un usager du système s'est authentifié. Il reçoit un message comme quoi il doit changer son mot de passe, puisque le temps restant depuis le dernier changement est écoulé.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO16		
Pré-condition :	L'utilisateur est déjà authentifié.		
Post-condition :	Le dossier de l'utilisateur et la liste de ses anciens mots de passe sont à jour.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
		<u>1</u>	Détecte que le temps écoulé depuis le dernier changement de mot de passe est écoulé.
		<u>2</u>	Demande la saisie d'un nouveau mot de passe.
<u>3</u>	Saisit un mot de passe.		
		<u>4</u>	Valide la forme du mot de passe saisi.
		<u>5</u>	Valide que le mot de passe saisi n'est pas le mot de passe actuel et ne fait pas partie de la liste des 10 derniers
		<u>6</u>	Enregistre le changement de mot de passe dans la BD et met à jour l'historique.
		<u>7</u>	Confirme par message le changement de mot de passe.
<u>Exceptions :</u>	E4. a. Le mot de passe ne respecte pas les exigences de forme. E5. a. Le mot de passe correspond à celui qui a été utilisé en dernier. b. Le mot de passe a été utilisé auparavant.		

3.1.17 Créer un nouvel atelier

ID :	UC17		
Nom :	Créer un nouvel atelier		
Acteur :	Personnel administratif		
Objectif :	Création d'un nouvel atelier dans le système par un usager. L'atelier peut être directement entré avec les renseignements de base (nom, lieu, horaire).		
Résumé :	L'organisme souhaite imposer un nouvel atelier éducatif obligatoire. Un membre du personnel administratif est chargé de saisir ce nouvel atelier dans le système.		
Type :	Importance (primaire), Modélisation (essentiel)		
Références :	CO17 ; Créer un groupe-cours		
Pré-condition :	L'acteur est authentifié et a le droit de créer un nouvel atelier.		
Post-condition :	La liste des ateliers est à jour.		
Interactions :			
	Acteurs		Système
	1 Sollicite la création d'un nouvel atelier.		
		<u>2</u>	Présente l'écran de création d'un nouvel atelier.
	<u>3</u> Saisit les informations appropriées : nom, description de l'atelier.		
	<u>4</u> Demande l'enregistrement du nouvel atelier.		
		<u>5</u>	Valide les renseignements saisis.
		<u>6</u>	Enregistre le nouvel atelier dans la BD.
		<u>7</u>	Demande si l'acteur souhaite créer tout de suite un nouveau groupe-cours.
88	<u>8</u> Ne souhaite pas créer tout de suite un nouveau groupe-cours.		
Exceptions :	E5. a. Les renseignements saisis ne sont pas valides ou présents. E8. a. L'acteur souhaite créer tout de suite un groupe-cours associé à l'atelier. I. Le système invoque le use-case Créer un groupe-cours.		

3.1.18 Mettre à jour un atelier

ID :	UC18		
Nom :	Mettre à jour un atelier		
Acteur :	Personnel administratif		
Objectif :	Permettre la mise à jour des informations d'un atelier.		
Résumé :	L'utilisateur souhaite mettre à jour l'une des données d'un atelier (nom, description, etc.)		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO18 ; Invoque Créer un groupe-cours , Mettre à jour un groupe-cours et Supprimer un groupe-cours .		
Pré-condition :	L'utilisateur est authentifié et autorisé à mettre à jour un atelier. L'atelier est déjà en consultation.		
Post-condition :	L'atelier est à jour.		
Interactions :			
	Acteurs		Système
1	Demande l'édition d'un atelier.		
		2	Permet d'éditer les champs appropriés (e.g. nom, description).
3	Saisit les renseignements appropriés.		
4	Demande d'enregistrer les modifications.		
		5	Enregistre les modifications au niveau de la BD.
		6	Demande si l'acteur souhaite modifier aussi les groupes-cours associés à l'atelier.
7	Ne souhaite pas modifier les groupes-cours associés à l'atelier.		
Exceptions :	E5. a. Les renseignements saisis ne sont pas valides ou présents. E7. a. L'acteur souhaite modifier les groupes-cours associés à l'atelier. I. Le système exécute Créer un groupe-cours, Mettre à jour un groupe-cours ou Supprimer un groupe-cours, selon le cas.		

3.1.19 Supprimer un atelier

ID :	UC19		
Nom :	Supprimer un atelier		
Acteur :	Personnel administratif		
Objectif :	Permettre de marquer comme supprimé un atelier que l'on ne souhaite plus imposer aux utilisateurs.		
Résumé :	Un des ateliers obligatoires de l'organisme ne doit plus être imposé aux utilisateurs. Un membre du personnel administratif est chargé de le marquer comme supprimé.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO19 ; Invoque Supprimer un groupe-cours.		
Pré-condition :	L'acteur est authentifié et autorisé à supprimer un atelier.		
Post-condition :	La liste des ateliers est à jour. La liste des inscriptions liées aux groupe-cours de l'atelier n'existe plus. Le statut des utilisateurs est à jour.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Demande la suppression de l'atelier		
		<u>2</u>	Présente la liste des conséquences associées à la suppression de l'atelier.
<u>3</u>	Confirme la demande de suppression.		
		<u>4</u>	Vérifie qu'il n'a pas de groupe-cours associé à l'atelier.
		<u>5</u>	Change le statut des utilisateurs conditionnels n'ayant que cet atelier à compléter à actif.
		<u>6</u>	Marque l'atelier comme étant supprimé.
<u>Exceptions :</u>	E4. a. Un ou plusieurs groupe-cours sont associés à l'atelier. I. Le système invoque « Supprimer un groupe-cours » pour chacun de ceux-ci.		

3.1.20 Créer un groupe-cours

ID :	UC20		
Nom :	Créer un groupe-cours		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre de créer un groupe-cours (dans le cadre d'un atelier, une « représentation » de celui-ci par un enseignant) auquel les utilisateurs pourront être inscrits.		
Résumé :	Un membre du personnel éducatif souhaite offrir un atelier. Il crée un groupe-cours en spécifiant son lieu et son moment, pour permettre d'y inscrire des utilisateurs.		
Type :	Importance (primaire), Modélisation (essentiel)		
Références :	CO20		
Pré-condition :	L'acteur est authentifié et autorisé à créer un groupe-cours. L'atelier associé est déjà en consultation.		
Post-condition :	Le groupe-cours figure dans la liste des groupe-cours de l'atelier associé.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Sélectionne l'option « Créer un groupe-cours ».		
		<u>2</u>	Présente l'écran de création de groupe-cours.
<u>3</u>	Saisit le lieu, le moment et le membre du personnel éducatif associé* au groupe-cours.		
<u>4</u>	Demande l'enregistrement du nouveau groupe-cours.		
		<u>5</u>	Vérifie qu'il n'y a pas de conflit d'horaire pour le membre du personnel éducatif (et que l'horaire est valide dans le temps).
		<u>6</u>	Enregistre le nouveau groupe-cours dans la BD.
<u>Exceptions :</u>	E5. a. Le système détecte un conflit d'horaire entre le groupe-cours actuel et ceux déjà enregistrés. I. Retour à E3 après un message d'erreur significatif.		

* Dans le cas où l'acteur serait un membre du personnel éducatif, cette option ne peut être changée : il ne peut créer que des groupe-cours où il sera lui-même l'enseignant.

3.1.21 Mettre à jour un groupe-cours

ID :	UC21		
Nom :	Mettre à jour un groupe-cours		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre de modifier un groupe-cours (dans le cadre d'un atelier, une « représentation » de celui-ci par un enseignant) auquel les utilisateurs pourront être inscrits.		
Résumé :	Un membre du personnel éducatif souhaite offrir un atelier. Il crée un groupe-cours en spécifiant son lieu et son moment, pour permettre d'y inscrire des utilisateurs. Cependant, il s'avère plus tard nécessaire de modifier les informations y figurant. L'acteur sélectionne l'atelier et choisit le groupe-cours. Il peut ensuite le modifier.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO21		
Pré-condition :	L'acteur est authentifié et autorisé à modifier un groupe-cours. L'atelier associé est déjà en consultation et le groupe-cours existe.		
Post-condition :	La liste de groupe-cours de l'atelier est à jour.		
Interactions :			
	<u>Acteurs</u>		<u>Systeme</u>
<u>1</u>	Choisit un groupe-cours et sélectionne l'option « Modifier un groupe-cours ».		
		<u>2</u>	Présente l'écran de modification de groupe-cours alimentée par toutes les informations connues à propos de celui-ci.
<u>3</u>	Modifie le lieu, le moment et le membre du personnel éducatif associé* au groupe-cours.		
<u>4</u>	Demande l'enregistrement des modifications apportées au groupe-cours.		
		<u>5</u>	Vérifie qu'il n'y a pas de conflit d'horaire pour le membre du personnel éducatif (et que le temps choisi n'est pas dans le passé).
		<u>6</u>	Enregistre le groupe-cours dans la BD.
<u>Exceptions :</u>	E4. a. L'acteur demande l'annulation des modifications. E5. a. Le système détecte un conflit d'horaire entre le groupe-cours actuel et ceux déjà enregistrés. I. Retour à E3 après un message d'erreur significatif.		

* Dans le cas où l'acteur serait un membre du personnel éducatif, cette option ne peut être changée : il ne peut modifier que des groupe-cours où il sera lui-même l'enseignant, ce qui le limite aux choix de lieu et de moment.

3.1.22 Supprimer un groupe-cours

ID :	UC22		
Nom :	Mettre à jour un groupe-cours		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre de supprimer un groupe-cours (dans le cadre d'un atelier, une « représentation » de celui-ci par un enseignant) auquel les utilisateurs pourront être inscrits.		
Résumé :	Un membre du personnel éducatif souhaite offrir un atelier. Il crée un groupe-cours en spécifiant son lieu et son moment, pour permettre d'y inscrire des utilisateurs. Cependant, il s'avère plus tard nécessaire de modifier les informations y figurant. L'acteur sélectionne l'atelier et choisit le groupe-cours. Il peut ensuite le modifier.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO22		
Pré-condition :	L'acteur est authentifié et autorisé à modifier un groupe-cours. L'atelier associé est déjà en consultation.		
Post-condition :	La liste de groupe-cours de l'atelier est à jour. La liste des inscriptions du groupe-cours est à jour.		
Interactions :			
	Acteurs		Système
1	Choisit un groupe-cours et sélectionne l'option « Supprimer un groupe-cours »*.		
		2	Demande confirmation de la suppression du groupe-cours en avertissant de la suppression d'inscriptions associées, le cas échéant.
3	Confirme la suppression du groupe-cours.		
		4	Enregistre le groupe-cours dans la BD.
Exceptions :	E3. a. L'acteur demande l'annulation de la suppression.		

* Dans le cas où l'acteur serait un membre du personnel éducatif, cette option ne sera active que si le groupe-cours devait être enseigné par lui-même.

3.1.23 Inscrire un utilisateur à un atelier

ID :	UC23		
Nom :	Inscrire un utilisateur à un atelier		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre d'inscrire un utilisateur à un groupe-cours d'un atelier.		
Résumé :	Un membre du personnel administratif sélectionne un utilisateur, puis choisit un groupe-cours d'atelier pour y inscrire cet utilisateur. Le système vérifie si l'inscription est possible (et nécessaire), et dans l'affirmative enregistre l'inscription.		
Type :	Importance (primaire), Modélisation (essentiel)		
Références :	CO23 ; Consulter le dossier d'un utilisateur		
Pré-condition :	L'acteur est authentifié et autorisé à inscrire un utilisateur à un atelier		
Post-condition :	La liste des inscriptions liées aux groupe-cours de l'atelier est à jour.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Demande l'interface d'inscription des utilisateurs à un atelier		
		<u>2</u>	Présente l'interface d'inscription des utilisateurs à un atelier
<u>3</u>	Sélectionne l'atelier pour lequel faire l'inscription		
		<u>4</u>	Affiche les groupe-cours associés à cet atelier.*
<u>5</u>	Sélectionne l'utilisateur que l'on souhaite inscrire.		
<u>6</u>	Demande l'inscription de l'utilisateur à ce groupe-cours.		
		<u>7</u>	Valide que l'utilisateur n'a pas déjà réussi l'atelier.
		<u>8</u>	Valide qu'il reste suffisamment de place à ce groupe-cours.
		<u>9</u>	Enregistre l'inscription dans la BD.
<u>Exceptions :</u>	<p>E4. a. Aucun groupe-cours n'est associé à l'atelier</p> <p>E7. a. L'utilisateur a déjà réussi l'atelier.</p> <p>E8. a. Il ne reste plus de place pour l'utilisateur dans le groupe-cours.</p> <p>Dans tous les cas, arrêt du use-case après l'affichage d'un message d'erreur significatif.</p>		

*Seuls les groupe-cours à venir sont affichés – ceux déjà passés (en se fiant à la date système) étant ignorés.

3.1.24 Supprimer l'inscription d'un utilisateur à un atelier

ID :	UC24		
Nom :	Supprimer l'inscription d'un utilisateur à un atelier		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre de supprimer l'inscription d'un utilisateur à un atelier, opération devant être faite pour une raison quelconque.		
Résumé :	Un membre du personnel de l'organisme doit supprimer l'inscription d'un utilisateur à un atelier. Il sélectionne l'atelier et le groupe-cours, puis choisit un utilisateur qui est inscrit à ce dernier. Le système lui demande de confirmer la suppression, puis met à jour la base de données.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO24		
Pré-condition :	L'acteur est authentifié et autorisé à supprimer une inscription. L'atelier dont on désire supprimer une inscription est en consultation.		
Post-condition :	La liste des inscriptions dans les groupe-cours associés à l'atelier est à jour.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Choisit de consulter les inscriptions à un atelier.		
		<u>2</u>	Présente la liste des groupes-cours et des inscriptions qui leurs sont associées.
<u>3</u>	Choisit un utilisateur dont il désire supprimer l'inscription.		
		<u>4</u>	Demande confirmation de la suppression de l'inscription
<u>5</u>	Confirme la suppression de l'inscription.		
		<u>6</u>	Supprimer l'inscription dans la BD.
<u>Exceptions :</u>	E3. a. Aucun utilisateur n'est inscrit à ce groupe-cours. E4. a. Le système refuse la suppression, car le groupe-cours a déjà eu lieu. E5. a. L'utilisateur annule la suppression.		

3.1.25 Saisir le résultat d'un utilisateur à un atelier

ID :	UC25		
Nom :	Saisir le résultat d'un utilisateur à un atelier.		
Acteur :	Personnel éducatif		
Objectif :	Permettre de saisir le résultat d'un utilisateur à un atelier, ce qui peut mettre à jour son statut.		
Résumé :	Un groupe-cours d'atelier a eu lieu. Un membre du personnel éducatif saisit le résultat d'un utilisateur qui a participé à son atelier en choisissant de saisir les résultats, en sélectionnant l'utilisateur, puis en lui assignant un résultat. Le système enregistre la note, puis met à jour le statut de l'utilisateur en conséquence.		
Type :	Importance (primaire), Modélisation (essentiel)		
Références :	CO25		
Pré-condition :	L'acteur est authentifié et autorisé à saisir le résultat d'un utilisateur à un atelier.		
Post-condition :	La liste des résultats de l'utilisateur est à jour. Le statut de l'utilisateur est à jour.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Demande la saisie de résultats d'utilisateurs à un atelier.		
		<u>2</u>	Présente l'interface de saisie des inscriptions d'un utilisateur.
<u>3</u>	Sélectionne un atelier.		
		<u>4</u>	Présente les groupe-cours et les utilisateurs inscrits à ces derniers.
<u>5</u>	Choisit un utilisateur et une note à lui attribuer.		
		<u>6</u>	Enregistre le résultat dans la BD.
		<u>7</u>	Change le statut des utilisateurs conditionnels n'ayant que cet atelier à compléter à actif.
<u>Exceptions :</u>	E4. a. Aucun groupe-cours n'est associé à l'atelier. E5. a. Aucun utilisateur n'est associé à l'atelier. E6. a. L'atelier n'a pas eu lieu – impossible de saisir le résultat.		

3.1.26 Obtenir la liste des inscriptions à un atelier

ID :	UC26		
Nom :	Obtenir la liste des inscriptions à un atelier		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre d'obtenir la liste des inscriptions à un atelier.		
Résumé :	Un membre du personnel éducatif souhaite obtenir la liste des inscriptions à un des groupe-cours qu'il enseigne. Il choisit l'atelier et le groupe-cours, puis choisit d'imprimer la liste des inscriptions.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO26 ;		
Pré-condition :	L'acteur est authentifié et autorisé à obtenir la liste des inscriptions d'un atelier.*		
Post-condition :			
Interactions :			
	Acteurs		Système
1	Demande de consulter les renseignements d'un atelier.		
		2	Affiche l'interface de consultation d'un atelier.
3	Choisit un groupe-cours et en demande l'impression.		
		4	Imprime le nom des utilisateurs inscrits à ce groupe-cours.
Exceptions :	E4. a. Aucune imprimante n'est visible par le logiciel.		

*Un membre du personnel éducatif n'est autorisé à afficher que les groupe-cours qu'il enseigne, alors qu'un membre du personnel administratif ne souffre pas de cette restriction.

3.1.27 Créer une activité

ID :	UC27		
Nom :	Créer une activité		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Création d'une nouvelle activité dans le système par un usager.		
Résumé :	L'organisme souhaite organiser une nouvelle activité. Un membre du personnel administratif est chargé de saisir ce nouvel atelier dans le système – un membre du personnel éducatif peut aussi le faire, mais seulement s'il l'enseigne.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO27		
Pré-condition :	L'acteur est authentifié et a le droit de créer une nouvelle activité.		
Post-condition :	La liste des activités est à jour.		
Interactions :			
	Acteurs		Système
1	Sollicite la création d'une nouvelle activité.		
		2	Présente l'écran de création d'une nouvelle activité.
3	Saisit les informations appropriées : nom, description de l'activité, lieu et temps.		
4	Demande l'enregistrement de la nouvelle activité.		
		5	Valide les renseignements saisis.
		6	Enregistre la nouvelle activité dans la BD.
Exceptions :	E5. a. Les renseignements saisis ne sont pas valides ou présents.*		

*Il est à noter que, tout comme un groupe-cours d'atelier, une activité ne peut être inscrite dans le passé !

3.1.28 Mettre à jour une activité

ID :	UC28		
Nom :	Mettre à jour une activité		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre à l'administrateur du système de gérer les permissions des usagers.		
Résumé :	L'utilisateur souhaite mettre à jour l'une des données d'un atelier (nom, description, etc.)		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO28		
Pré-condition :	L'administrateur du système est déjà authentifié. L'activité est déjà en consultation.		
Post-condition :	L'activité est à jour.		
Interactions :			
	Acteurs		Système
1	Demande l'édition d'une activité.		
		2	Permet d'éditer les champs appropriés (e.g. nom, description, lieu, temps).
3	Saisit les renseignements appropriés.		
4	Demande d'enregistrer les modifications.		
		5	Valide les renseignements saisis.
		6	Enregistre les modifications dans la BD.
Exceptions :	E5. a. Les renseignements saisis ne sont pas valides ou présents.*		

*Comme nous venons de le voir au use-case précédent, il est à noter que, tout comme un groupe-cours d'atelier, une activité ne peut être inscrite dans le passé !

3.1.29 Supprimer une activité

ID :	UC29		
Nom :	Supprimer une activité		
Acteur :	Personnel administratif		
Objectif :	Permettre de marquer comme supprimé une activité qui ne sera pas enseignée – ou qui l’a déjà été.		
Résumé :	Une activité est annulée, ou doit être supprimée. Un membre du personnel administratif consulte l’activité, et demande la suppression. Le système annule toutes les inscriptions et supprime l’activité.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO29		
Pré-condition :	L’acteur est authentifié et autorisé à supprimer une activité. L’activité est en consultation.		
Post-condition :	La liste des activités est à jour. La liste des inscriptions liées à l’activité n’existe plus.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Demande la suppression de l’activité.		
		<u>2</u>	Présente la liste des conséquences associées à la suppression de l’activité.
<u>3</u>	Confirme la demande de suppression.		
		<u>6</u>	Supprime l’activité et les inscriptions qui lui sont associées.
Exceptions :			

3.1.30 Inscrire un utilisateur à une activité

ID :	UC30		
Nom :	Inscrire un utilisateur à une activité		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre d'inscrire un utilisateur à une activité		
Résumé :	Un membre du personnel administratif sélectionne un utilisateur, puis choisit une activité pour y inscrire cet utilisateur. Le système vérifie si l'inscription est possible et dans l'affirmative enregistre l'inscription.		
Type :	Importance (primaire), Modélisation (essentiel)		
Références :	CO30		
Pré-condition :	L'acteur est authentifié et autorisé à inscrire un utilisateur à une activité		
Post-condition :	La liste des inscriptions liées à l'activité est à jour.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Demande l'interface d'inscription des utilisateurs à une activité		
		<u>2</u>	Présente l'interface d'inscription des utilisateurs à une activité
<u>3</u>	Sélectionne l'activité pour laquelle faire l'inscription		
<u>5</u>	Sélectionne l'utilisateur que l'on souhaite inscrire.		
<u>6</u>	Demande l'inscription de l'utilisateur à cette activité.		
		<u>8</u>	Valide qu'il reste suffisamment de place à cette activité.
		<u>9</u>	Enregistre l'inscription dans la BD.
<u>Exceptions :</u>	E8. a. Il ne reste plus de place pour l'utilisateur dans le groupe-cours. I. Arrêt du use-case après l'affichage d'un message d'erreur significatif.		

*Seules les activités à venir sont affichés – celles déjà passées (en se fiant à la date système) étant ignorées.

3.1.31 Supprimer l'inscription d'un utilisateur à une activité

ID :	UC31		
Nom :	Supprimer l'inscription d'un utilisateur à une activité		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre de supprimer l'inscription d'un utilisateur à une activité, opération devant être faite pour une raison quelconque.		
Résumé :	Un membre du personnel de l'organisme doit supprimer l'inscription d'un utilisateur à une activité. Il sélectionne l'activité, puis choisit un utilisateur qui est inscrit à ce dernier. Le système lui demande de confirmer la suppression, puis met à jour la base de données.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO31		
Pré-condition :	L'acteur est authentifié et autorisé à supprimer une inscription. L'activité dont on désire supprimer une inscription est en consultation.		
Post-condition :	La liste des inscriptions de l'activité est à jour.		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Choisit de consulter les inscriptions à une activité.		
		<u>2</u>	Présente la liste des inscriptions.
<u>3</u>	Choisit un utilisateur dont il désire supprimer l'inscription.		
		<u>4</u>	Demande confirmation de la suppression de l'inscription
<u>5</u>	Confirme la suppression de l'inscription.		
		<u>6</u>	Supprimer l'inscription dans la BD.
<u>Exceptions :</u>	E3. a. Aucun utilisateur n'est inscrit à cette activité. E4. a. Le système refuse la suppression, car l'activité a déjà eu lieu. E5. a. L'utilisateur annule la suppression.		

3.1.32 Obtenir la liste des inscriptions à une activité

ID :	UC32		
Nom :	Obtenir la liste des inscriptions à une activité		
Acteur :	Personnel administratif et personnel éducatif		
Objectif :	Permettre d'obtenir la liste des inscriptions à une activité.		
Résumé :	Un membre du personnel éducatif souhaite obtenir la liste des inscriptions à une des activités qu'il enseigne. Il la sélectionne, puis choisit d'imprimer la liste des inscriptions.		
Type :	Importance (secondaire), Modélisation (essentiel)		
Références :	CO32		
Pré-condition :	L'acteur est authentifié et autorisé à obtenir la liste des inscriptions d'une activité.*		
Post-condition :			
Interactions :			
	Acteurs		Système
1	Demande de consulter les renseignements d'une activité.		
		2	Affiche l'interface de consultation d'une activité.
3	Choisit une activité et en demande l'impression.		
		4	Imprime le nom des utilisateurs inscrits à cette activité.
Exceptions :	E4. a. Aucune imprimante n'est visible par le logiciel.		

*Un membre du personnel éducatif n'est autorisé à afficher que les activités qu'il enseigne, alors qu'un membre du personnel administratif ne souffre pas de cette restriction.

3.1.33 Générer un rapport

ID :	UC33		
Nom :	Générer un rapport		
Acteur :	Gestionnaire		
Objectif :	Permettre d'obtenir divers rapports concernant les données conservées par le système.		
Résumé :	Un gestionnaire désire obtenir des rapports administratifs concernant les données gérées par le logiciel. Il sélectionne le générateur de rapports, choisit un type de rapport, saisit les filtres appropriés (lorsque nécessaire) et demande la génération. Le système imprime les informations demandées.		
Type :	Importance (primaire), Modélisation (essentiel)		
Références :	CO33		
Pré-condition :	L'acteur est authentifié et autorisé à générer des rapports.		
Post-condition :			
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
<u>1</u>	Sélectionne le générateur de rapports		
		<u>2</u>	Affiche la liste de rapports possibles
<u>3</u>	Choisit un rapport.		
		<u>4</u>	Affiche la liste des filtres liés au rapport sélectionné.
<u>5</u>	Remplit les champs appropriés des filtres.		
<u>6</u>	Demande la génération du rapport sélectionné en fonction des filtres choisis.		
		<u>7</u>	Valide les données contenues dans les champs des filtres.
		<u>8</u>	Imprime les informations liées au rapport choisi.
Exceptions :			

3.1.34 Afficher les statistiques du système

ID :	UC34		
Nom :	Afficher les statistiques du système		
Acteur :	Gestionnaire		
Objectif :	Permettre d'obtenir les statistiques liées au fonctionnement quotidien de l'organisme.		
Résumé :	Un gestionnaire désire obtenir les statistiques quotidiennes concernant les données gérées par le logiciel. Il sélectionne le générateur de statistiques et demande la génération. Le système imprime les informations appropriées.		
Type :	Importance (primaire), Modélisation (essentiel)		
Références :	CO34		
Pré-condition :	L'acteur est authentifié et autorisé à obtenir les statistiques du système.		
Post-condition :			
Interactions :			
	Acteurs		Système
<u>1</u>	Sélectionne le générateur de statistiques		
<u>2</u>	Demande l'impression des statistiques.		
		<u>3</u>	Imprime les informations liées aux statistiques du système.
Exceptions :			

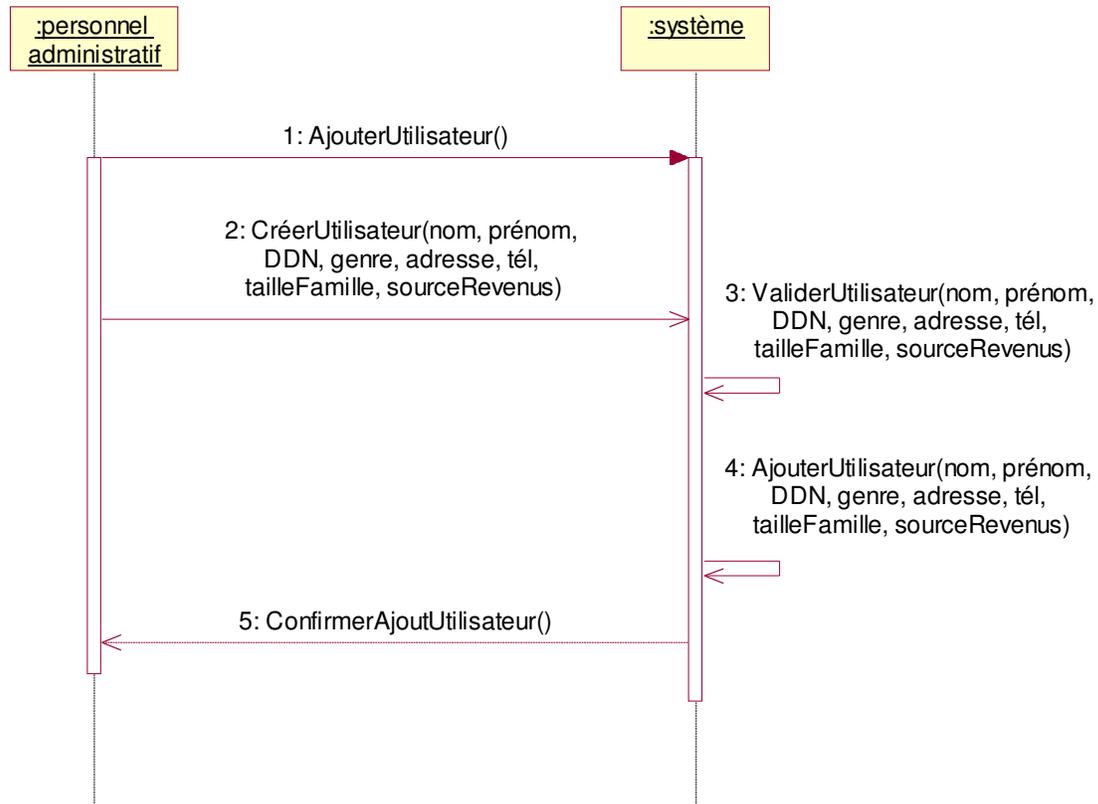
3.1.35 Authentifier un usager

ID :	UC35		
Nom :	Authentifier un usager		
Acteur :	Tous		
Objectif :	Permettre d'authentifier un usager pour qu'il puisse utiliser le système.		
Résumé :	Un usager souhaite effectuer une séance de travail sur le système. Il saisit son nom d'utilisateur et son mot de passe, que le système vérifie. Si les informations sont valides, le logiciel charge l'interface appropriée au niveau de permission de l'utilisateur.		
Type :	Importance (primaire), Modélisation (essentiel)		
Références :	CO35		
Pré-condition :			
Post-condition :	L'utilisateur est authentifié (si ses informations étaient valides)		
Interactions :			
	<u>Acteurs</u>		<u>Système</u>
		<u>1</u>	Affiche l'écran de bienvenue dans le logiciel.
<u>2</u>	Saisit son nom d'utilisateur et son mot de passe.		
		<u>3</u>	Valide les informations de connexion.
		<u>4</u>	Permet l'accès aux fonctions permises à l'utilisateur en fonction de son niveau de permission.
<u>Exceptions :</u>	E3. 1. Les informations ne sont pas valides. I. Le système affiche un message d'erreur approprié et retourne à E1.		

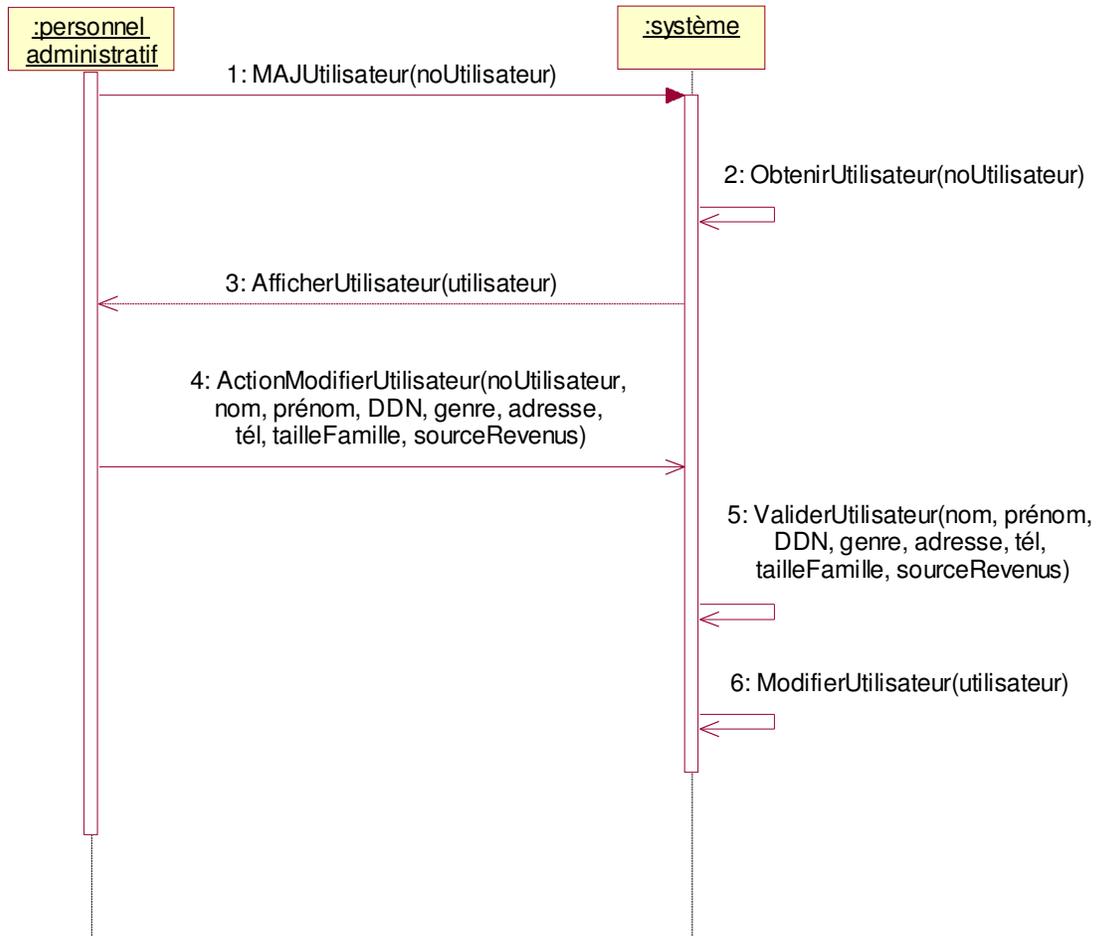
3.2 Spécifications fonctionnelles

3.2.1 Diagrammes de séquence

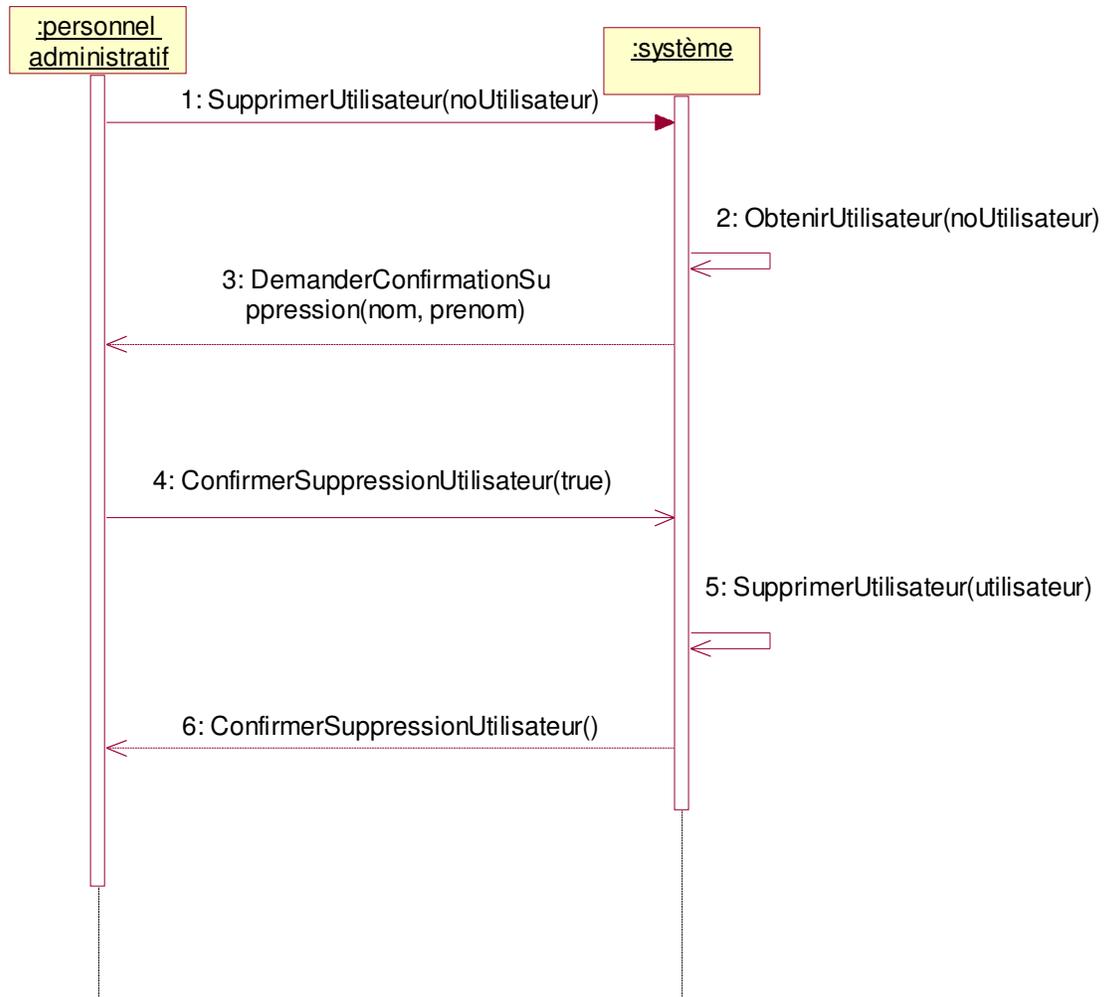
3.2.1.1 Ajouter un dossier d'utilisateur



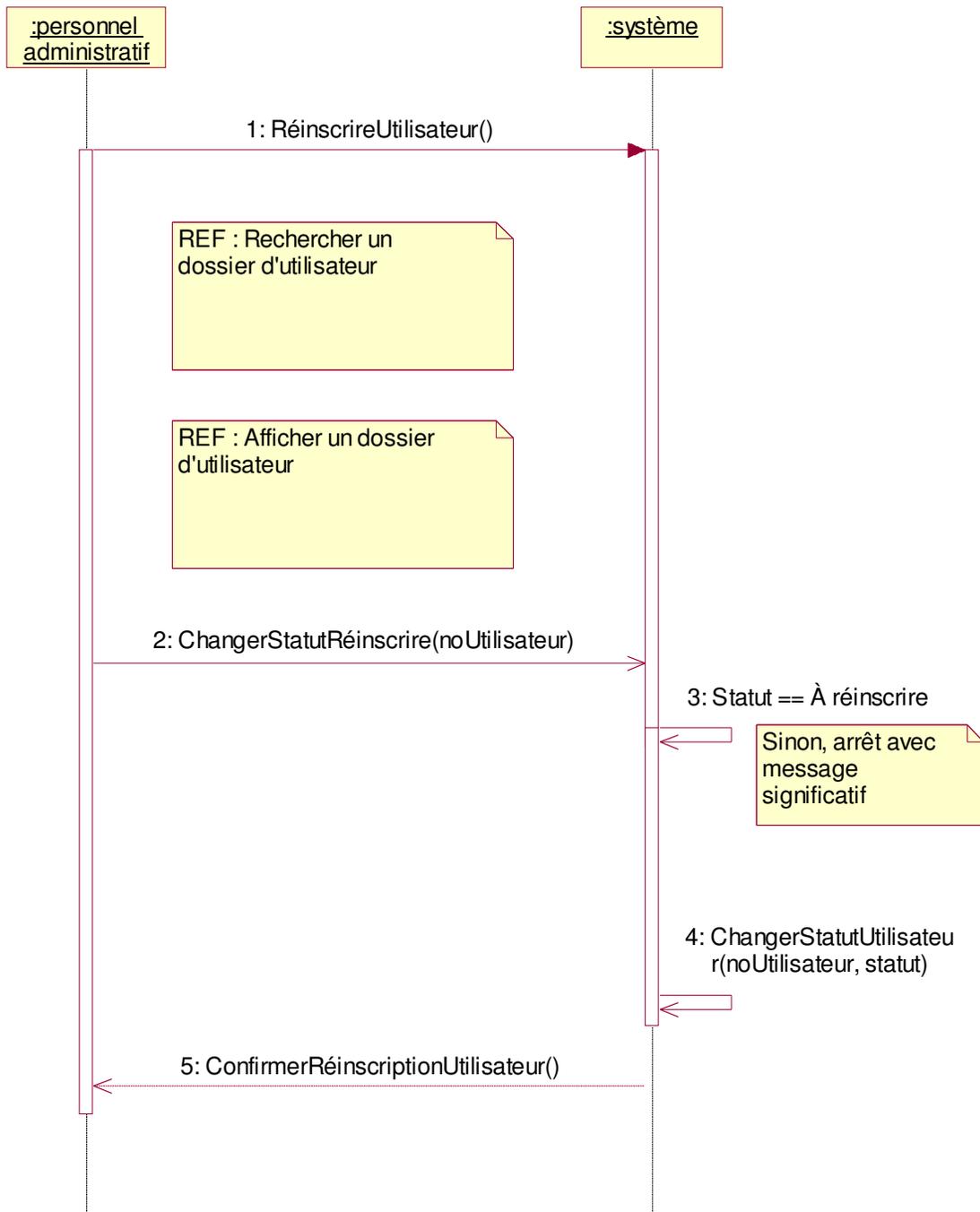
3.2.1.2 Mettre à jour un dossier d'utilisateur



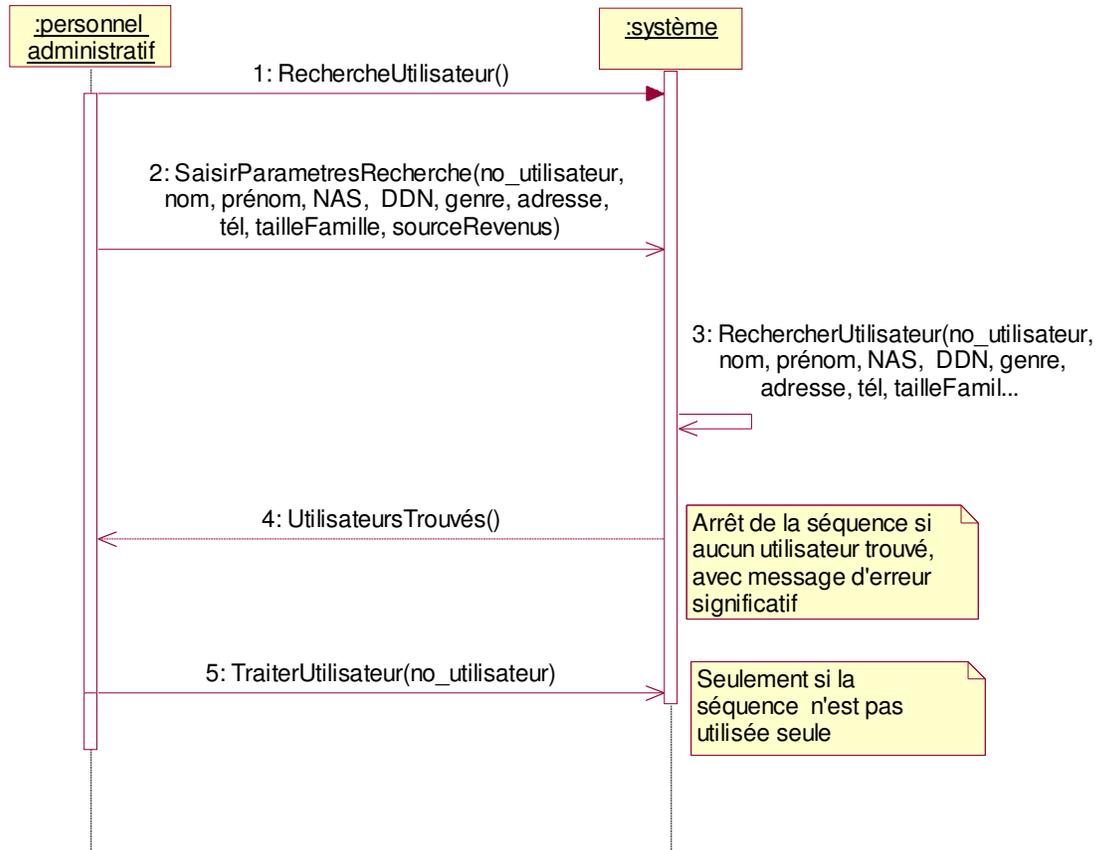
3.2.1.3 Supprimer un dossier d'utilisateur



3.2.1.4 Réinscrire un utilisateur

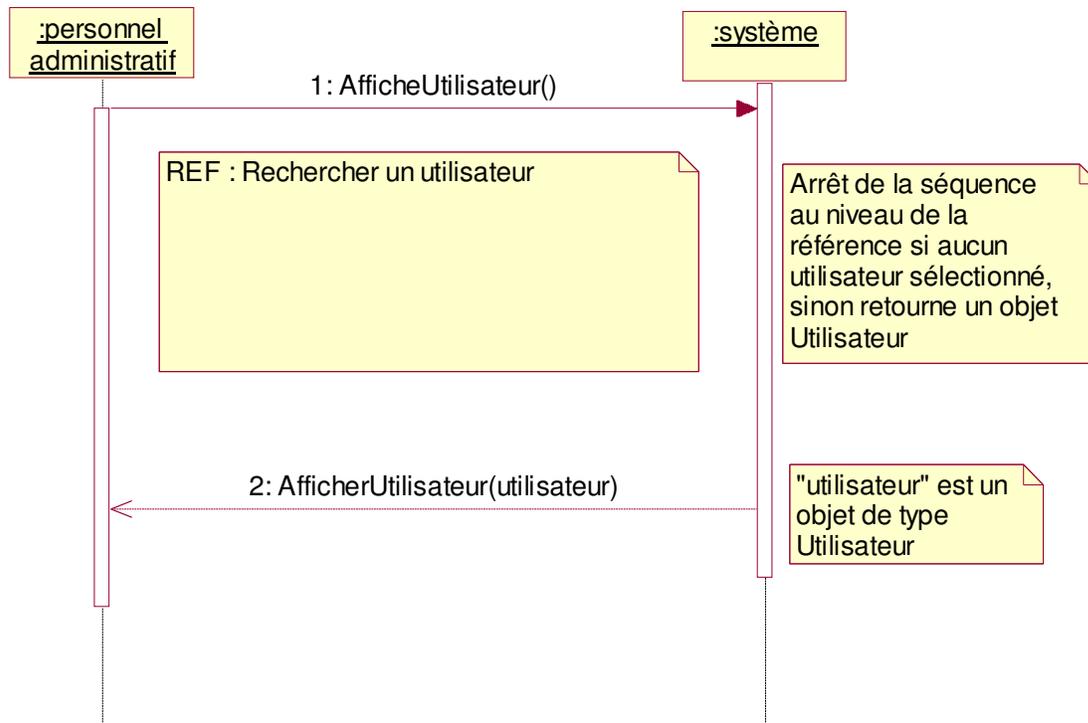


3.2.1.5 Rechercher un dossier d'utilisateur

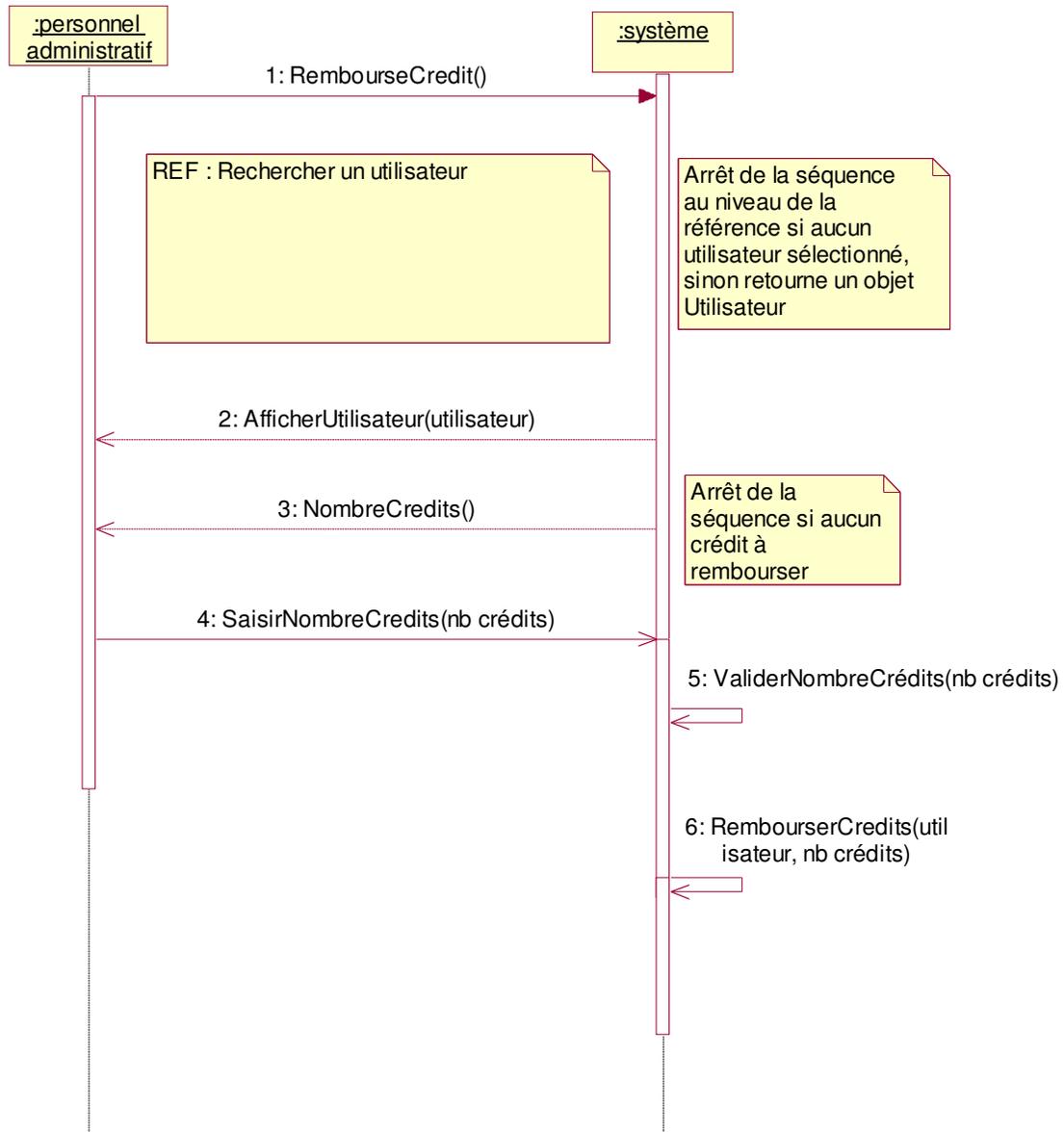


Si la fonction `TraiterUtilisateur` est invoquée, le système continue l'exécution au cas d'utilisation appelant.

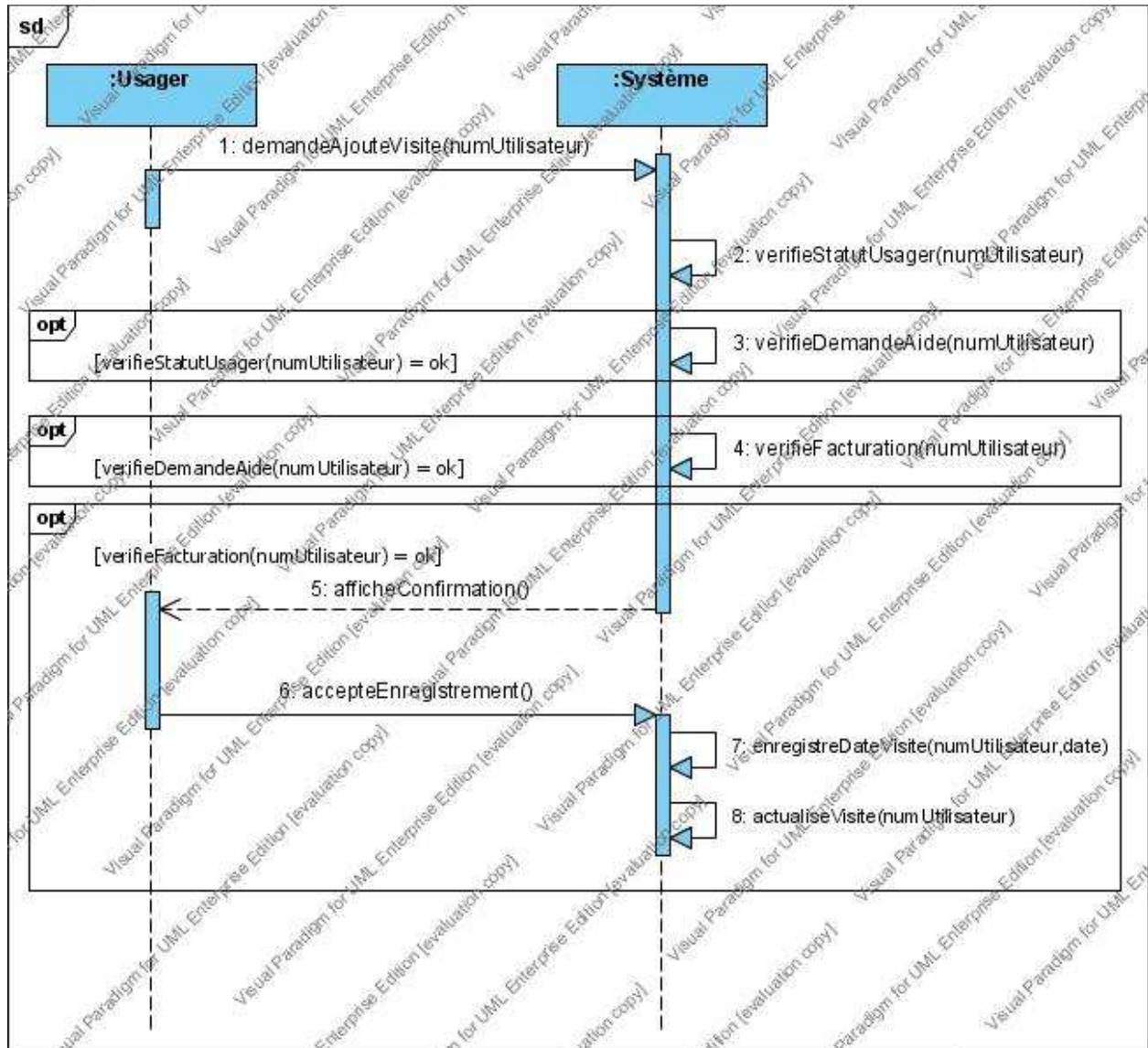
3.2.1.6 Afficher un dossier d'utilisateur

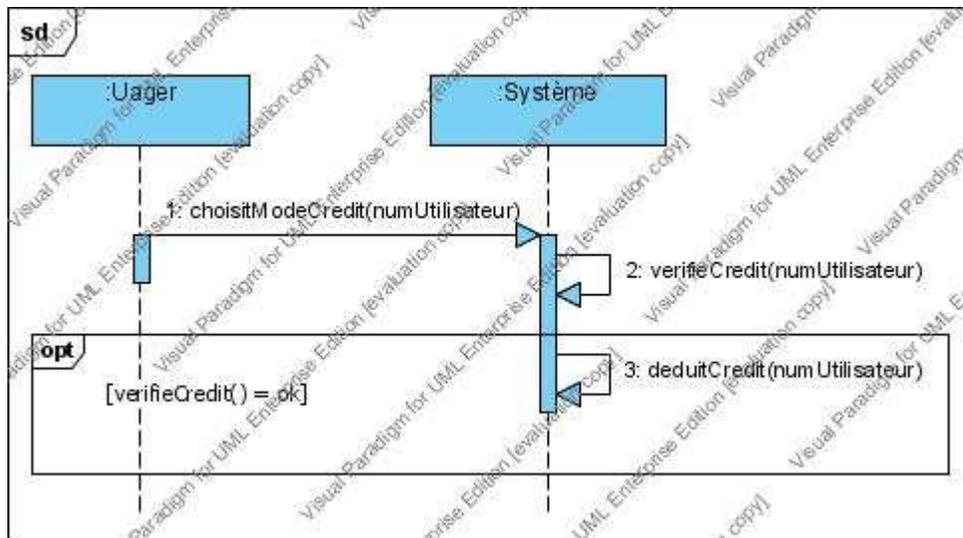


3.2.1.7 Rembourser un crédit

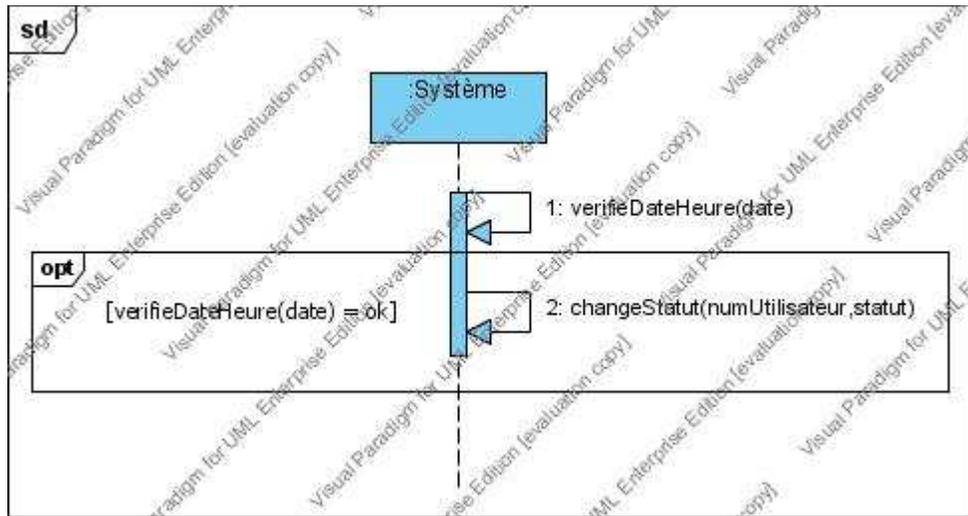


3.2.1.8 Enregistrer la visite d'un utilisateur

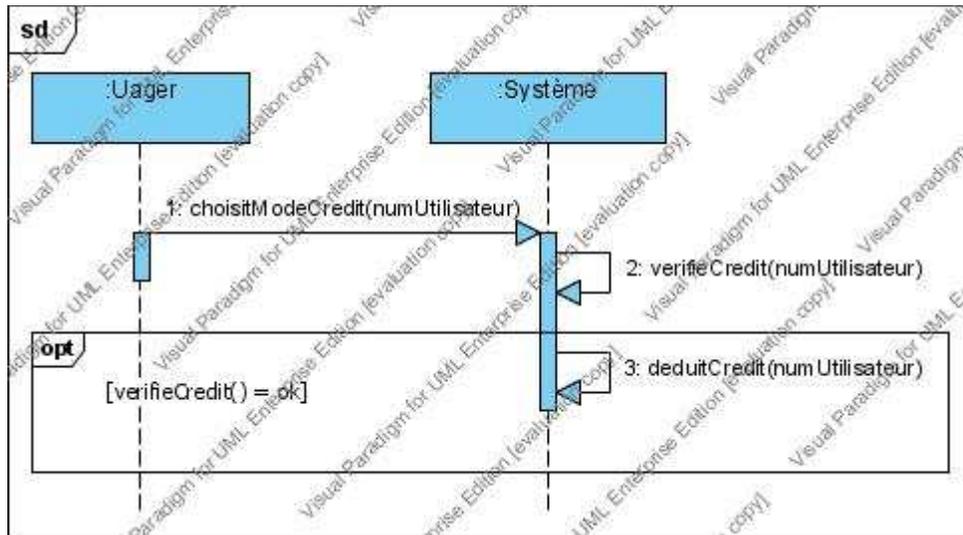




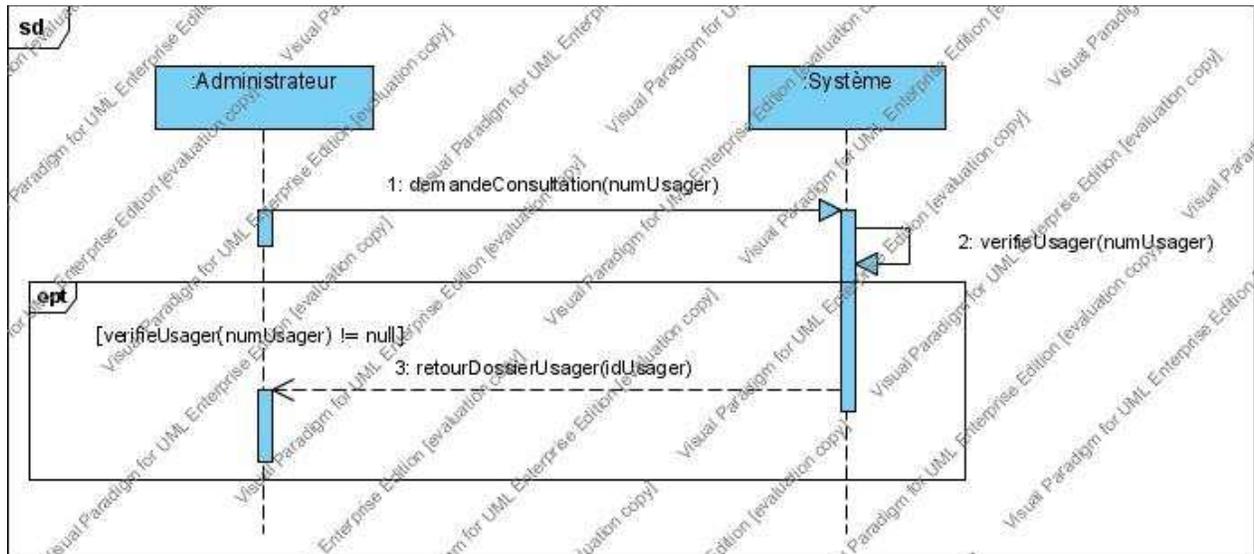
3.2.1.9 Changer le statut des utilisateurs pour « à réinscrire »



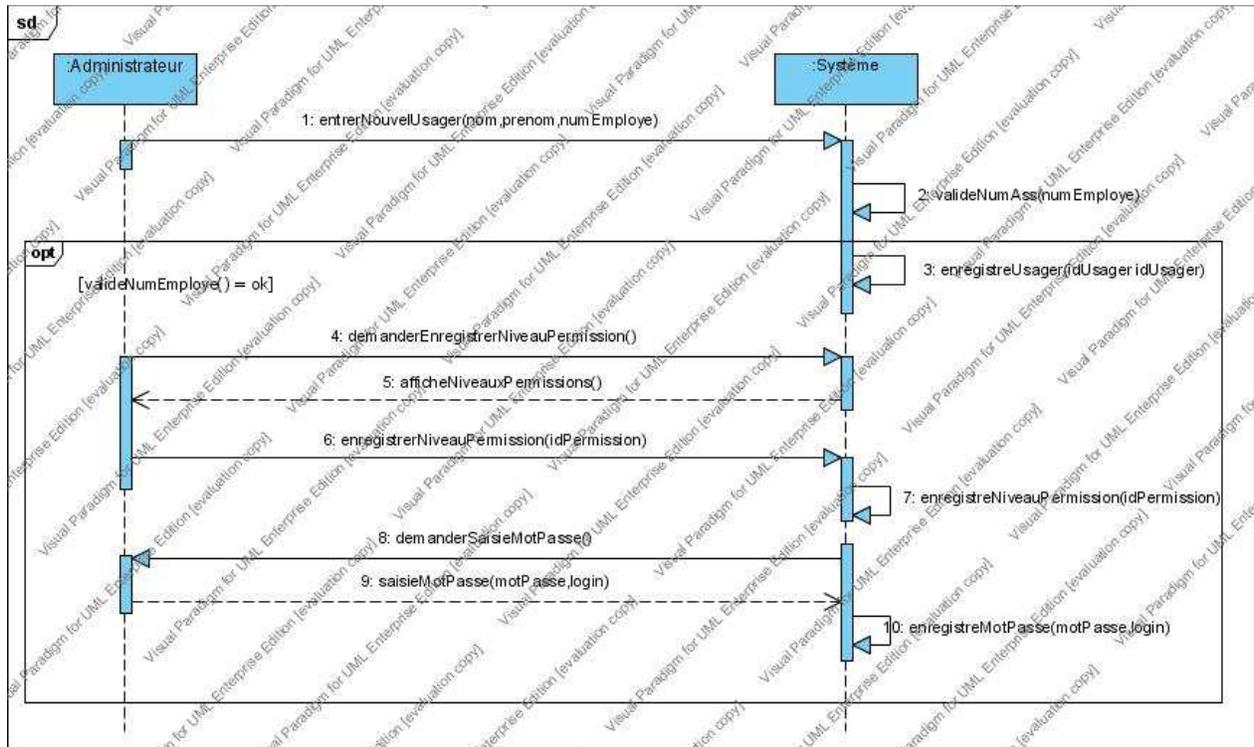
3.2.1.10 Enregistrer un crédit à un utilisateur



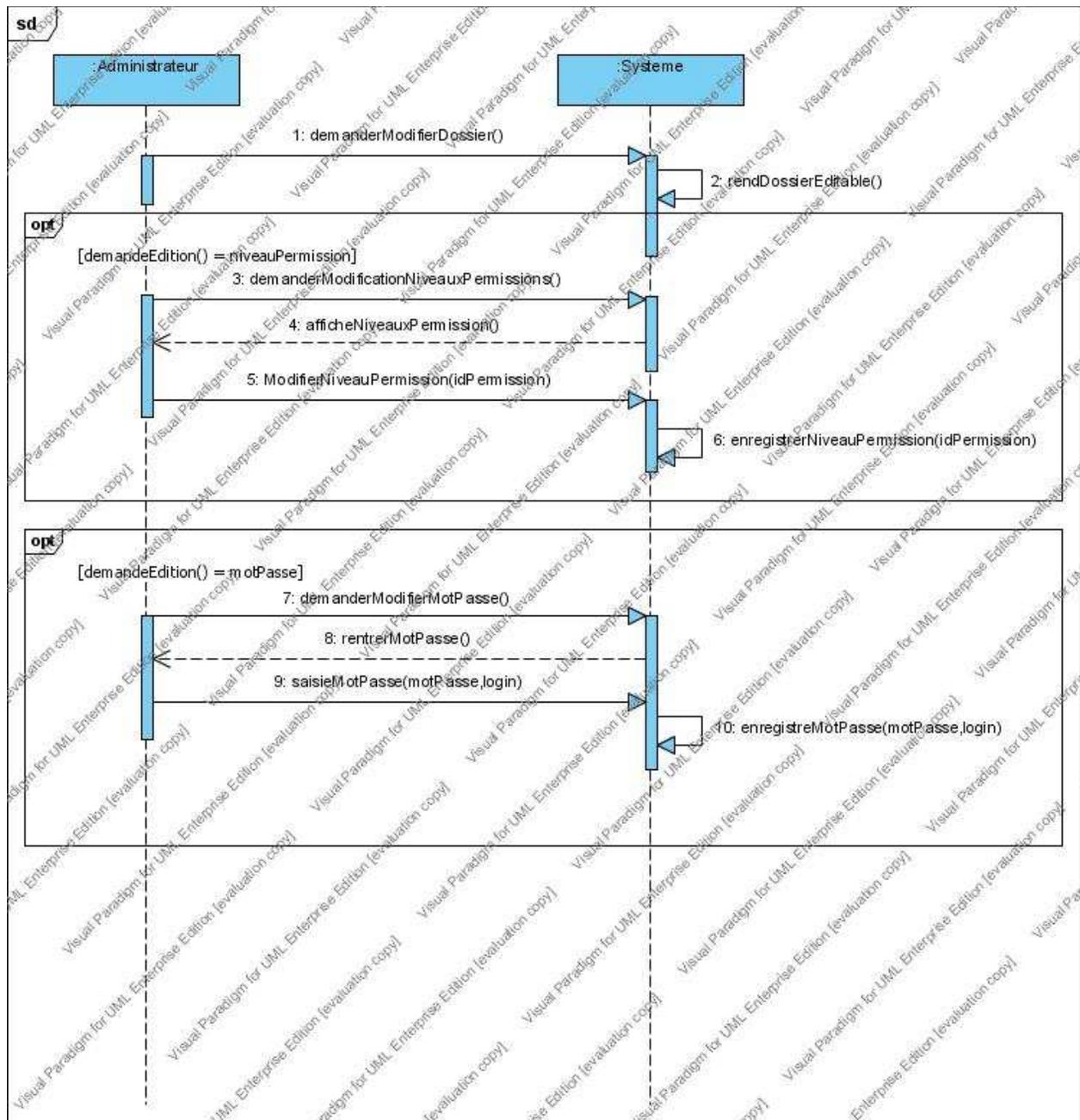
3.2.1.1 Consulter un compte d'utilisateur



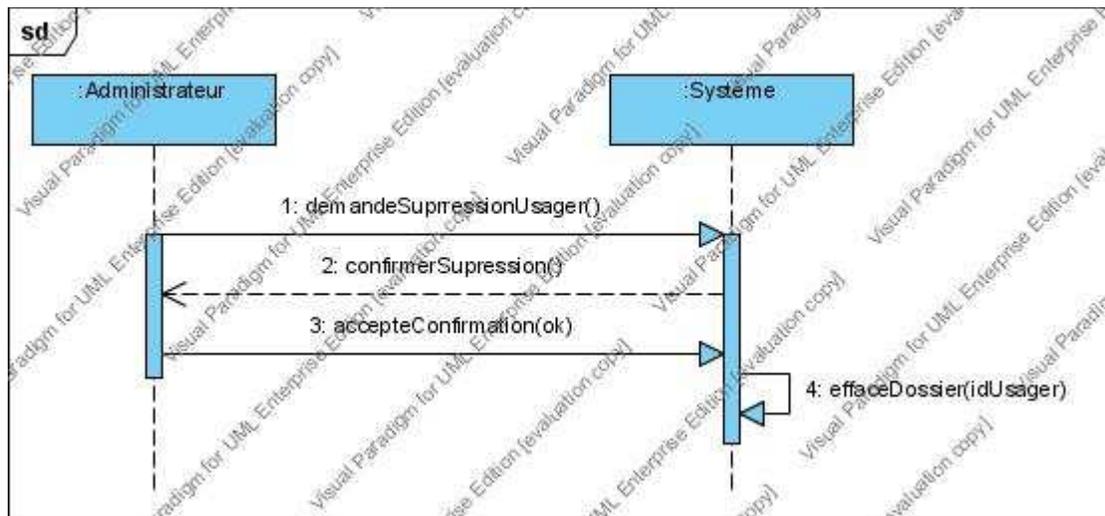
3.2.1.12 Créer un compte d'utilisateur



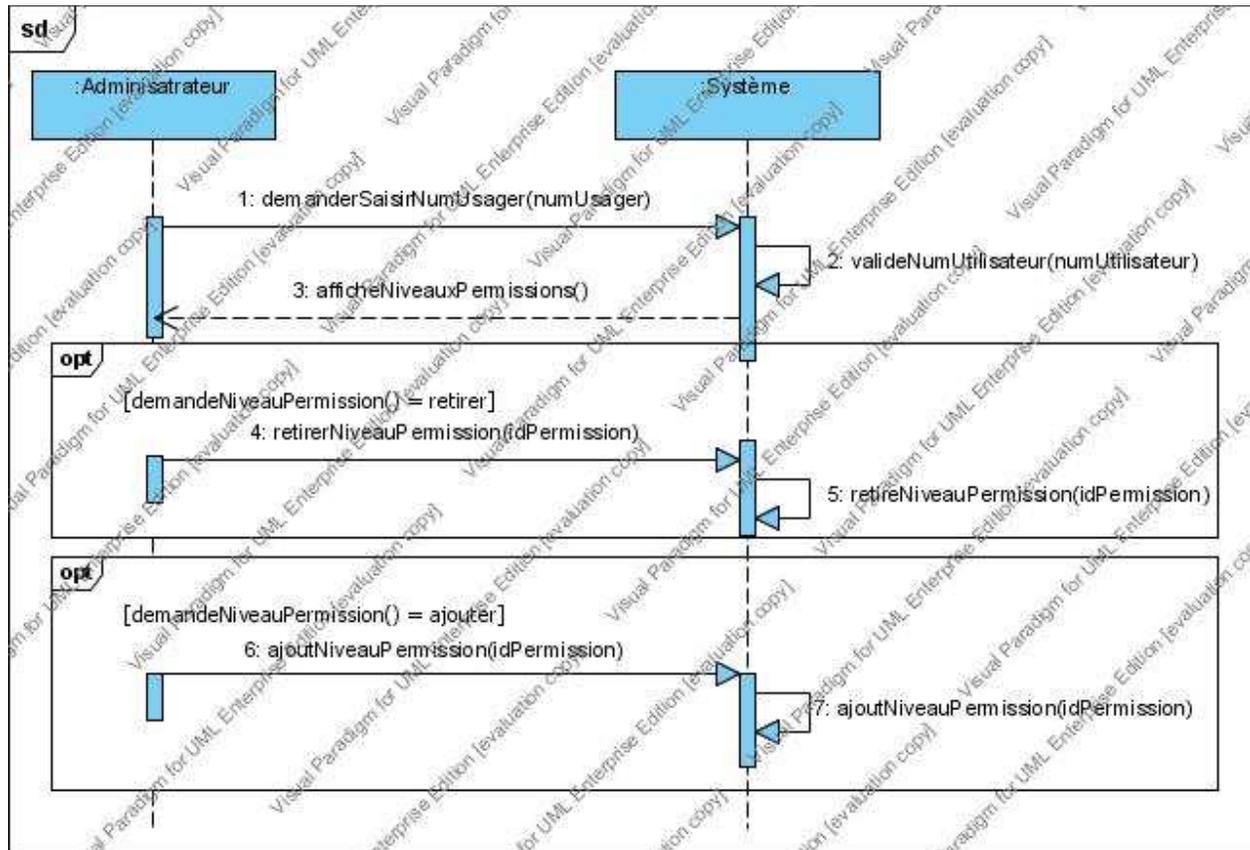
3.2.1.13 Mettre à jour un compte d'utilisateur



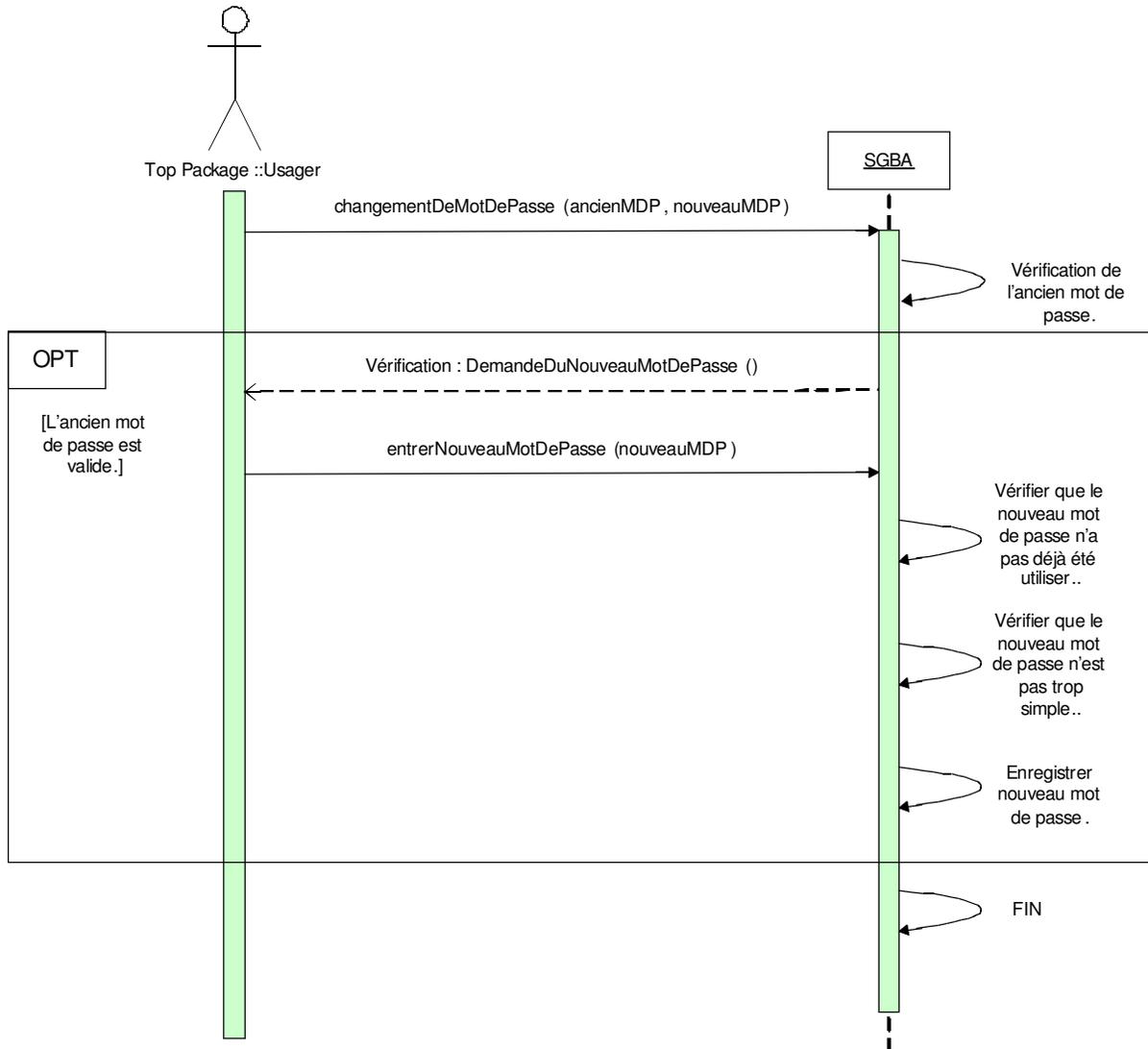
3.2.1.14 Supprimer un compte d'utilisateur



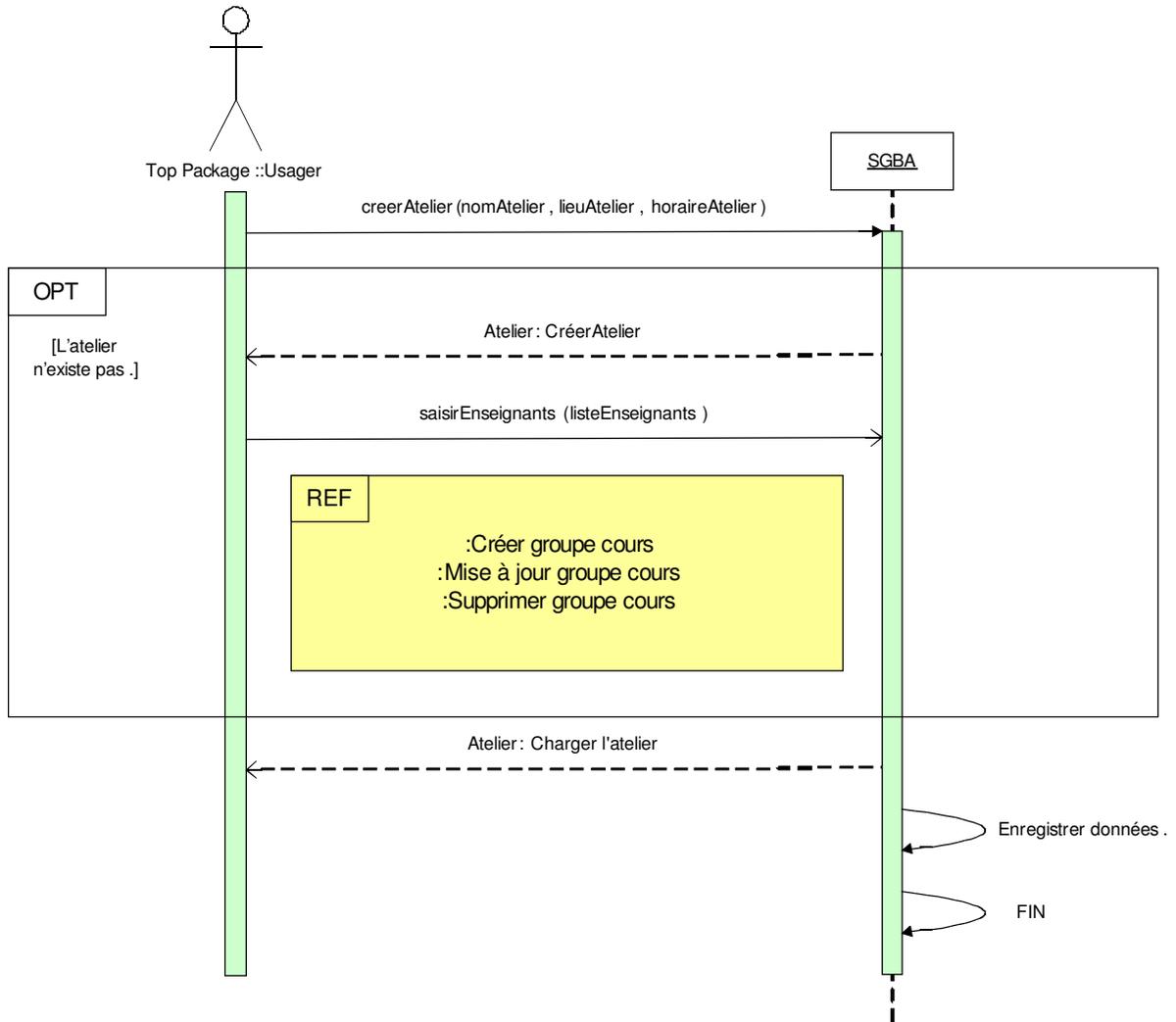
3.2.1.15 Gérer le niveau de permission d'un usager



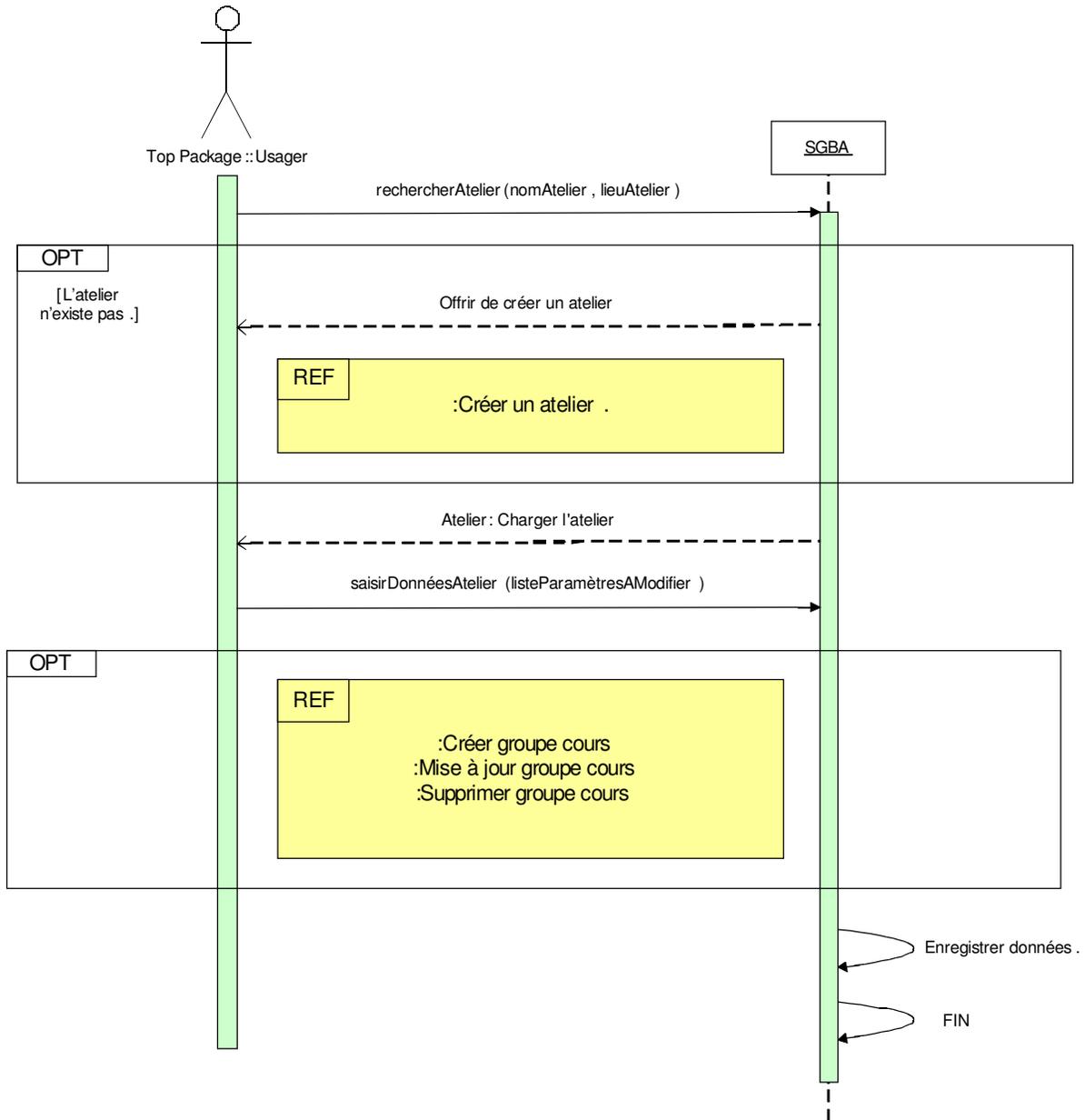
3.2.1.16 Obliger l'utilisateur à changer son mot de passe



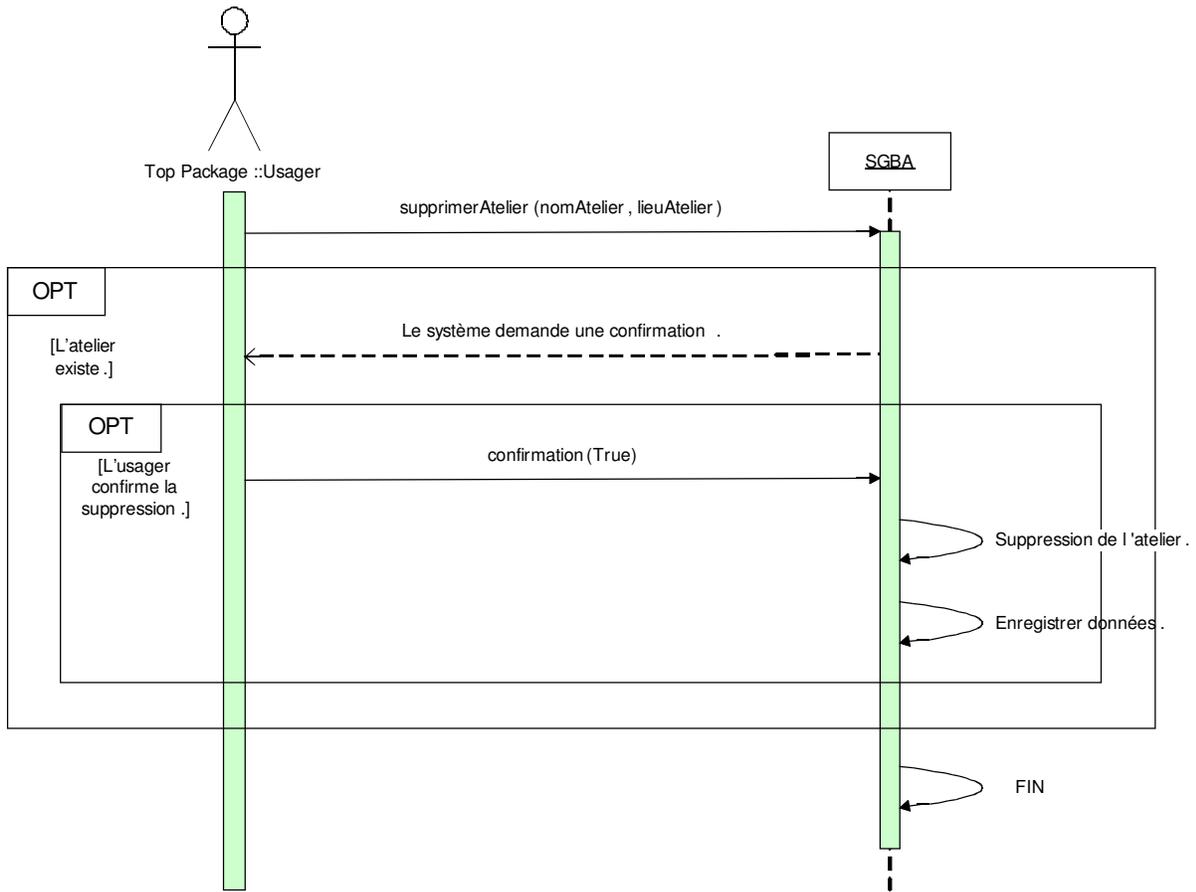
3.2.1.17 Créer un nouvel atelier



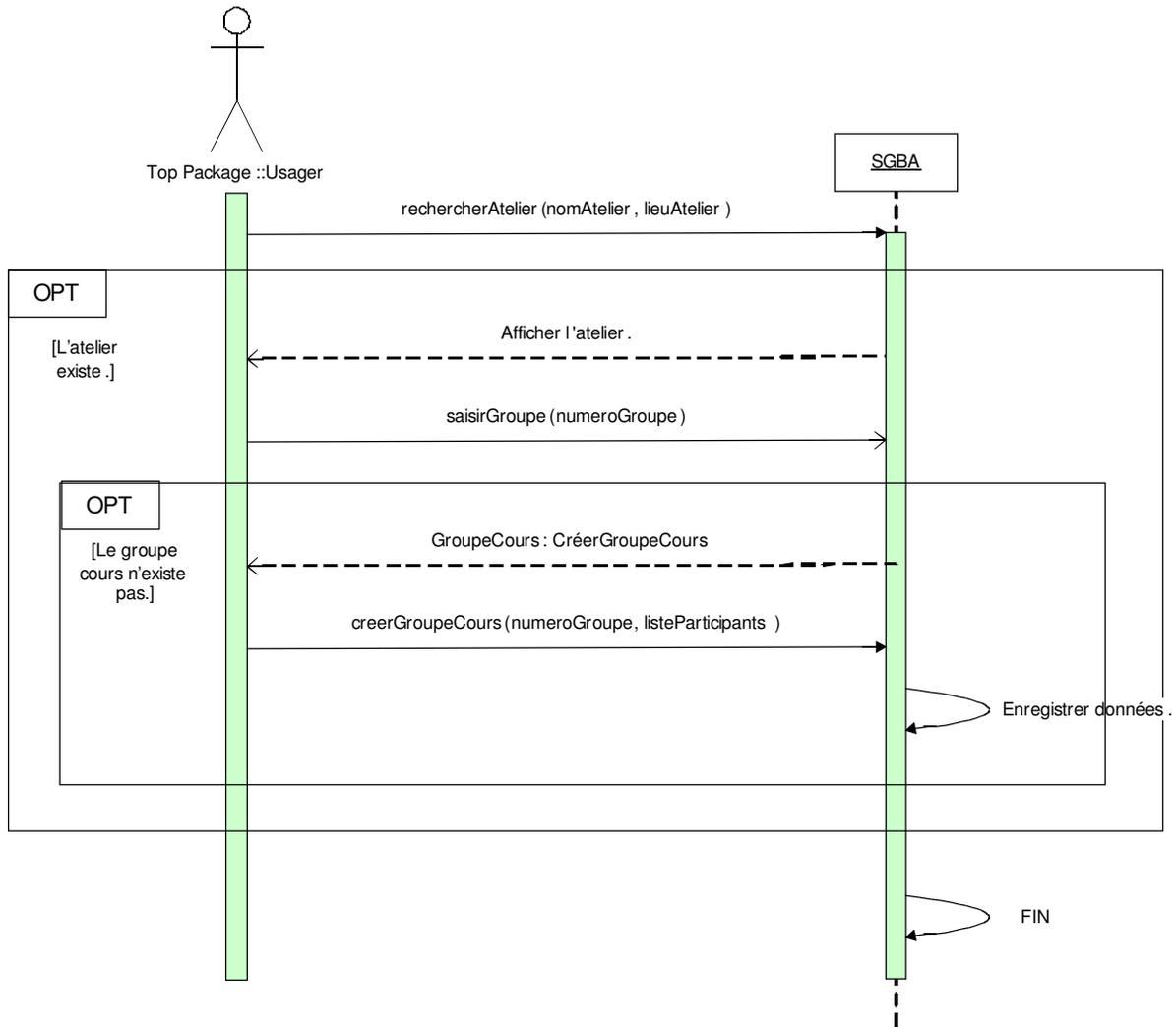
3.2.1.18 Mettre à jour un atelier



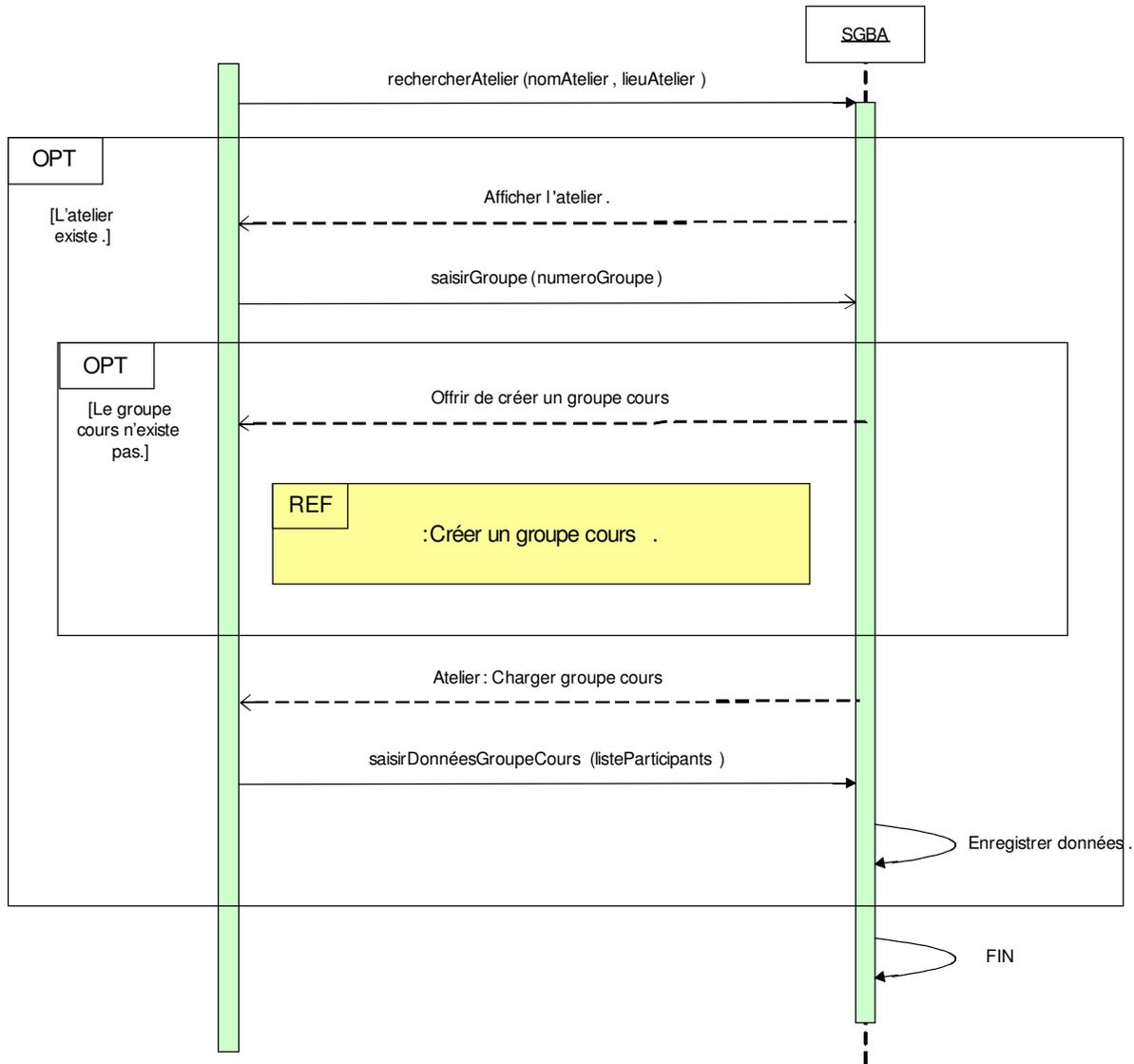
3.2.1.19 Supprimer un atelier



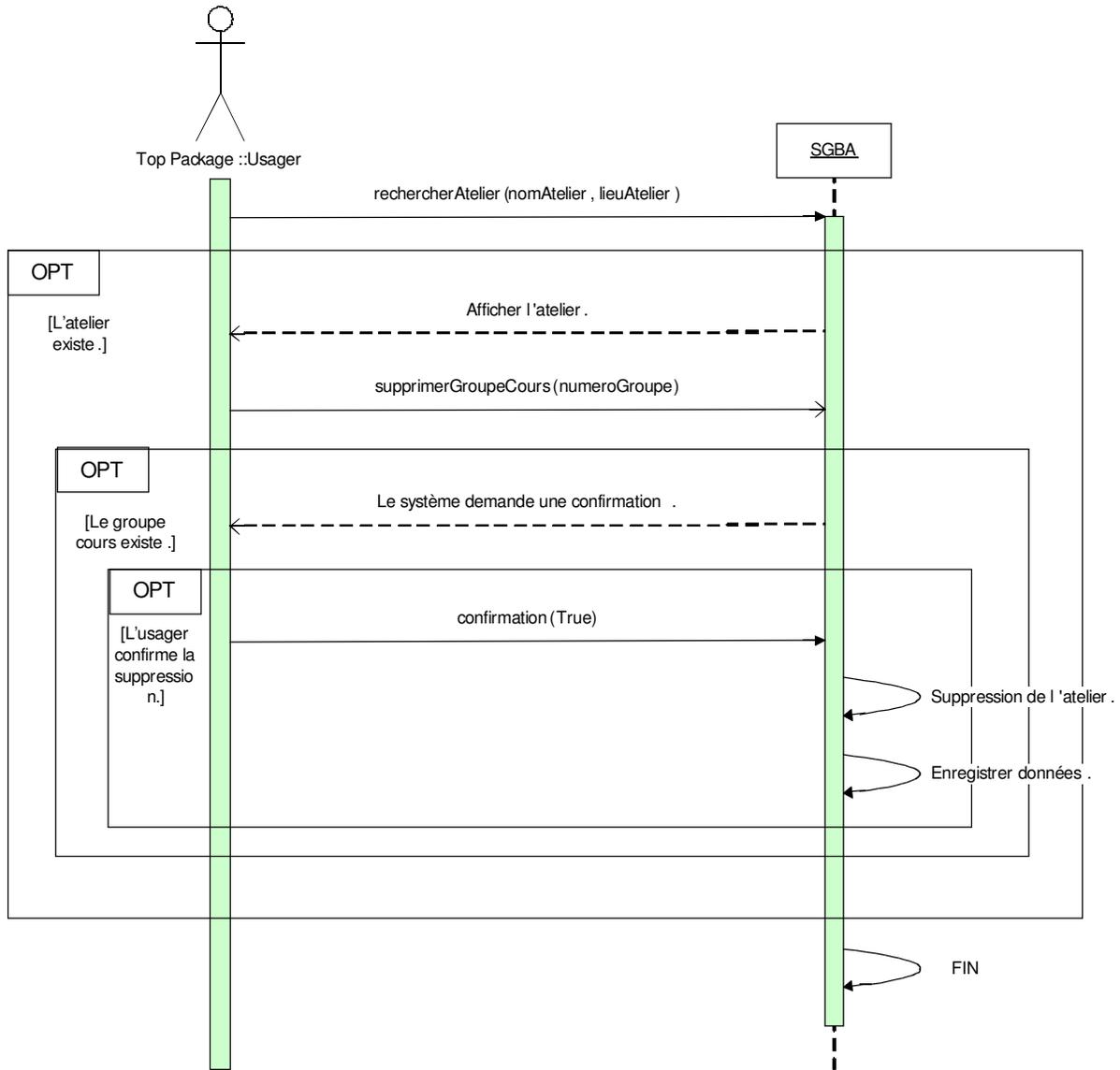
3.2.1.20 Créer un groupe-cours



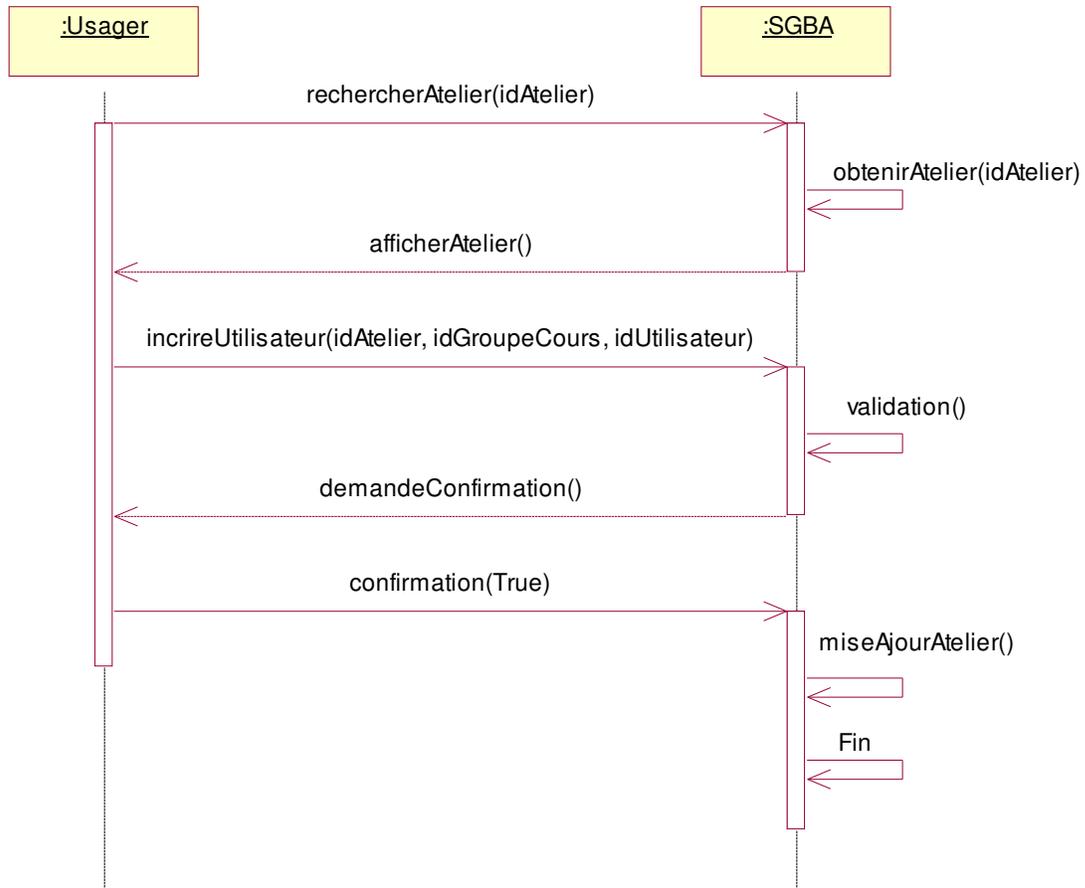
3.2.1.21 Mettre à jour un groupe-cours



3.2.1.22 Supprimer un groupe-cours



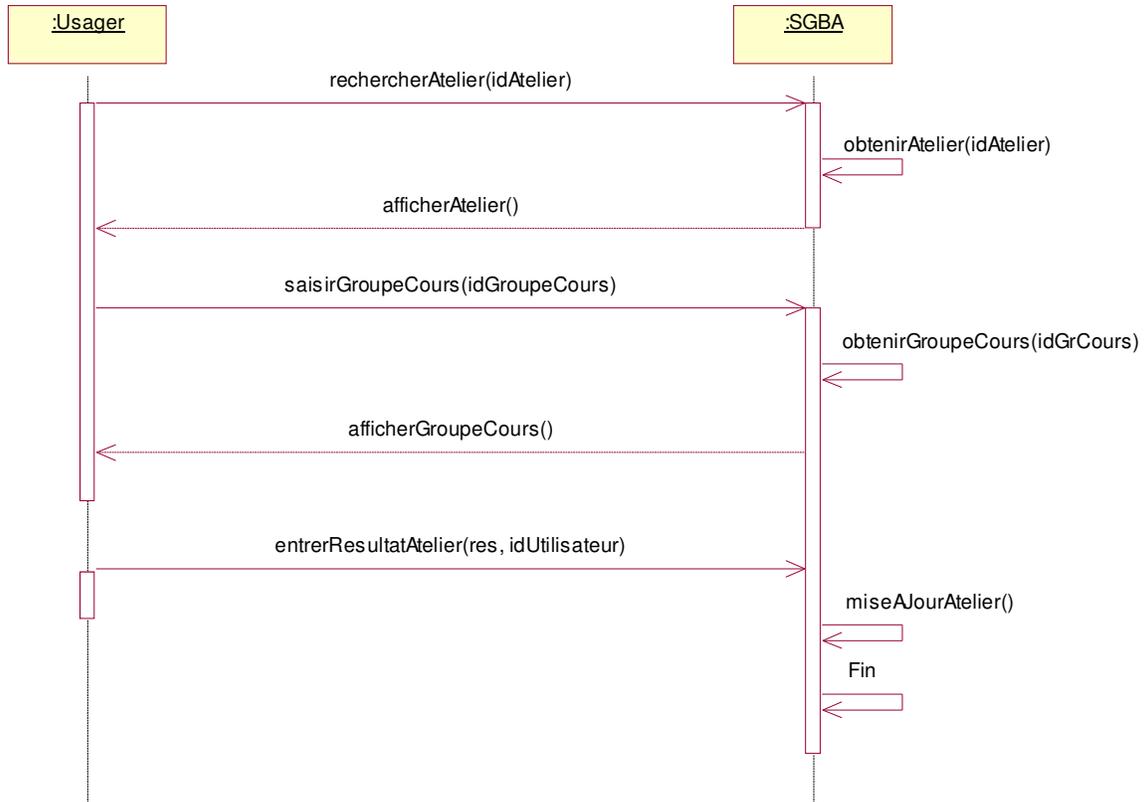
3.2.1.23 Inscrire un utilisateur à un atelier



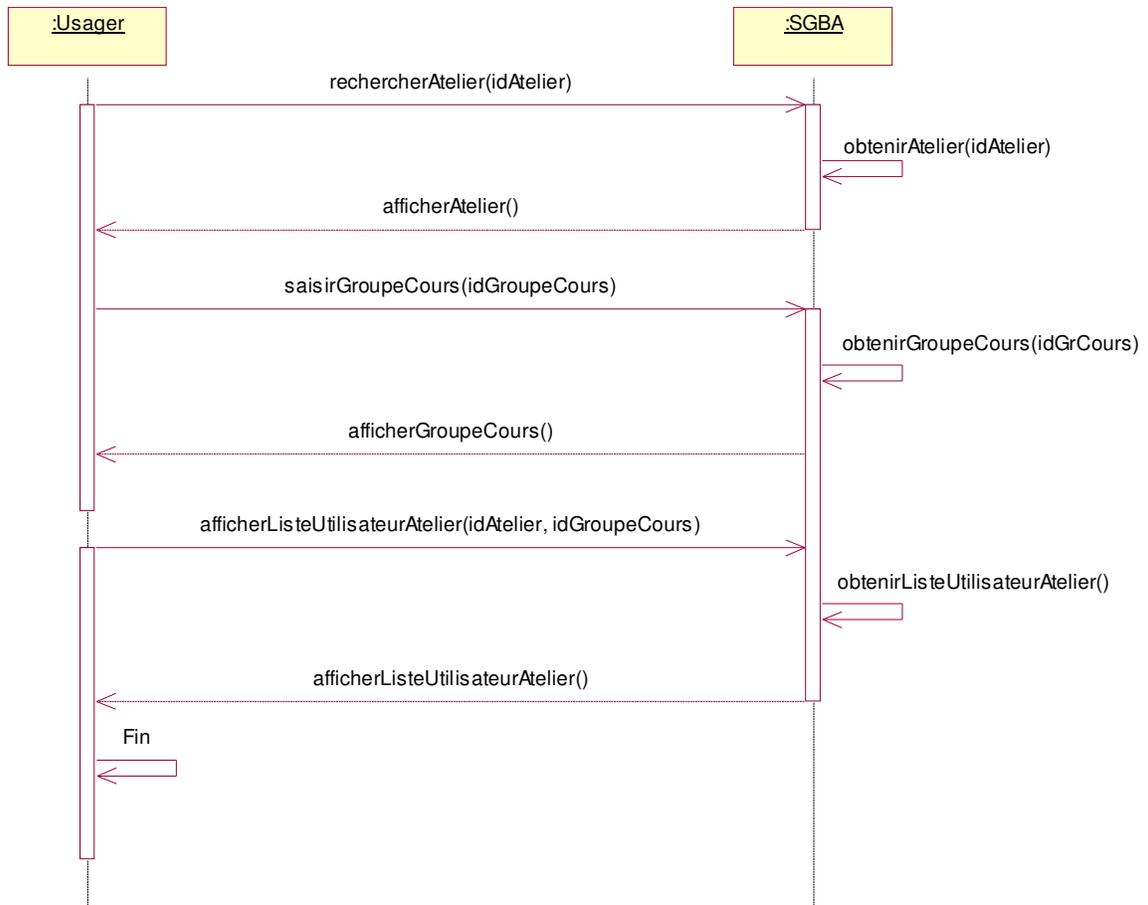
3.2.1.24 Supprimer l'inscription d'un utilisateur à un atelier



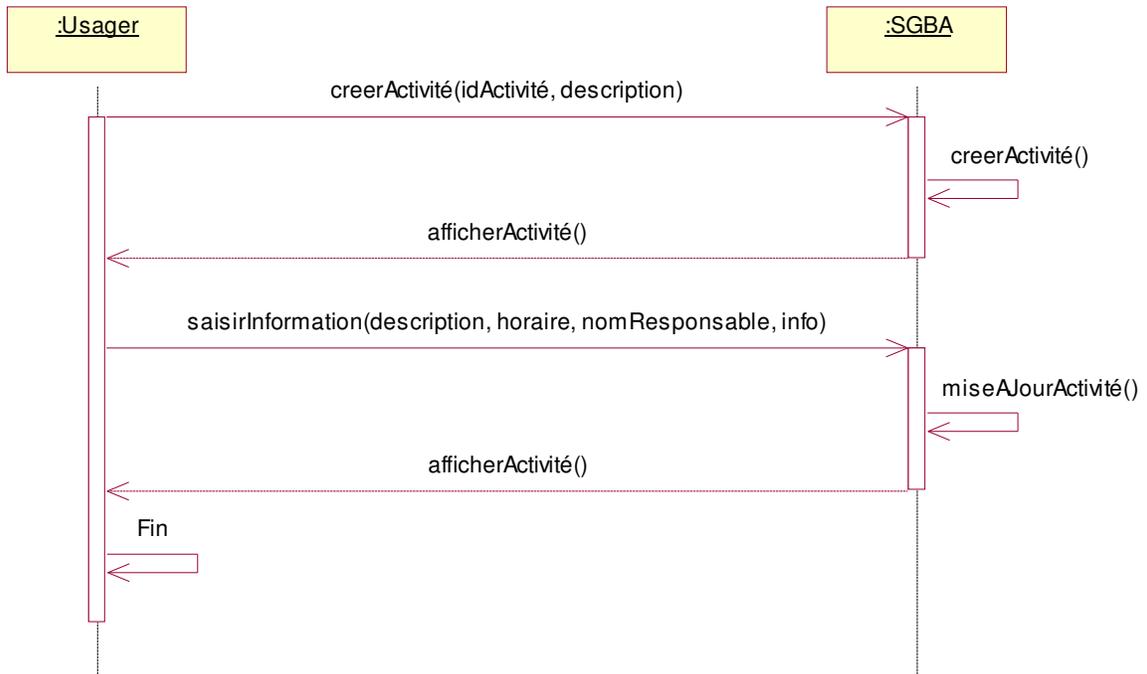
3.2.1.25 Saisir le résultat d'un utilisateur à un atelier



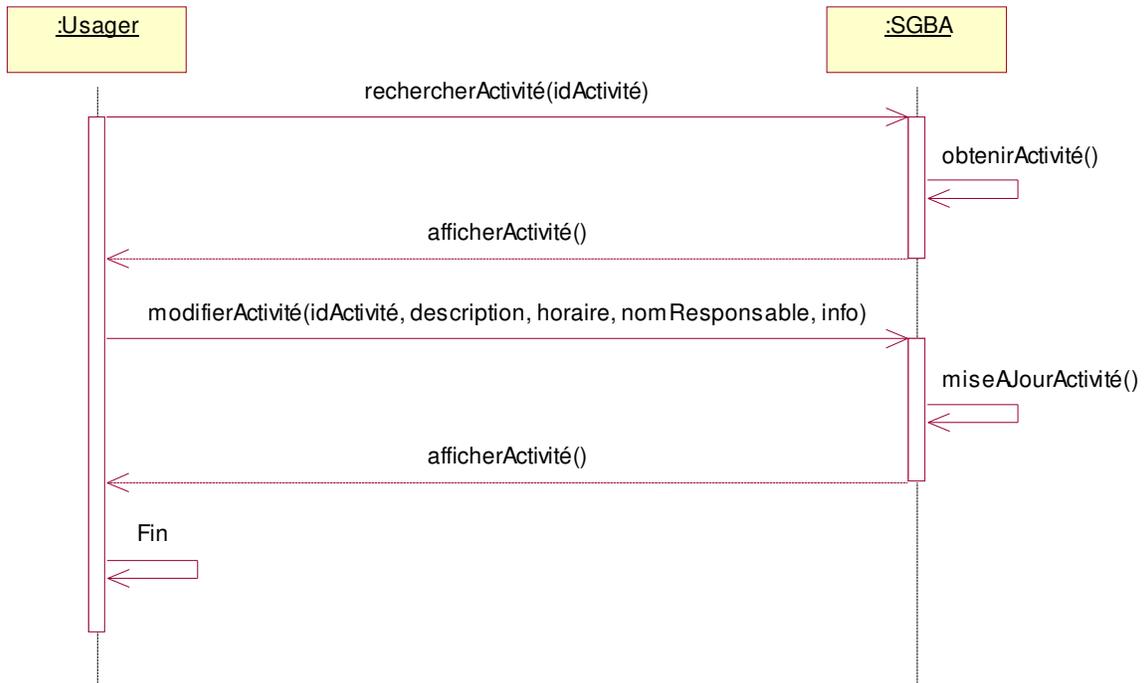
3.2.1.26 Obtenir la liste des inscriptions à un atelier



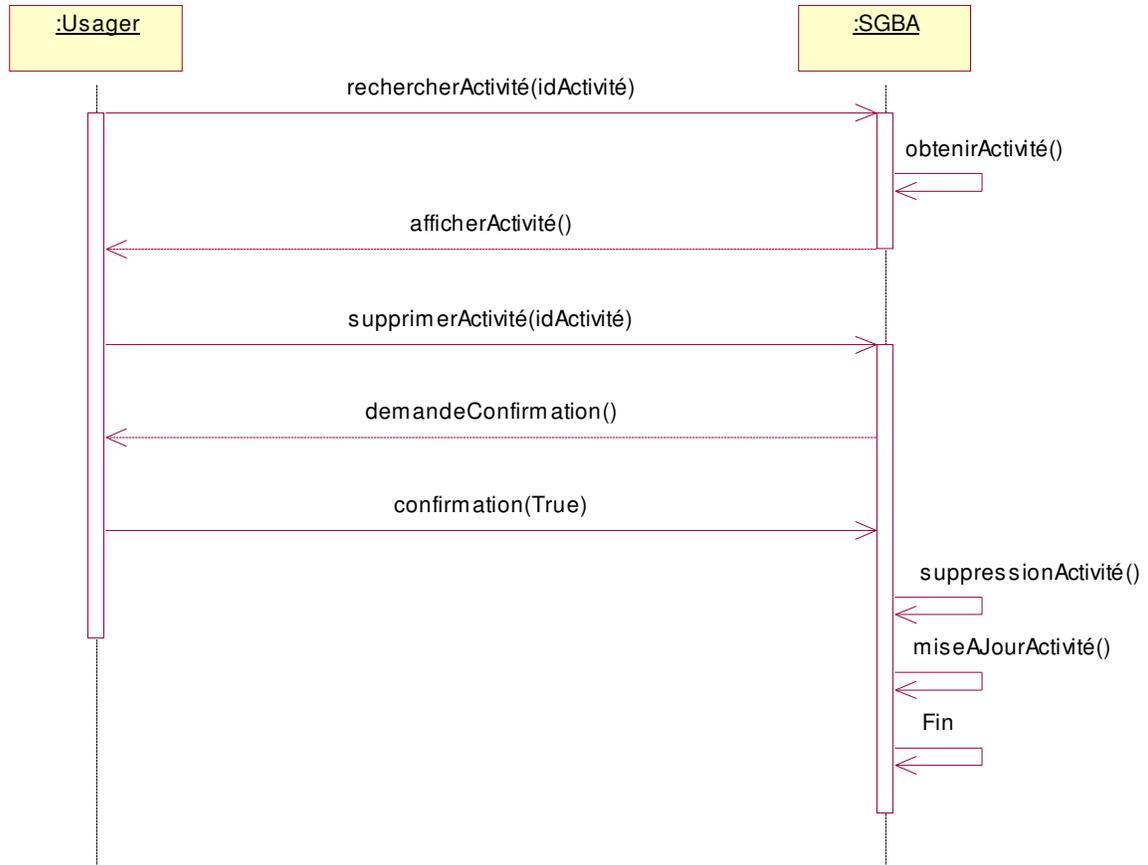
3.2.1.27 Créer une activité



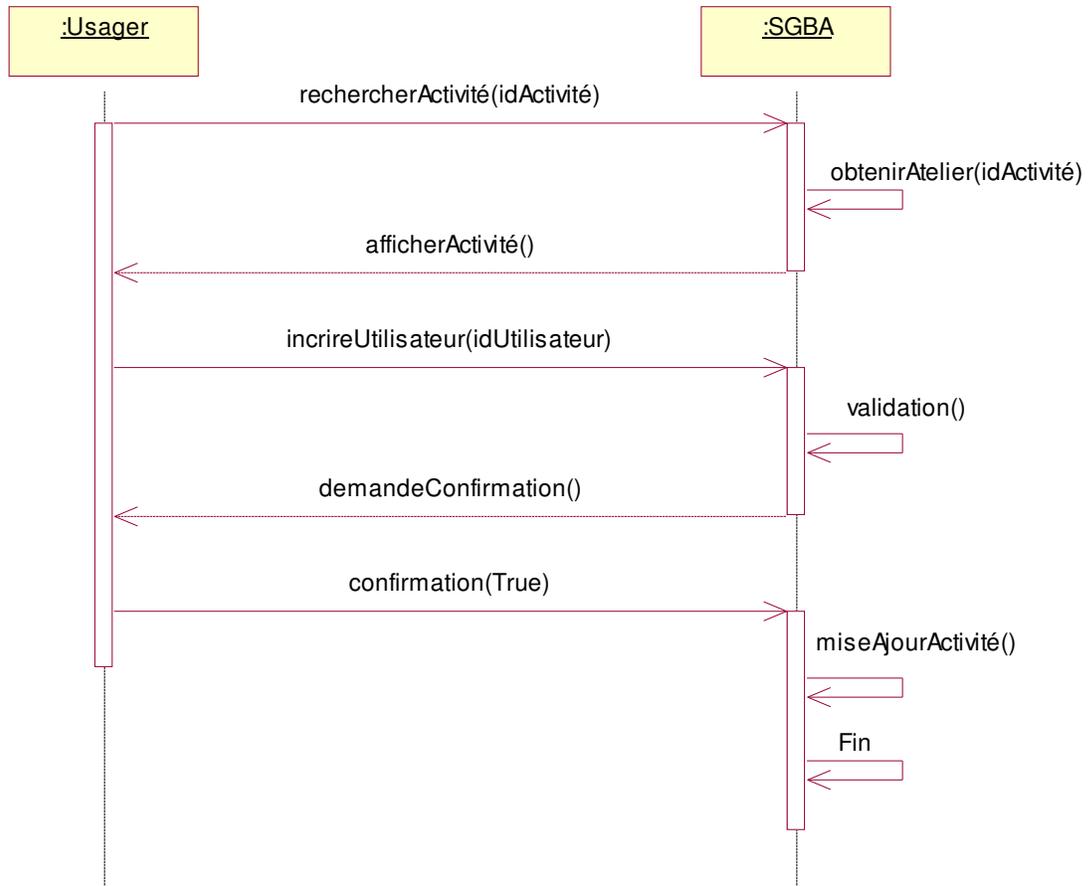
3.2.1.28 Mettre à jour une activité



3.2.1.29 Supprimer une activité



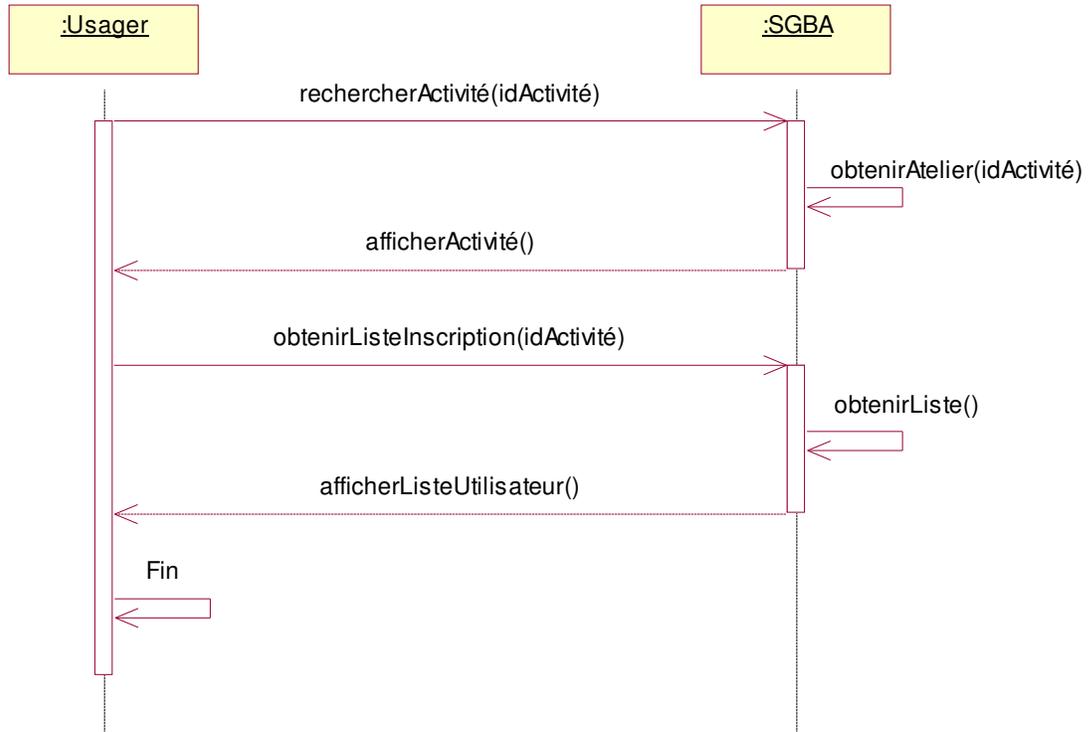
3.2.1.30 Inscrire un utilisateur à une activité



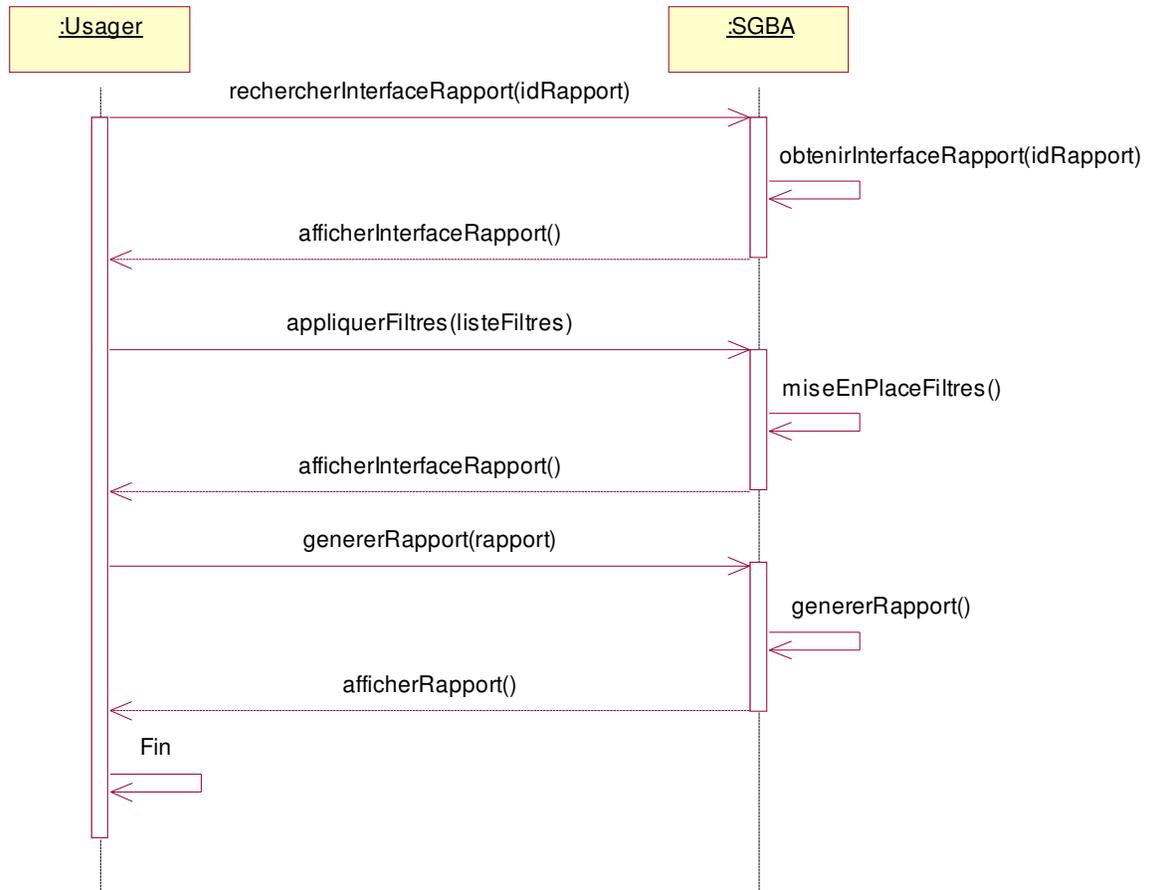
3.2.1.31 Supprimer l'inscription d'un utilisateur à une activité



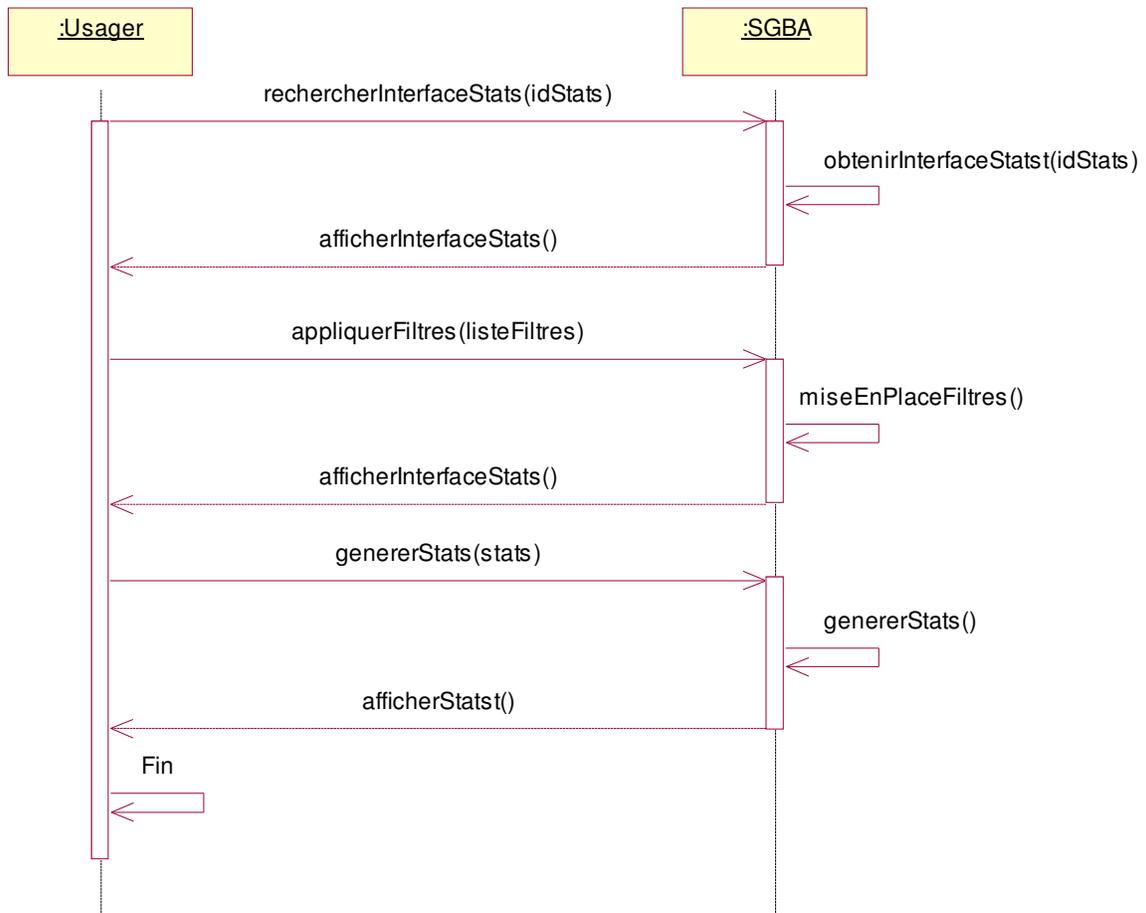
3.2.1.32 Obtenir la liste des inscriptions à une activité



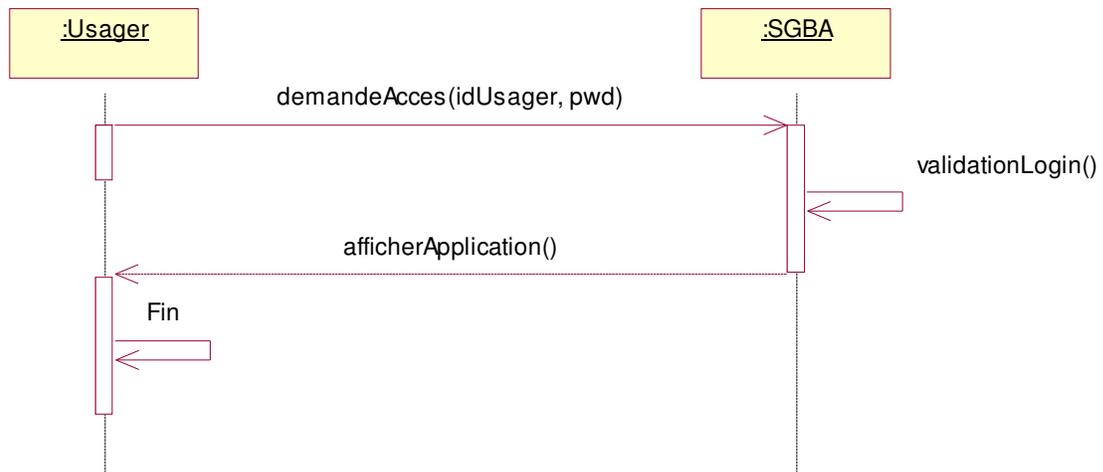
3.2.1.33 Générer un rapport



3.2.1.34 Afficher les statistiques du système



3.2.1.35 Authentifier un usager



3.2.2 Contrats

3.2.2.1 CO01 Ajouter un dossier d'utilisateur

Id :	CO01
Opération :	AjouterUtilisateur(nom, prenom, DDN, genre, adresse, tél, tailleFamille, sourceRevenus)
Références croisées :	UC01 : Ajouter un dossier d'utilisateur
Pré-condition :	
Post-condition :	L'utilisateur fait partie de la liste des premiers.

3.2.2.2 CO02 Mettre à jour un dossier d'utilisateur

Id :	CO02
Opération :	ModifierUtilisateur(utilisateur)
Références croisées :	UC02 : Mettre à jour un dossier d'utilisateur
Pré-condition :	L'utilisateur existe et est en cours d'édition. Les données saisies sont valides et rangées dans l'objet « utilisateur ».
Post-condition :	L'utilisateur est mis à jour dans la BD.

3.2.2.3 CO03 Supprimer un dossier d'utilisateur

Id :	CO03
Opération :	SupprimerUtilisateur(utilisateur)
Références croisées :	UC03 : Supprimer un dossier d'utilisateur
Pré-condition :	L'utilisateur existe.
Post-condition :	Le dossier de l'utilisateur n'existe plus. Les liens familiaux de l'utilisateur n'existent plus. Les résultats de l'utilisateur n'existent plus.

3.2.2.4 CO04 Réinscrire un utilisateur

Id :	CO04
Opération :	ChangerStatutUtilisateur(noUtilisateur, statut)
Références croisées :	UC04 : Réinscrire un utilisateur
Pré-condition :	L'utilisateur existe. Le statut de l'utilisateur est « à réinscrire ».
Post-condition :	Le statut de l'utilisateur est rétabli en fonction de la complétion de ses ateliers.

3.2.2.5 CO05 Rechercher un dossier d'utilisateur

Id :	CO05
Opération :	RechercherUtilisateur(no_utilisateur, nom, prenom, NAS, DDN, genre, adresse, tél, tailleFamille, sourceRevenus) (<i>plusieurs signatures, en fonction des champs utilisés</i>)
Références croisées :	UC05 : Rechercher un dossier d'utilisateur
Pré-condition :	Les champs passés à la fonction sont valides.
Post-condition :	Les utilisateurs correspondants aux champs passés sont retournés.

3.2.2.6 CO06 Afficher un dossier d'utilisateur

Id :	CO06
Opération :	AfficherUtilisateur(utilisateur)
Références croisées :	UC06 : Afficher un dossier d'utilisateur
Pré-condition :	L'utilisateur existe.
Post-condition :	Le dossier de l'utilisateur est affiché.

3.2.2.7 CO07 Rembourser un crédit

Id :	CO07
Opération :	RembourserCrédit(utilisateur, nbCrédits)
Références croisées :	UC07 : Rembourser un crédit
Pré-condition :	L'utilisateur existe. L'utilisateur a enregistré au moins un visite à la quelle un crédit est associé.
Post-condition :	Le remboursement du crédit est enregistré dans la BD.

3.2.2.8 CO08 Enregistrer la visite d'un utilisateur

Id :	CO08
Opération :	EnregistrerVisiteUtilisateur(no_utilisateur,date)
Références croisées :	UC08 : Enregistrer la visite d'un utilisateur
Pré-condition :	Un dossier utilisateur est en mode édition. Une action enregistrer une visite d'un utilisateur est en cours. La fonction VerifieEnregistrerVisiteUtilisateur(no_utilisateur) retourne ok.
Post-condition :	La date de la visite est enregistrée dans le dossier utilisateur.

3.2.2.9 CO09 Changer le statut des utilisateurs pour « à réinscrire »

Id :	CO09
Opération :	InactiverDossierUtilisateur(no_utilisateur)
Références croisées :	UC09 : Changer le statut des utilisateurs pour « à réinscrire »
Pré-condition :	La date du système atteint le moment voué à cet effet.
Post-condition :	Les statuts des utilisateurs sont « à réinscrire ».

3.2.2.10 CO10 Enregistrer un crédit à un utilisateur

Id :	CO10
Opération :	EnregistrerCreditUtilisateur (no_utilisateur, date)
Références croisées :	UC10 : Enregistrer un crédit à un utilisateur
Pré-condition :	Un dossier utilisateur est en mode édition. Le choix mode paiement est en option mode de crédit. La fonction VerifierCreditUtilisateur(no_utilisateur) retourne ok.
Post-condition :	La date du crédit est enregistré dans le système et le système retire de un le nombre de crédit disponible à l'utilisateur.

3.2.2.11 CO11 Consulter un compte d'utilisateur

Id :	CO11
Opération :	ConsulterUsager(no_utilisateur)
Références croisées :	UC11 : Consulter un compte d'utilisateur
Pré-condition :	La fonction VerifierExisteUsager(no_utilisateur) retourne ok.
Post-condition :	Le système retourne le dossier usager (idUsager).

3.2.2.12 CO12 Créer un compte d'utilisateur

Id :	CO12
Opération :	CreerUsager(no_utilisateur)
Références croisées :	UC12 : Créer un compte d'utilisateur
Pré-condition :	La fonction VerifierExisteUsager(no_utilisateur) retourne faux.
Post-condition :	Le système enregistre l'adresse, son nom et les fonctions de l'utilisateur.

3.2.2.13 CO13 Mettre à jour un compte d'utilisateur

Id :	CO13
Opération :	MiseJourUsager(no_utilisateur)
Références croisées :	UC13 : Mettre à jour un compte d'utilisateur
Pré-condition :	Le dossier usager est en mode de modification. Une action de modification du dossier est en cours.
Post-condition :	Le système enregistre les nouvelles données de l'utilisateur soit par les fonctions setAdresse(codePostal, rue, numero ... etc), setAjoutFonction(idFonction), setRetraitFonction(idFonction) setPwd(idMotPasse), GererPremission(no_utilisateur)

3.2.2.14 CO14 Supprimer un compte d'utilisateur

Id :	CO14
Opération :	SupprimerUsager(no_utilisateur)
Références croisées :	UC14 : Supprimer un compte d'utilisateur
Pré-condition :	La fonction VerifierExisteUsager(no_utilisateur) retourne ok.
Post-condition :	Le système supprime l'utilisateur.

3.2.2.15 CO15 Gérer le niveau de permission d'un usager

Id :	CO15
Opération :	GererPermission(no_utilisateur)
Références croisées :	UC15 : Gérer le niveau de permission d'un usager
Pré-condition :	Un dossier d'utilisateur est en édition.

Post-condition :	Le système ajoute ou supprime une permission d'un usager selon les fonctions suivantes AjoutPermission(idPermission), RetraitPermission(idPermission).
-------------------------	--

3.2.2.16 CO16 Obliger l'usager à changer son mot de passe

Id :	CO16
Opération :	changementDeMotDePasse(ancienMDP, nouveauMDP)
Références croisées :	UC16 : Obliger l'usager à changer son mot de passe
Pré-condition :	L'usager est authentifié auprès du système. Le nouveau mot de passe doit être différent de l'ancien et doit respecter les règles de création de mot de passe. (longueur de 6 caractères minimum)
Post-condition :	Le mot de passe a été changé dans la BD. L'usager doit s'authentifier de nouveau.

3.2.2.17 CO17 Créer un nouvel atelier

Id :	CO17
Opération :	creerAtelier(nomAtelier, lieuAtelier, horaireAtelier)
Références croisées :	UC17 : Créer un nouvel atelier
Pré-condition :	L'usager est authentifié auprès du système. L'atelier n'existe pas dans la BD.
Post-condition :	Le nouvel atelier est créé dans la BD.

3.2.2.18 CO18 Mettre à jour un atelier

Id :	CO18
Opération :	saisirDonnéesAtelier(listeParamètresAModifier)
Références croisées :	UC18 : Mettre à jour un atelier
Pré-condition :	L'usager est authentifié auprès du système. Un atelier a été recherché et affiché à l'usager.
Post-condition :	Les informations d'un atelier ont été modifiées. (nomAtelier, lieuAtelier, horaireAtelier)

3.2.2.19 CO19 Supprimer un atelier

Id :	CO19
Opération :	supprimerAtelier(nomAtelier, lieuAtelier)
Références croisées :	UC19 : Supprimer un atelier.

Pré-condition :	L'utilisateur est authentifié auprès du système. Un atelier a été recherché et affiché à l'utilisateur. L'utilisateur a confirmé la demande de suppression.
Post-condition :	L'atelier est supprimé de la BD.

3.2.2.20 CO20 Créer un groupe-cours

Id :	CO20
Opération :	creerGroupeCours(numeroGroupe, listeParticipants)
Références croisées :	UC20 : Créer un groupe cours.
Pré-condition :	L'utilisateur est authentifié auprès du système. Un atelier a été recherché et affiché à l'utilisateur. Le groupe cours n'existe pas dans la BD.
Post-condition :	Le groupe cours est créé dans la BD.

3.2.2.21 CO21 Mettre à jour un groupe-cours

Id :	CO21
Opération :	saisirDonnéesGroupeCours(listeParticipants)
Références croisées :	UC21 : Mettre à jour un groupe-cours
Pré-condition :	L'utilisateur est authentifié auprès du système. Un atelier a été recherché et affiché à l'utilisateur. Un groupe cours a été affiché à l'utilisateur.
Post-condition :	Les informations d'un groupe cours ont été modifiées dans la BD. (numeroGroupe, listeParticipants)

3.2.2.22 CO22 Supprimer un groupe-cours

Id :	CO22
Opération :	supprimerGroupeCours(numeroGroupe)
Références croisées :	UC22 : Supprimer un groupe cours
Pré-condition :	L'utilisateur est authentifié auprès du système. Un atelier a été recherché et affiché à l'utilisateur. Un groupe cours a été affiché à l'utilisateur. L'utilisateur a confirmé la demande de suppression.
Post-condition :	Le groupe cours est supprimé de la BD.

3.2.2.23 CO23 Inscrire un utilisateur à un atelier

Id :	CO23
Opération :	inscrireUtilisateurAtelier(idAtelier, idGroupe, idUtilisateur)
Références croisées :	UC23 : Inscrire un utilisateur à un atelier

Pré-condition :	Le dossier du client existe. L'atelier existe. L'atelier n'a pas encore eu lieu. L'utilisateur n'a pas encore réussi cet atelier.
Post-condition :	La liste des inscriptions à l'atelier est à jour.

3.2.2.24 CO24 Supprimer l'inscription d'un utilisateur à un atelier

Id :	CO24
Opération :	supprimerInscriptionAtelier(idAtelier, idGroupe, idUtilisateur)
Références croisées :	UC24 : Supprimer l'inscription d'un utilisateur à un atelier
Pré-condition :	Le dossier du client existe. L'atelier existe. L'atelier n'a pas encore eu lieu.
Post-condition :	La liste des inscriptions à l'atelier est à jour.

3.2.2.25 CO25 Saisir le résultat d'un utilisateur à un atelier

Id :	CO25
Opération :	entrerResultatAtelier(res, idUtilisateur)
Références croisées :	UC25 : Saisir le résultat d'un utilisateur à un atelier
Pré-condition :	L'atelier existe. L'atelier a eu lieu. L'utilisateur a le droit d'accéder aux résultats.
Post-condition :	La liste des résultats de l'atelier est affichée.

3.2.2.26 CO26 Obtenir la liste des inscriptions à un atelier

Id :	CO26
Opération :	afficherListeUtilisateurAtelier(idAtelier, idGroupeCours)
Références croisées :	UC26 : Obtenir la liste des inscriptions à un atelier
Pré-condition :	Le dossier du membre du personnel pédagogique existe. L'utilisateur a le droit d'accéder à la liste. L'atelier existe.
Post-condition :	La liste des inscriptions est affichée.

3.2.2.27 CO27 Créer une activité

Id :	CO27
Opération :	creerActivité(idActivité, description)

Références croisées :	UC27 : Créer une activité
Pré-condition :	
Post-condition :	La liste des activités est à jour.

3.2.2.28 CO28 Mettre à jour une activité

Id :	CO28
Opération :	modifierActivité(idActivité, description, horaire, nomResponsable, info)
Références croisées :	UC28 : Mettre à jour une activité
Pré-condition :	L'activité existe.
Post-condition :	La liste des activités mise est à jour.

3.2.2.29 CO29 Supprimer une activité

Id :	CO29
Opération :	supprimerActivite(idActivite)
Références croisées :	UC29 Supprimer une activité
Pré-condition :	L'activité existe.
Post-condition :	La liste des activités mise est à jour. S'il y avait des inscriptions, elles seront annulées.

3.2.2.30 CO30 Inscrire un utilisateur à une activité

Id :	CO30
Opération :	inscrireUtilisateurActivité(idUtilisateur)
Références croisées :	UC30 : Inscrire un utilisateur à une activité
Pré-condition :	L'utilisateur existe. L'activité existe. L'utilisateur n'est pas déjà inscrit à l'activité.
Post-condition :	Liste des inscriptions du groupe-cours contient l'utilisateur.

3.2.2.31 CO31 Supprimer l'inscription d'un utilisateur à une activité

Id :	CO31
Opération :	Supprimer_utilisateur_activité (Utilisateur, Activité)
Références croisées :	UC31 : Supprimer l'inscription d'un utilisateur à une activité
Pré-condition :	L'utilisateur est inscrit à l'activité appropriée. La date de l'activité n'a pas été dépassée.
Post-condition :	Liste des inscriptions de l'activité ne contient plus l'utilisateur.

3.2.2.32 CO32 Obtenir la liste des inscriptions à une activité

Id :	CO32
Opération :	obtenirListeInscription(idActivité)
Références croisées :	UC32 : Obtenir la liste des inscriptions à une activité
Pré-condition :	L'activité existe. Il y a au moins 1 utilisateur inscrit à l'activité.
Post-condition :	

3.2.2.33 CO33 Générer un rapport

Id :	CO33
Opération :	genererRapport(Rapport rapport)
Références croisées :	UC33 : Générer un rapport
Pré-condition :	Les différents champs d'informations existent.
Post-condition :	

3.2.2.34 CO34 Afficher les statistiques du système

Id :	CO34
Opération :	genererStats(Statistique stats)
Références croisées :	UC34 : Afficher les statistiques du système
Pré-condition :	Les différents champs d'informations existent.
Post-condition :	Les statistiques produites représentent fidèlement les informations contenues dans le SGBA.

3.2.2.35 CO35 Authentifier un usager

Id :	CO35
Opération :	demandeAcces(idUsager, pwd)
Références croisées :	UC35 : Authentifier un usager
Pré-condition :	
Post-condition :	Si les informations d'authentification sont valides, l'utilisateur est authentifié.

3.3 Exigences d'opérations, de communications et de performance

3.3.1 Nombre de terminaux à supporter

Nous estimons que le nombre de terminaux que le SGBA devra être en mesure de supporter s'avérera fort restreint, c'est-à-dire de l'ordre d'une demi-douzaine de terminaux à la fois, en raison de la nature même des organismes – une organisation centralisée de taille modeste. Il n'est cependant pas impossible d'imaginer un nombre supérieur de stations clientes advenant le cas où le SGBA serait déployé dans un environnement de plus grande envergure, mais les contraintes seront alors celles du SGBD, que l'architecture du SGBA devra tenter de soulager autant que possible.

3.3.2 Nombre d'utilisateurs simultanés

Une utilisation du SGBA par un organisme typique entraînerait une utilisation de la base de données centrale par de trois à cinq utilisateurs simultanés. Nous ne croyons pas que cette exigence s'avère bien difficile à atteindre, en raison de la puissance des machines actuelles, d'autant plus que la machine la plus performante devrait être réservée à la base de données.

3.3.3 Quantité et types d'information à traiter

Le modèle de données sur SGBA est conçu pour respecter la contrainte qui consiste à stocker le moins d'informations possible, afin de conserver de l'espace disque, quitte à sacrifier légèrement la performance de l'ensemble, en raison des limites imposées par le SGBD choisi (telles que la taille de la base ne pouvant dépasser quatre giga-octets). Cependant, une attention particulière doit être portée à la protection des données stockées, en raison de la nature éminemment privée de celles-ci.

3.3.4 Nombre de transactions à traiter par unité de temps sous conditions normales et en période de pointe

Sans être d'influence négligeable, nous ne considérons pas que le nombre de transactions à traiter s'avère un critère de conception majeur, pour les raisons que nous avons citées à maintes reprises à travers ce document et le concept des opérations : un programme déployé sur une courte distance qui permet d'ignorer les contraintes de latence réseau, qui échange peu d'informations et dont le traitement sera effectué majoritairement par les terminaux clients.

3.4 Exigences logiques de bases de données

Dans le cadre du SGBA, toutes les informations stockées au sein de la base de données relèveront directement des données reliées aux personnes et aux activités manipulées par le système, sans concerner le logiciel client (pas de données conservées en ce qui a trait à l'affichage ou aux couches applications et métier) ou les règles d'affaires (déployées du côté client), à l'exception des comptes d'utilisateurs de ceux qui seront appelés à manipuler le logiciel.

La base de données utilise une approche consistant à stocker aussi peu d'informations qu'il est possible, en éliminant tout champ qu'il sera possible de calculer, ce qui réduit la taille des données stockées sans trop augmenter le temps de traitement. Vu le nombre restreint d'accès simultanés, nous n'avons pas estimé utile d'implanter une solution de marquage de données suivi d'une suppression ultérieure – toute donnée traitée par le SGBA voit ses changements répercutés « en dur » sur la base de données, aussitôt que possible, au moyen, bien sûr, d'un mécanisme de verrou et de sélection des rangées pour traitement, géré par le gestionnaire de persistance.

3.4.1 Types d'informations utilisées par les différentes fonctions

La majorité des informations utilisées seront des chaînes de caractères, mais il y aura aussi des entiers et des dates. Ces informations seront converties en objets métiers par le gestionnaire de persistance dont la tâche sera d'effectuer la conversion objet/relationnel et d'assurer l'obtention ainsi que le stockage des données utiles.

3.4.2 Fréquence d'utilisation

La fréquence d'utilisation de la base sera directement proportionnelle au nombre de terminaux connectés. La base sera accédée lors de la consultation des dossiers, de l'ajout ou de la modification d'informations et lors de l'accès « login » d'un usager.

3.4.3 Capacité d'accès

La capacité d'accès sera déterminée par plusieurs contraintes, tant physiques que logiques, en particulier si le matériel utilisé pour soutenir la base de données s'avérait de mauvaise qualité, ce qui le transformerait en goulot d'étranglement. La capacité maximale de quatre giga-octets de la base (SGBD Oracle édition expresse) sera une contrainte à ne pas oublier, mais ne représentera pas un problème considérable au niveau de l'espace disque.

De manière générale, la capacité d'accès ne représente pas un problème significatif, en raison de la nature locale du déploiement planifié du SGBA. Advenant le cas où la base de données serait déployée en un site distant, la bande passante se devra d'être

Utilisateur : Table contenant les informations personnelles des utilisateurs, à l'exception des membres de leurs ménages (table Liens Familiaux). Pour des raisons de confidentialité et dans le but de réduire les risques en cas de fuite d'informations privilégiées, le jour de naissance des utilisateurs sera omis. Les villes possibles devront être gérées par l'application au niveau client, ce qui élimine le besoin d'une table « ville » séparée.

no_utilisateur	INTEGER	NOT NULL
NAS	VARCHAR2	
nom	VARCHAR2	NOT NULL
prenom	VARCHAR2	NOT NULL
mois_naissance	CHAR	NOT NULL
annee_naissance	CHAR	NOT NULL
genre	CHAR	NOT NULL
telephone	VARCHAR2	
nb_sacs_permis	INTEGER	NOT NULL
source_revenus	VARCHAR2	
adresse_civique	VARCHAR2	NOT NULL
code_postal	VARCHAR2	NOT NULL
ville	VARCHAR2	NOT NULL

Historique_Statut_Utilisateur : Table stockant l'historique des statuts des utilisateurs, c'est-à-dire, pour chacun des utilisateurs, le statut qu'il s'est vu décerner, la date d'obtention, et un commentaire justifiant l'obtention dudit statut.

no_utilisateur	INTEGER	NOT NULL
date_changement	DATE	NOT NULL
statut_accorde	VARCHAR2	NOT NULL
commentaires	VARCHAR2	

Type_Lien_Familial : Table spécifiant, pour un lien familial, le type de lien qui lui est associé – du point de vue de l'utilisateur. Elle permet d'éviter une clause « check » qui serait inutilement complexe.

no_type_lien_familial	INTEGER	NOT NULL
description	VARCHAR2	NOT NULL
genre_associe	CHAR	NOT NULL

Liens_Familiaux : Table représentant les membres de la famille immédiate d'un utilisateur (et, de manière générale, tous ceux qui vivent sous son toit). Le nombre et la nature des liens familiaux seront déterminants dans le choix du nombre de sacs de nourriture accordés à l'utilisateur lors d'une visite.

no_utilisateur	INTEGER	NOT NULL
no_lien	INTEGER	NOT NULL

nom	VARCHAR2	NOT NULL
prenom	VARCHAR2	NOT NULL
mois_naissance	CHAR	NOT NULL
annee_naissance	CHAR	NOT NULL
genre	CHAR	NOT NULL
type_lien_familial	INTEGER	NOT NULL

Visite : Une visite représente la collecte de nourriture d'un utilisateur. Cette table conserve un historique complet des visites de tous les utilisateurs.

no_visite	INTEGER	NOT NULL
no_utilisateur	INTEGER	NOT NULL
date_visite	DATE	NOT NULL
notes_personnel	VARCHAR2	

Credit : Table associée à une visite qui enregistre un crédit en argent qui aurait été accordé lors de celle-ci. La date de remboursement sera également conservée.

no_visite	INTEGER	NOT NULL
remboursement	DATE	

Usager : Table contenant les informations personnelles des usagers du système.

no_usager	INTEGER	NOT NULL
no_employe	VARCHAR2	NOT NULL
nom	VARCHAR2	NOT NULL
prenom	VARCHAR2	NOT NULL
nom_compte	VARCHAR2	NOT NULL
mot_passe	VARCHAR2	NOT NULL
date_mot_passe	DATE	NOT NULL

Permission_Usager : Table conservant le niveau de permission de l'utilisateur – cette table est distincte de la table « usager » pour des raisons de sécurité, notamment face à des attaques du type « SQL Injection ».

no_usager	INTEGER	NOT NULL
permission	VARCHAR2	NOT NULL

Historique_Mot_Passe : Table conservant l'historique des mots de passe utilisés par chacun des usagers.

no_historique	INTEGER	NOT NULL
---------------	---------	----------

no_usager	INTEGER	NOT NULL
rang	INTEGER	NOT NULL
mot_passe	VARCHAR2	NOT NULL

Audit_Operations : Table conservant l'historique de toutes les opérations effectuées par les usagers ayant un impact sur le système. Il s'agit d'une description textuelle, ce qui la rend fort flexible en ce qui concerne les informations pouvant y être conservées.

no_operation	INTEGER	NOT NULL
no_usager	INTEGER	NOT NULL
description	VARCHAR2	NOT NULL
moment	DATE	NOT NULL

Atelier : Table contenant les descriptions liées aux ateliers de l'organisme bénéficiaire.

no_atelier	INTEGER	NOT NULL
description	VARCHAR2	

Groupe_Cours_Atelier : Table contenant les groupe-cours d'ateliers, mais pas les inscriptions aux ateliers, ni les résultats obtenus par les utilisateurs.

no_groupe_cours	INTEGER	NOT NULL
no_atelier	INTEGER	NOT NULL
no_usager	INTEGER	NOT NULL
date_atelier	DATE	NOT NULL
lieu	VARCHAR2	NOT NULL
capacite	INTEGER	NOT NULL
commentaires	VARCHAR2	

Inscription_Atelier : Table contenant les inscriptions aux groupe-cours (et, transitivement, aux ateliers) et les résultats.

no_groupe_cours	INTEGER	NOT NULL
no_utilisateur	INTEGER	NOT NULL
resultat	CHAR	

Activite : Table contenant les informations liées aux activités.

no_activite	INTEGER	NOT NULL
description	VARCHAR2	NOT NULL
date_activite	DATE	NOT NULL
lieu	VARCHAR2	NOT NULL

capacite	INTEGER	NOT NULL
commentaires	VARCHAR2	
no_usager	INTEGER	NOT NULL

Inscription_Activite : Table contenant les inscriptions aux activités.

no_activite	INTEGER	NOT NULL
no_utilisateur	INTEGER	NOT NULL

3.4.5 Contraintes d'intégrité

Les contraintes d'intégrité sont exprimées sur le modèle relationnel cité au point précédent, dans l'objectif de maintenir la base de données dans un état stable et contenant des données uniformes. Le mécanisme de respect des contraintes intégré au SGBD sera utilisé dans cette optique, ce qui ne sera pas sans conséquences au niveau du gestionnaire de persistance de données, qui devra intégrer ce respect et « savoir » comment réagir en cas d'erreur lancée par la base de données.

3.4.6 Exigences de rétention de données

Les exigences de rétention de données, dans le contexte du SGBA, concerneront essentiellement le niveau de cryptage des informations stockées ainsi que la durée durant laquelle lesdites données seront conservées. Au moment d'écrire ces lignes, une politique formelle en terme de mois ou d'années n'a pas encore été établie, et nous croyons qu'elle relève directement des normes que s'imposeront l'organisme bénéficiaire.

Par contre, pour ce qui a trait au cryptage des informations de la base de données, le SGBA aura recours aux services de chiffrement transparent offert par le SGBD, ce qui peut contribuer à réduire les risques de fuite d'informations au cas où la base de données principale (ou l'une des copies de sauvegarde) serait volée ou accédée par un utilisateur non autorisé.

3.5 Contraintes de conception

Les contraintes pesant sur la conception du SGBA seront essentiellement entraînées par celles de son environnement d'exploitation, tant au niveau du matériel que des données manipulées : une clientèle peu familière avec l'exploitation d'un logiciel de gestion, un équipement parfois peu fiable en raison d'une usure avancée et l'absence de moyens financiers qui pourrait mettre en péril la remise sur pied du système advenant un problème technique.

3.5.1 Contraintes de matériel

Le SGBA étant, par définition, un système réparti, l'exploitation de différentes machines sera souhaitable, si les moyens financiers de l'organisme le permettent, que nous pouvons décrire ici en quelques lignes :

- Serveur de base de données : la machine qui sera la plus sollicitée par l'exploitation du SGBA, le serveur de base de données servira d'entrepôt central d'informations, auquel tous les terminaux devront se connecter pour pouvoir fonctionner. Hormis l'utilisation d'une base de données Oracle et d'un réseau Ethernet, aucune contrainte particulière ne régit cette machine, sinon la possession d'une quantité de mémoire vive aussi grande que l'on peut obtenir et d'un disque dur de taille raisonnable.⁶
- Terminal : aucune contrainte particulière n'est à prévoir au niveau des terminaux, sur lesquels le SGBA sera déployé, excepté la capacité de rejoindre la base de données centrale, bien entendu.

3.5.2 Format des rapports

Les rapports produits par le SGBA se devront d'offrir au moins autant d'informations que celles qui étaient produites avant son implantation. Le paramétrage de ceux-ci sera laissé à la disposition de l'organisme, le système devant donc être développé dans une optique de flexibilité et de personnalisation par l'administrateur de l'application. Comme nous l'avons déjà mentionné, ces paramètres relèvent davantage de la partie de locale du logiciel que dans la base de données, et pourrait être reléguée à une version ultérieure du produit, la version initiale ne possédant qu'un nombre réduit, quoique suffisant, des rapports paramétrables offerts par défaut, qui devraient couvrir les besoins de la majorité des bénéficiaires.

Les rapports devront pouvoir être imprimés sur différents formats papiers, notamment les formats Lettre US et Légal US.

⁶ De notre point de vue, possédant un espace de stockage d'au moins 8 giga-octets, sans compter le système d'exploitation et les outils nécessaires au fonctionnement du SGBD.

3.5.3 Nomenclature des données

Comme nous l'avons déjà mentionné à maintes reprises à travers ce document, l'application sera basée sur un modèle réparti par couches. La base de données sera la couche la plus basse, à laquelle les autres couches parleront, un niveau à la fois, au moyen du gestionnaire de persistance. Contrairement à la base de données qui sera sur sa propre machine, toutes les autres couches seront déployées en tant qu'application sur la machine cliente (terminaux). Une couche application sera chargée de l'exécution des cas d'utilisation, tout en dialoguant avec le gestionnaire de sécurité pour assurer le droit d'exécution de toute fonction (en plus de permettre à une extension future de perfectionner le modèle de sécurité) et la couche métier. La couche interface sera au plus haut niveau et complètement découplée des autres, au moyen de techniques inspirées des patrons architecturaux permettant ces séparations.

Nous prévoyons que le SGBA sera développé en langage Java.

3.5.4 Procédures de comptabilité

La compatibilité ne sera pas une contrainte majeure de conception pour le SGBA, étant donné qu'il sera voué à remplacer un système manuel et qu'il ne communique avec aucune application externe. Cependant, il sera nécessaire de conserver une compatibilité totale entre les données des versions de l'application même, du moins au point de vue « terminal » (c'est-à-dire que le terminal SGBA pourra communiquer avec la base de données, peu importe les mises à jour que pourrait subir l'application cliente, quitte à restreindre ou empêcher l'utilisation de certaines fonctions dont l'entrepôt de données ne serait pas déployé), mais idéalement dans le deux sens.

3.5.5 Exigences d'audit, traçabilité des traitements

Au niveau de l'audit, chacune des opérations effectuées par les usagers sera emmagasinée au moyen d'une description textuelle, tant au niveau de l'écriture que de la consultation, ce qui permettra de diminuer les risques de fuites de données si des opérations suspectes étaient détectées, et ce, dès la version initiale. Cependant, la consultation des archives d'opérations ne disposera pas d'une interface propre pour cette version, qui sera implantée lors d'une livraison ultérieure, ce qui ne restreint pas la vérification des données d'audit par l'administrateur système au moyen de la console du SGBD. En plus d'une description texte, chaque opération sera jumelée à l'utilisateur concerné et horodaté du moment de la transaction.

La conception d'un module voué à l'extraction et à l'analyse des informations d'audit, destiné à l'administrateur ou aux gestionnaires, pourrait aisément faire l'objet d'un développement ultérieur, et s'avérerait un puissant argument en faveur du SGBA pour un organisme soucieux de la rigueur de la confidentialité des données qu'elle manipule et voulant s'assurer de la loyauté de ses employés et bénévoles. Le système que nous décrivons ici sera tout à fait en mesure de s'accommoder facile face à pareil ajout.

3.6 Exigences non-fonctionnelles

3.6.1 Fiabilité

La fiabilité du SGBA devra être irréprochable en ce qui concerne les données conservées, en particulier pour la régularité des visites et le nombre de sacs qu'un utilisateur peut emporter lors de chacune. Les sauvegardes devront être faites en conséquence, particulièrement dans un contexte de support de stockage de mauvaise qualité. De plus, les rapports permettront de confirmer la validité générale des informations contenues par le système par rapport aux moyennes et statistiques connues avant l'implantation.

3.6.2 Disponibilité

Le SGBA se devra d'être complètement disponible à l'intérieur des heures d'ouverture de l'organisme bénéficiaire (que nous estimons à environ 60 heures par semaine, de jour et de soir uniquement). Il ne sera pas nécessaire d'instaurer une politique d'accès 24 heures sur 24 pour un système non accessible de l'extérieur.

3.6.3 Sécurité

L'accès au SGBA devra être sécurisé, ce document l'a déjà mentionné, et cet accès se fera au moyen de mots de passe et de niveaux de permissions qui permettront de restreindre l'accès aux fonctions du système selon les besoins. À ce contrôle des accès, un système d'audit gardera trace de toutes les actions de tous les utilisateurs, ce qui permettrait de faciliter la localisation d'une fuite à l'interne (employé « taupe » de groupes d'intérêts autres). Finalement, les échanges et le stockage des informations devront être chiffrés.

3.6.4 Rendement

Le rendement ne semble guère représenter une difficulté pour le SGBA. Les informations devront être échangées suffisamment vite pour assurer une utilisation agréable, soit en moins de 5 secondes.

3.6.5 Portabilité

L'utilisation de la technologie Java pour le développement du SGBA assure une portabilité sans changements vers virtuellement tous les systèmes d'exploitation (en réalité, ceux pour lesquels une machine virtuelle Java récente a été portée et disposant d'une pile TCP/IP). Cependant, le SGBD Oracle XE n'est pour l'instant disponible que pour les systèmes Microsoft Windows et Linux.

4. Informations complémentaires

4.1 Annexe : aperçu de l'interface

4.1.1 Menu principal (Personnel administratif)

The screenshot shows a software window titled "Menu Utilisateur". At the top, there is a search bar with the text "RechercherUtilisateur\MenuModifierUtilisateur". Below this is a navigation bar with several icons and the title "Gestion des dossiers utilisateurs".

The main interface is divided into several sections:

- Left sidebar:** Contains five icons with corresponding text: "Modifier le dossier d'un utilisateur", "Inscrire ou modifier une activité", "Inscrire ou modifier un atelier", "Enregistrer ou modifier une visite", and "Supprimer un Utilisateur".
- Form area:** Contains input fields for:
 - Numéro d'assurance sociale:
 - Nom de l'utilisateur:
 - Prénom de l'utilisateur:
 - Date de naissance:
 - Téléphone:
 - Adresse:
 - Code postal:
 - Ville:
 - Genre: (dropdown menu)
 - Nombre de sacs: (dropdown menu)
- Tables:** There are four tables with scrollable content:
 - Table 1: Columns: Activité, Date, Lieu
 - Table 2: Columns: Nom, Prénom, Date de naissance, Lien
 - Table 3: Columns: Note, Atelier, Date
 - Table 4: Columns: Date, Statut, Commentaire
 - Table 5: Columns: Numéro, Date, Notes
 - Table 6: Columns: Numéro de la visite, Date de remboursement
- Decorative element:** A 3D graphic of three stylized human figures in green, blue, and yellow is located on the right side of the form area.

Figure 3 : Menu principal du personnel administratif (ébauche)

4.1.2 Gestion des utilisateurs – Recherche et création d'un dossier



Figure 4 : Ébauche du menu de gestion des utilisateurs

4.1.3 Modification d'un utilisateur (aperçu)

Menu Utilisateur

RechercherUtilisateur\NouvelUtilisateur

Gestion des dossiers utilisateurs

Enregistrer le nouvel utilisateur

Numéro d'assurance sociale:

Nom de l'utilisateur:

Prénom de l'utilisateur:

Date de naissance:

Téléphone:

Adresse:

Code postal:

Ville:

Genre: Nombre de sacs:

	Nom	Prénom	Date de naissance	Lien
*				

	Date	Statut	Commentaire
*			

Figure 5 : Ébauche de la création d'un nouvel utilisateur

4.1.4 Traitement des crédits (aperçu)

Menu Utilisateur

RechercherUtilisateur\MenuModifierUtilisateur\Visite

Gestion des dossiers utilisateurs

Numéro d'assurance sociale:

Nom de l'utilisateur:

Prénom de l'utilisateur:

Date de naissance:

	Numéro visite	Date	Notes	Crédit	Remboursé	Date de remboursement
*				<input type="checkbox"/>	<input type="checkbox"/>	

Figure 6 : Ébauche de l'enregistrement des visites

4.2 Exemple de script Oracle XE

-- Création des tables et des contraintes d'intégrité référentielle

```
CREATE TABLE Utilisateur
(no_utilisateur INTEGER NOT NULL,
NAS          VARCHAR2(10),
nom          VARCHAR2(50) NOT NULL,
prenom      VARCHAR2(50) NOT NULL,
mois_naissance CHAR(2) NOT NULL,
annee_naissance CHAR(4) NOT NULL,
genre       CHAR(1) NOT NULL,
telephone   VARCHAR2(14),
nb_sacs_permis INTEGER DEFAULT 1 NOT NULL,
source_revenus VARCHAR2(20),
adresse_civique VARCHAR2(100) NOT NULL,
code_postal  VARCHAR2(6) NOT NULL,
ville        VARCHAR2(30) NOT NULL,
PRIMARY KEY (no_utilisateur))
/

CREATE TABLE Historique_Statut_Utilisateur
(no_utilisateur INTEGER NOT NULL,
date_changement DATE NOT NULL,
statut_accorde VARCHAR2(15) NOT NULL CHECK (statut_accorde IN ('Conditionnel',
'Actif', 'Inactif', 'A reinscrire')),
commentaires  VARCHAR2(100),
PRIMARY KEY (no_utilisateur, date_changement),
FOREIGN KEY (no_utilisateur) REFERENCES Utilisateur(no_utilisateur))
/

CREATE TABLE Type_Lien_Familial
(no_type_lien_familial INTEGER NOT NULL,
description          VARCHAR2(15) NOT NULL,
genre_associe       CHAR(1) NOT NULL,
PRIMARY KEY (no_type_lien_familial))
/
```