

الجمهورية الجزائرية الديمقراطية الشعبية  
**République Algérienne Démocratique et Populaire**  
وزارة التعليم العالي و البحث العلمي  
**Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique**

Réf : /13.

**CENTRE UNIVERSITAIRE DE MILA  
INSTITUT DES SCIENCES ET DE LA TECHNOLOGIE**

**Mémoire de fin d'étude**  
Présenté pour l'obtention du diplôme de  
**LICENCE ACADEMIQUE**

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Informatique**

*Thème*

**La gestion de la scolarité automatisée au sein du**

**Centre Universitaire de Mila**

Presenté par : - Kassah Laouar Seifeddine.  
- Benguesmia Samir.

Sous la direction de : Hettab Abdelkamel.

*Année universitaire 2012-2013*

# *Remerciement*

*Nous tenons à remercier notre encadreur, Mr Hettab Abdelkamel, pour nous avoir orientés et de nous avoir offert l'opportunité de travailler sur ce projet.*

*Nous le remercions pour son grand soutien pédagogique, scientifique et moral, pour ses conseils, ses suggestions et ses encouragements qu'il n'a cessé de nous apporter pendant toute la durée de l'élaboration de ce mémoire.*

*Nous tenons à remercier et à saluer tous les professeurs du département science et technologie, et particulièrement tout les professeurs du 3 eme année informatique pour son aide.*

*Enfin un grand merci à Mohamed et ses amis du master informatique pour leur aide et leur soutien qu'ils nous ont accordé.*

*Merci à tous.*

# *Dédicaces*

*Je dédie ce modeste travail aux êtres les plus chers dans ma vie, mes parents et tous ma famille et spécialement mes grand mère et père, qui je doit les remercier beaucoup pour leur soutien morales matériel, leur compréhension, leur tendresse, leur amour et leurs acritique.*

*À mon frère Zineddine.*

*À ma sœur Sondosse.*

*À mes très chères amies Razak, Hamza, Ahmed, Amine, Mousab, Radouane, Yazid, Boujema, Ryad,*

*À tous mes enseignants du primaire jusqu'a aujourd'hui.*

*À mon binôme Samir ainsi que toute sa famille.*

*À mon chère amis issa te dieu de garde avec sa grande miséricorde dans sont vaste paradis.*

*A tous qui m'aime.*

*À tous ceux qui me connaissent.*

*Seifeddine.*

## *Introduction générale*

L'informatique est une science récente, un nouveau monde apparu seulement depuis quelques dizaines d'années, qui se développe de jour en jour afin de nous faciliter les tâches dans tous les domaines par sa rapidité de traitement de l'information. L'informatique se charge de traiter des informations par un support spécifique de manière automatique et systématique avec un maximum de rationalité en vue d'en obtenir des résultats. Ces informations sont sous forme de textes, d'images ou encore de sons, et peuvent concerner tous les domaines, elle permet donc la manipulation, la gestion, et le stockage de l'information.

Cette science est née au besoin de l'homme à mieux maîtriser les processus d'évaluation dont elle a connu un développement extraordinaire dans ces dernières années. L'utilisation des outils informatiques offre beaucoup d'avantages, parmi ces avantages on peut citer :

- Possibilité de manipulation et traitement rapide d'un grand nombre d'information.
- Eviter les recherches fastidieuses dans les fichiers.
- Sécurisation des informations.

La réforme dite "réforme LMD" initiée et mise en œuvre dans notre pays depuis l'année 2003/2004, vise à harmoniser notre système de formation supérieure avec le reste du monde, à lui assurer la possibilité de jouer son rôle de levier du développement social et économique basé sur la recherche et l'innovation, et à le rendre capable de contribuer à relever les défis de la société du savoir.

La réforme LMD, articulée sur les trois niveaux de formation : Licence – Master – Doctorat, est venue pour répondre aux objectifs suivants :

- Assurer une formation de qualité, en prenant en charge la satisfaction de la demande sociale, légitime, en matière d'accès à l'enseignement supérieur ;
- Réaliser une véritable osmose avec l'environnement socio-économique en développant toutes les interactions possibles entre l'université et le monde qui l'entoure ;
- Développer les mécanismes d'adaptation continue aux évolutions des métiers .
- Consolider sa mission culturelle par la promotion des valeurs universelles qu'exprime l'esprit universitaire, notamment celles de la tolérance et du respect de l'autre.
- Asseoir les bases d'une bonne gouvernance fondée sur la participation et la concertation.

La nouvelle architecture retenue pour l'Enseignement Supérieur est articulée selon trois paliers de formation, correspondant chacun à un diplôme :

- Le niveau Licence, correspondant à un cycle de formation de trois années après le baccalauréat ;
- Le niveau Master, correspondant à 2 années supplémentaires après le niveau Licence .
- Le niveau Doctorat, correspondant à 3 années supplémentaires après le niveau Master.

### **Motivation :**

La gestion de la délibération est une tâche critique et pointue parmi les fonctions de l'institut. Cette tâche est devenue un peu plus compliquée qu'avant avec le système LMD car un étudiant peut étudier à cheval de plusieurs années (date). Les modules sont couronnées de plusieurs paramètres tel que le coefficient, le crédit, l'unité etc. la délibération passe aussi par plusieurs étapes: PV de matière, PV d'unité, PV de semestre etc. et ils existent plusieurs types de PV: PV ordinaire, PV étudiants avec dettes etc. Donc, on a besoin d'une application informatique qui permet, une fois les données concernant les étudiants et les modules sont saisies, de produire les différents PVs. Nous signalons ici l'existence déjà d'une application informatique qui est responsable de la tâche de délibération automatique. Cette application souffre d'un certain nombre d'anomalies et rend parfois la délibération automatique plus difficile à vérifier qu'une délibération qu'on peut faire manuellement.

### **Objectif :**

L'objectif de notre projet est la gestion scolaire pour le centre universitaire de MILA.

# Chapitre

## Langage de modélisation

- 1-Concepts UML.*
- 2 -Modélisation avec UML.*
- 3-Processus unifié (UP)*
- 4-Les phases.*
- 5-L'UP simplifier.*
- 6-Identification des besoins et spécification des fonctionnalités.*
- 7-Phase de conception.*
- 8-Phases d'analyse.*
- Conclusion.*

1

## Introduction

### « UNIFIED MODELING LANGUAGE »

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifié à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. UML a une dimension symbolique et ouvre une nouvelle voie d'échange de visions systémiques précises. Ce langage est certes issu du développement logiciel mais pourrait être appliqué à toute science fondée sur la description d'un système. Dans l'immédiat, UML intéresse fortement les spécialistes de l'ingénierie système.

## 1. Concepts UML :

### 1.1. Historique des méthodes de modulation :

- Les années 1980 : utilisation de méthodes adaptées à la programmation impérative (notamment Merise).
- Fin 80/début 90 la programmation par objets prend de l'importance.
- Conséquence naturelle : mise en place de méthodes orientées objets. Plus de cinquante méthodes apparaissent entre 1990 et 1995 :
  - OOD : Objet Oriented Design (Booch, 1991)
  - OOM : Objet oriented Merise (Bouzeghoub, Rochfeld, 1993)
- 1994 : le consensus se fait autour de trois méthodes :
  - OMT (Objet Modeling Technique) de James Rumbaugh fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système.
  - OOD (Objet Oriented Design) de Grady Booch, définie pour le département of Defense (USA), introduit le concept de paquetage (package).
  - OOSE (Objet Oriented Software Engineering) d'Ivar Jacobson fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation).
- 1995 : Fusion des 3 principales méthodes pour définir un langage de modélisation commun : UML (Unified Modeling Language) [2].

## 1.2. Pourquoi UML ?

UML comme tout autre outil est née pour des raisons intéressantes, parmi ces raisons on trouve les suivantes qui semblent les plus importantes :

### 1.2.1. Besoins d'une norme pour les informations :

L'informatique est restée pour longtemps la seule industrie qui ne disposait pas d'une norme universelle pour la présentation de ses composants, produits et services.

### 1.2.2. UML pour l'inconvénient d'objet :

Parmi les principales raisons de l'apparition d'UML on trouve les inconvénients des méthodes objets :

- L'intuitive et l'approche fonctionnelle :

L'approche objet est moins intuitive que l'approche fonctionnelle : Malgré les apparences, il est plus naturel pour l'esprit humain de décomposer un problème informatique sous forme d'une hiérarchie de fonctions atomiques et de données qu'au terme d'objets et d'interactions entre ces objets [4].

- Approche objet et langage orientée objet :

Les langages orientés objets ne sont que des outils qui proposent une manière d'implémenter des concepts objet, ils ne valident en rien l'utilisation des moyens techniques pour concevoir un système conforme à la philosophie objet [4].

### 1.2.3. UML pour d'autres raisons :

Le langage de modélisation UML est créé pour d'autres raisons, tel que :

- Unifier les deux méthodes OMT et OOD au début.
- La modélisation objet n'est pas assez suffisante.
- Les langages de modélisations ont besoins d'un support.
- Besoins d'utilisation (lisibilité pour utilisateurs et facilite pour concepteurs).

- Fournir un moyen performant qui facilite le développement de l'informatique.
- Pour l'utilisation correcte de l'orienté objet.

En effet UML est une solution incontournable, définie pour être un standard de modélisation, facilite la compréhension et la communication d'une modélisation objet, tel que même les non spécialistes peuvent utiliser et développer les modèles UML sans un grand effort [4].

### **1.3. A quoi sert UML ?**

- UML n'est pas une méthode ou un processus !
  - Si l'on parle de méthode objet pour UML, c'est par abus de langage !
  - Ce constat vaut aussi pour OMT ou d'autres techniques / langages de modélisation.
- UML est un langage graphique : c'est un ensemble de diagrammes très complexes [5].

### **1.4. Les points forts d'UML :**

- UML est un langage formel et normalisé
  - gain de précision
  - gage de stabilité
  - encourage l'utilisation d'outils
- UML est un support de communication performant
  - Il cadre l'analyse.
  - Il facilite la compréhension de représentations abstraites complexes.

Son caractère polyvalent et sa souplesse en font un langage universel [5].

### **1.5. Les points faibles d'UML :**

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

- Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue.
- Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard [5].

## **2. Modélisation avec UML :**

### **2.1. Qu'est-ce qu'un modèle ?**

- Un modèle est une abstraction de la réalité.
- L'abstraction est un des piliers de l'approche objet.
- La modélisation consiste à créer une représentation simplifiée d'un problème en le décrivant (activité d'analyse) et en résolvant sa solution (activité de conception).

### **2.2. Les différents types de diagrammes UML :**

UML 2.0 comporte ainsi treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes :

#### **2.3.1. Diagrammes structurels ou diagrammes statiques (UML Structure) :**

- diagramme de classes (Class diagram)
- diagramme d'objets (Object diagram)
- diagramme de composants (Component diagram)
- diagramme de déploiement (Deployment diagram)
- diagramme de paquetages (Package diagram)
- diagramme de structures composites (Composite structure diagram) [6].

### 2.3.2. Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior) :

- diagramme de cas d'utilisation (Use case diagram)
- diagramme d'activités (Activity diagram)
- diagramme d'états-transitions (State machine diagram)
- Diagrammes d'interaction (Interaction diagram)
- diagramme de séquence (Sequence diagram)
- diagramme de communication (Communication diagram)
- diagramme global d'interaction (Interaction overview diagram)
- diagramme de temps (Timing diagram) [6].

## 3. Processus unifié (UP):

### 3.1. Définition :

Le processus unifié est une méthode de prise en charge du cycle de vie d'un logiciel et donc du développement, pour les logiciels orientés objets [10]. C'est une méthode générique, itérative et incrémentale, contrairement à la méthode séquentielle **Merise** [9] (ou **SADT**). PU vient compléter la systémique des modèles **UML**. Elle est le résultat final d'une évolution de l'approche d'Ericsson qui est au fondement d'une des premières méthodes de développement pour applications orientées objets, la méthode Objectory Process (1987). Objectory Process (version 1 à 3.8 en 1995) a elle-même servi de base à la société Rational pour la création de Rational Objectory Process (1997) (version 4.1), parente direct de RUP en 1998.

Les divers avatars d'UP sont : RUP, XUP, AUP, EUP, 2TUP, EssUP

### 3.2. Catégories :

#### 3.2.1. UP est itératif :

L'itération est une répétition d'une séquence d'instructions ou d'une partie de programme un nombre de fois fixé à l'avance ou tant qu'une condition définie n'est pas remplie, dans le but de reprendre un traitement sur des données différentes.

#### 3.2.2. UP est centré sur l'architecture :

Les différentes perspectives, indépendantes et complémentaires, qui permettent de définir

un modèle d'architecture sont proposées par Ph. Kruchten.

#### **4. Les phases :**

##### **a) Analyse des besoins :**

L'analyse des besoins donne une vue du projet sous forme de produit fini. Elle porte essentiellement sur les besoins principaux (du point de vue de l'utilisateur), l'architecture générale du système, les risques majeurs, les délais et les coûts. On met en place le projet. Elle répond aux questions suivantes :

Que va faire le système ? par rapport aux utilisateurs principaux, quels services va-t-il rendre?

Quelle va être l'architecture générale (cible) de ce système ?

Quels vont être : les délais, les coûts, les ressources, les moyens à déployer?

##### **b) Elaboration :**

L'élaboration reprend les éléments de la phase d'analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en oeuvre.

Elle permet de préciser la plupart des cas d'utilisation, de concevoir l'architecture du système et surtout de déterminer l'architecture de référence. Au terme de cette phase, les chefs de projet doivent être en mesure de prévoir les activités et d'estimer les ressources nécessaires à l'achèvement du projet.

Les tâches à effectuer dans la phase élaboration sont les suivantes :

- Créer une architecture de référence
- Identifier les risques
- Définir les niveaux de qualité à atteindre.
- Formuler les cas d'utilisation.
- Élaborer une offre abordant les questions de calendrier, de personnel et de budget

##### **c) Construction :**

La construction est le moment où l'on construit le produit. L'architecture de référence se métamorphose en produit complet. Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version.

##### **d) Transition :**

Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte

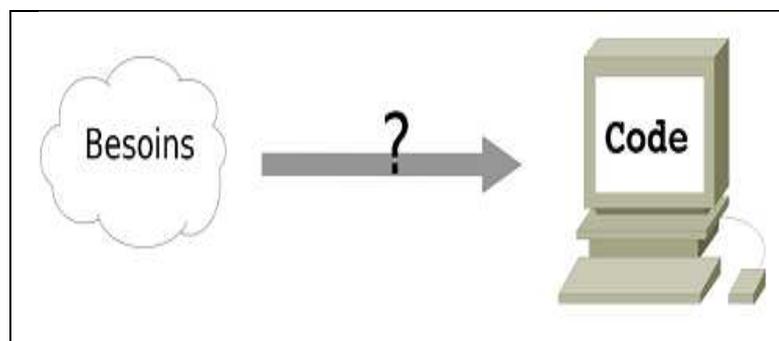
Les anomalies et défauts.

Cette phase suppose des activités (la formation des utilisateurs clients, la mise en œuvre

d'un service d'assistance, la correction des anomalies constatées).

## 5. L'UP simplifier :

UML n'est qu'un langage de modélisation, ce n'est pas une méthode. En effet, UML ne propose pas une démarche de modélisation explicitant et encadrant toutes les étapes d'un projet. nous allons présenter la méthode simple et générique proposé par Laurent AUDIBERT qui se situe à mi-chemin entre UP (Unified Process), qui constitue un cadre général très complet de processus de développement, et XP (EXtreme Programming) qui est une approche minimaliste à la mode centrée sur le code[6].

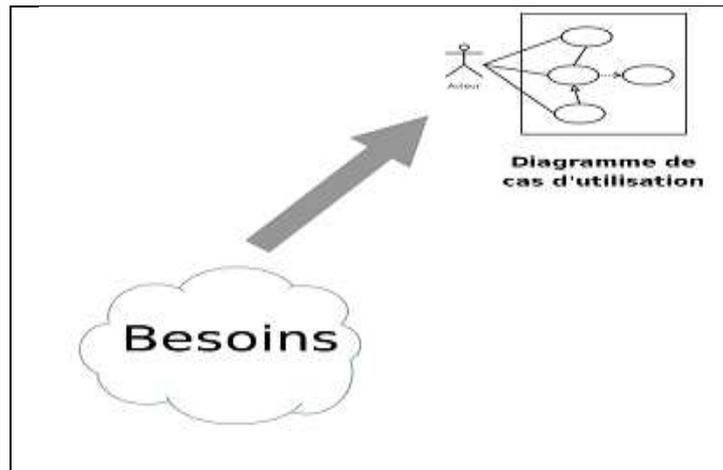


**Figure 1.1 :** Quelle méthode pour passer de l'expression des besoins au code de l'application?

## 6. Identification des besoins et spécification des fonctionnalités :

### 6.1. Identification et représentation des besoins : diagramme de cas d'utilisation

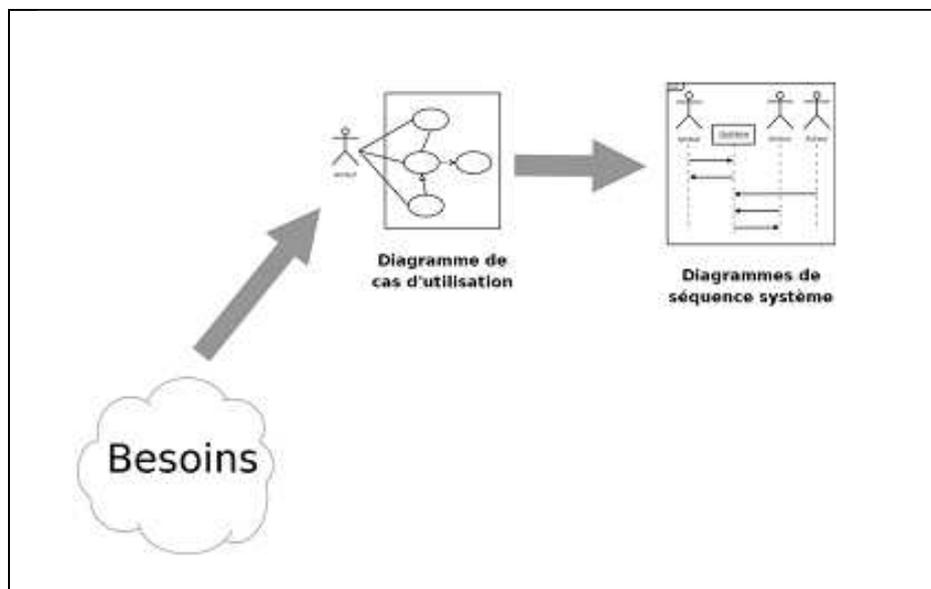
Les cas d'utilisation sont utilisés tout au long du projet. Dans un premier temps, on les crée pour identifier et modéliser les besoins des utilisateurs, ces besoins sont déterminés à partir des informations recueillies lors des rencontres entre informaticiens et utilisateurs. Durant cette étape, vous devrez déterminer les limites du système, identifier les acteurs et recenser les cas d'utilisation [6].



**Figure 1.2 :** Les besoins sont modélisés par un diagramme de cas d'utilisation

## 6.2. Spécification détaillée des besoins : diagrammes de séquence système :

Dans cette étape, on cherche à détailler la description des besoins par la description textuelle des cas d'utilisation, et la production de diagrammes de séquence système illustrant cette description textuelle. Cette étape amène souvent à mettre à jour le diagramme de cas d'utilisation puisque nous sommes toujours dans la spécification des besoins[6].



**Figure 1.3 :** Les diagrammes de séquence système illustrent la description textuelle des cas d'utilisation

### 6.3. Maquette de l'IHM de l'application (non couvert par UML) :

Une maquette d'IHM (Interface Homme-Machine) est un produit jetable permettant aux utilisateurs d'avoir une vue concrète mais non définitive de la future interface de l'application. La maquette peut très bien consister en un ensemble de dessins produits par un logiciel de présentation ou de dessin. Par la suite, la maquette pourra intégrer des fonctionnalités de navigation permettant à l'utilisateur de tester l'enchaînement des écrans ou des menus, même les fonctionnalités restent fictives. La maquette doit être développée rapidement afin de provoquer des retours de la part des utilisateurs [6].

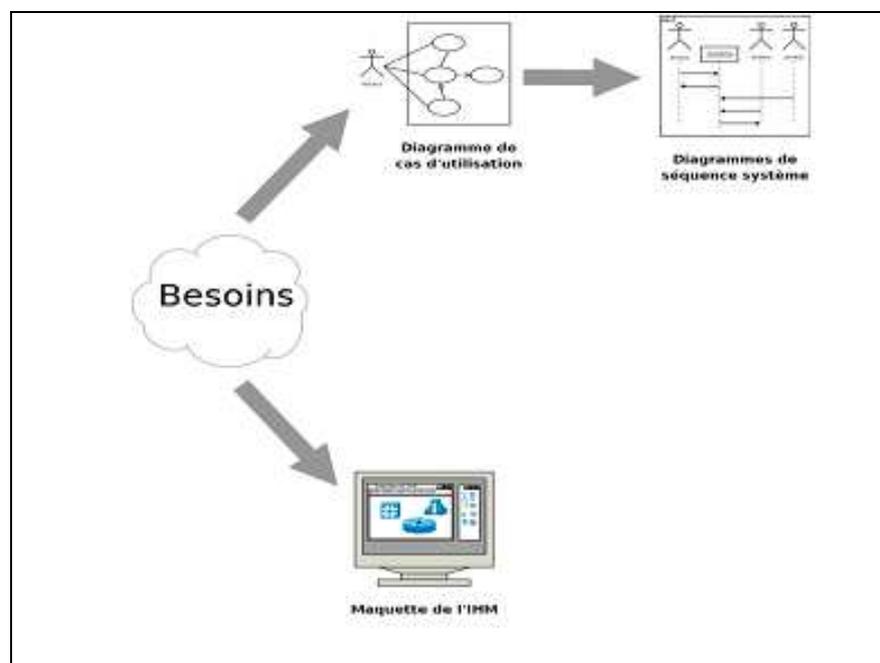


Figure 1.4 : Une maquette d'IHM facilite les discussions avec les futurs utilisateurs

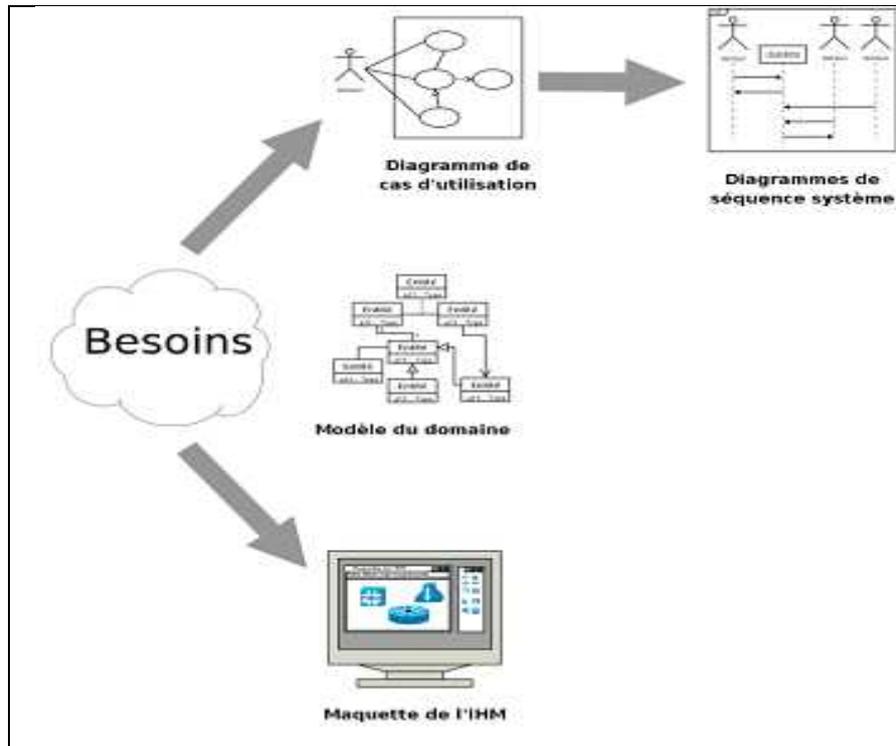
## 7. Phases d'analyse :

### 7.1. Analyse du domaine : modèle du domaine :

L'élaboration du modèle des classes du domaine permet d'opérer une transition vers une véritable modélisation objet. L'analyse du domaine est une étape totalement dissociée de l'analyse des besoins. Elle peut être menée avant, en parallèle ou après cette dernière.

La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes appelée modèle du domaine. Ce modèle doit définir les

classes qui modélisent les entités ou concepts présents dans le domaine (on utilise aussi le terme de métier) de l'application [6].



**Figure1.5 :** La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes.

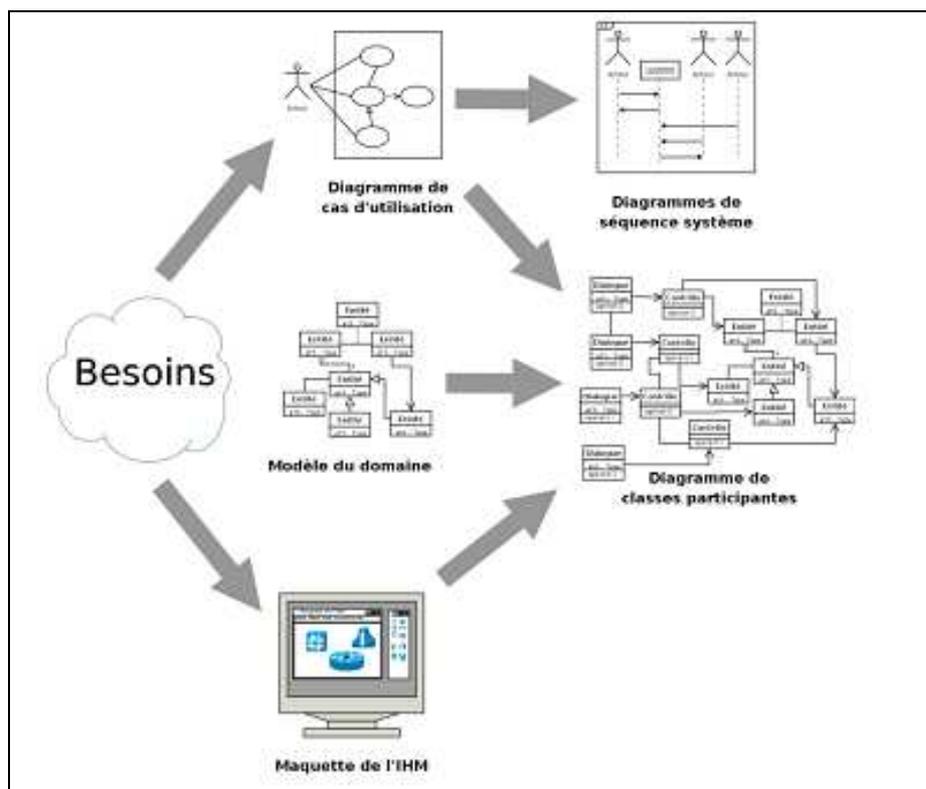
## 7.2. Diagramme de classes participantes :

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de conception logicielle que sont les diagrammes d'interaction et le diagramme de classes de conception[6].

Le diagramme de classes participantes modélise trois types de classes d'analyse, les dialogues, les contrôles et les entités ainsi que leurs relations :

- Les classes de dialogues : Les classes qui permettent les interactions entre l'IHM et les utilisateurs sont qualifiées de dialogues.
- Les classes de contrôles: Les classes qui modélisent la cinématique de l'application sont appelées contrôles. Elles font la jonction entre les dialogues et les classes métier en permettant aux différentes vues de l'application de manipuler des informations détenues par un ou plusieurs objets métier. Elles contiennent les règles applicatives et les isolent à la fois des dialogues et des entités.

- Les classes entités : Les classes métier, qui provient directement du modèle du domaine, sont qualifiées d'entités. Ces classes sont généralement persistantes, c'est-à-dire qu'elles survivent à l'exécution d'un cas d'utilisation particulier et qu'elles permettent à des données et des relations d'être stockées dans des fichiers ou des bases de données. Lors de l'implémentation, ces classes peuvent ne pas se concrétiser par des classes mais par des relations, au sens des bases de données relationnelles [6].

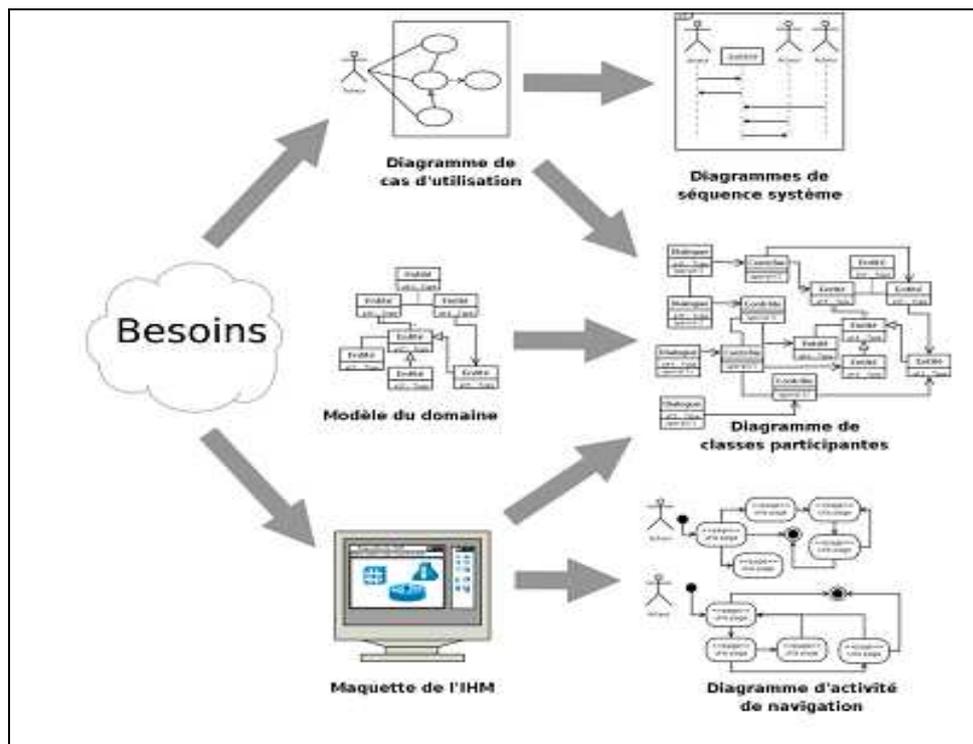


**Figure 1.6:** Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, Le modèle du domaine et les diagrammes de conception logicielle.

### 7.3. Diagrammes d'activités de navigation :

Les IHM modernes facilitent la communication entre l'application et l'utilisateur en offrant une gamme de moyens d'action et de visualisation comme des menus déroulants ou contextuels, des palettes d'outils, des boîtes de dialogues, des fenêtres de visualisation, etc. Cette combinaison possible d'options d'affichage, d'interaction et de navigation aboutis aujourd'hui à des interfaces de plus en plus riches et puissantes.

UML offre la possibilité de représenter graphiquement cette activité de navigation dans l'interface en produisant des diagrammes dynamiques. On appelle ces diagrammes des diagrammes de navigation. Les diagrammes d'activités de navigation sont à relier aux classes de dialogue du diagramme de classes participantes. Les différentes activités du diagramme de navigation peuvent être stéréotypées en fonction de leur nature : «fenêtre», «menu», «menu contextuel», «dialogue», etc [6].



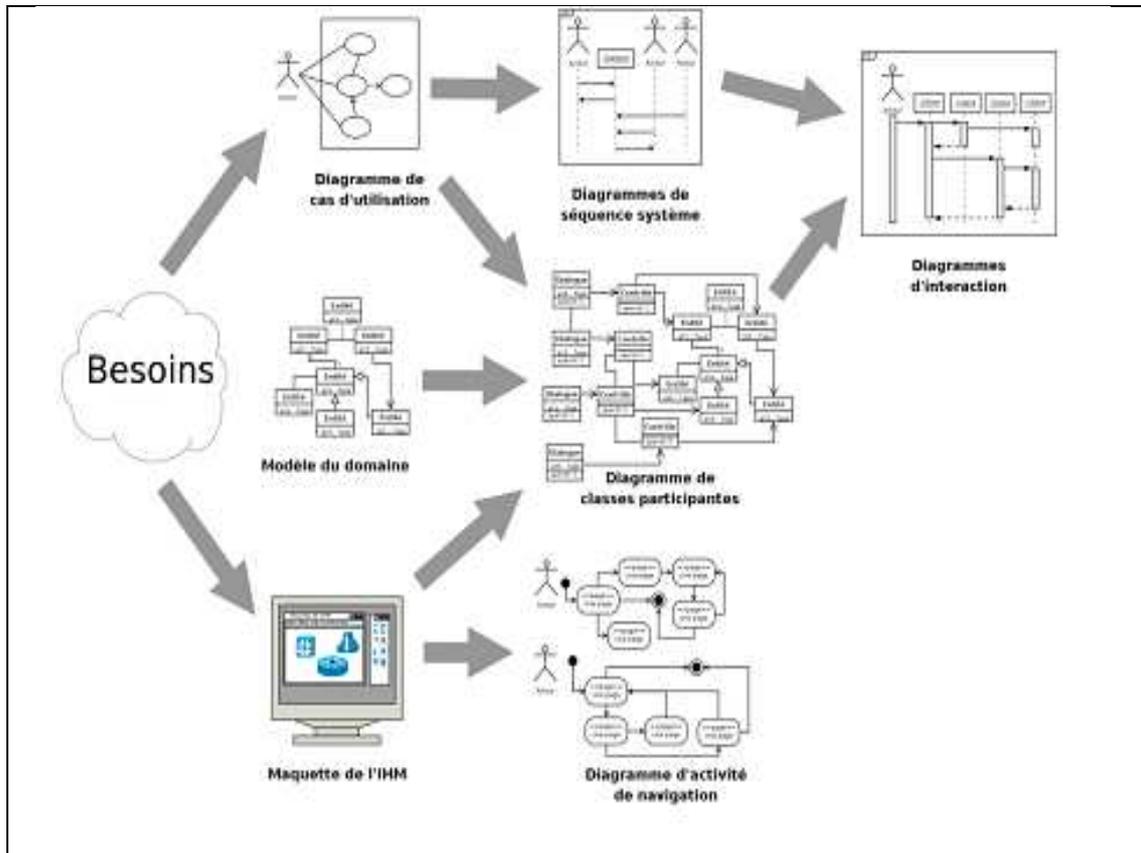
**Figure 1.7 :** Les diagrammes d'activités de navigation représentent graphiquement l'activité de Navigation dans l'IHM.

## 8. Phase de conception :

### 8.1. Diagrammes d'interaction :

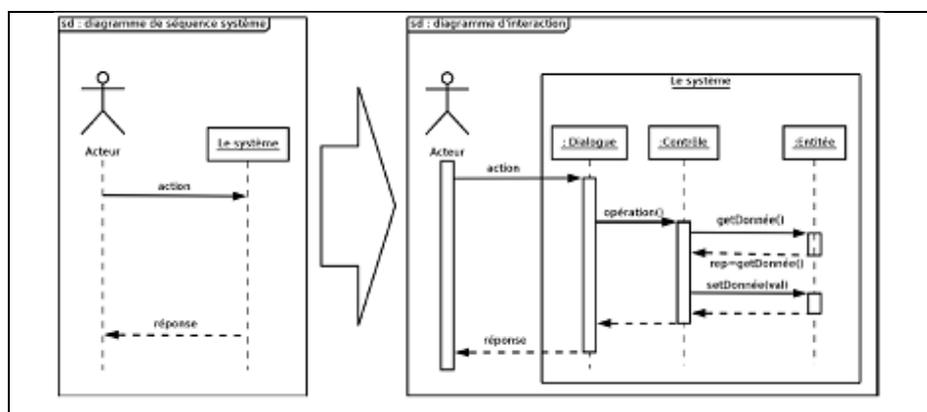
Maintenant, il faut attribuer précisément les responsabilités de comportement, dégagée par le diagramme de séquence système, aux classes d'analyse du diagramme de classes participantes. Les résultats de cette réflexion sont présentés sous la forme de diagrammes d'interaction UML. Parallèlement, une première ébauche de la vue statique de conception, c'est-à-dire du diagramme de classes de conception, est construite et complétée. Durant cette phase, l'ébauche du diagramme de classes

de conception reste indépendante des choix technologiques qui seront faits ultérieurement[6].



**Figure 1.8 :** Les diagrammes d’interaction permettent d’attribuer précisément les responsabilités de comportement aux classes d’analyse.

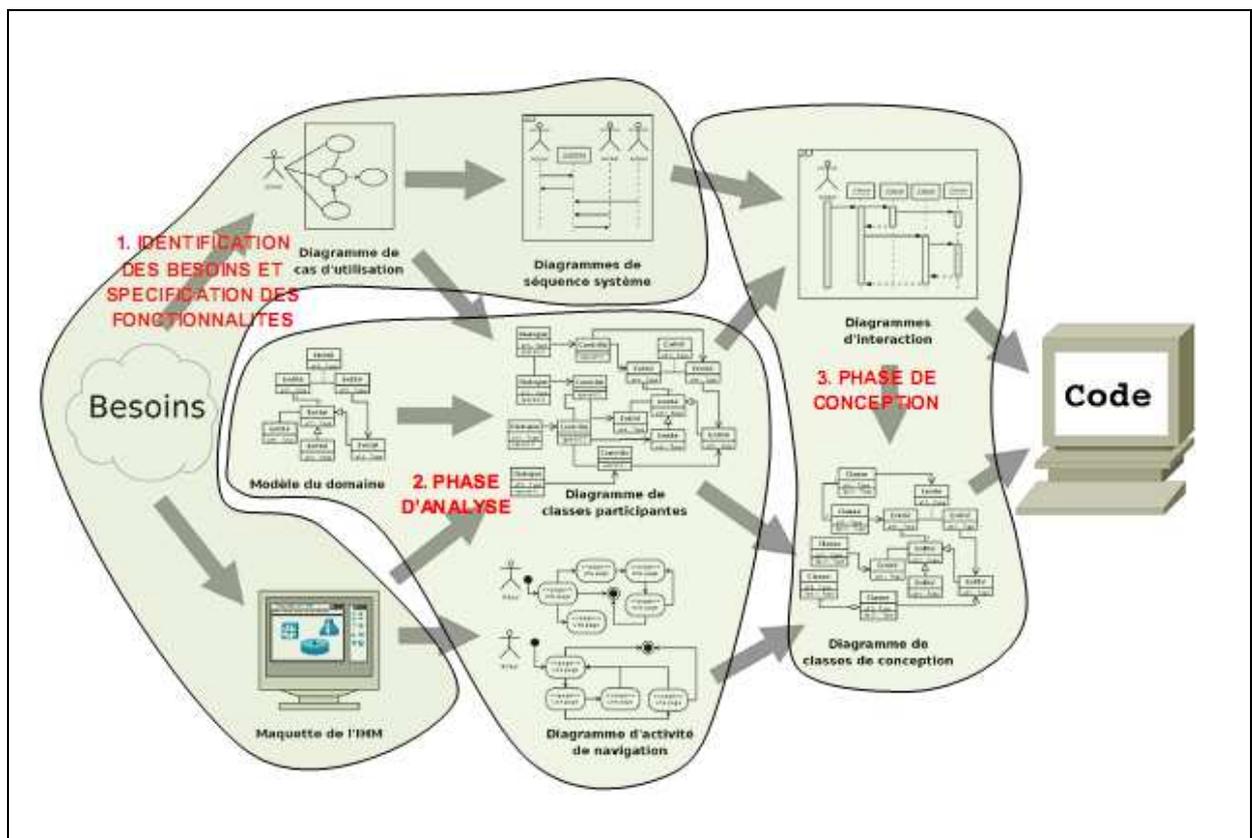
Par rapport aux diagrammes de séquences système, nous remplaçons ici le système, vu comme une boîte noire, par un ensemble d’objets en collaboration. Ces objets sont des instances des trois types de classes d’analyse du diagramme de classes participantes, à savoir des dialogues, des contrôles et des entités [6].



**Figure 1.9 :** Le système des diagrammes de séquences système, vu comme une boîte noire, est remplacé par un ensemble d’objets en collaboration.

## 8.2. Diagramme de classes de conception :

L'objectif de cette étape est de produire le diagramme de classes qui servira pour l'implémentation. Une première ébauche du diagramme de classes de conception a déjà été élaborée en parallèle du diagramme d'interaction. Il faut maintenant le compléter en précisant les opérations privées des différentes classes [6].



**Figure 1.10:** Chaîne complète de la démarche de modélisation du besoin jusqu'au code.

**Conclusion :**

Dans ce chapitre, nous avons passé en revue le langage de modélisation UML. UML est une notation graphique conçue pour représenter, spécifier, construire et documenter les systèmes logiciels .Ses deux principaux objectifs sont la modélisation de systèmes utilisant les techniques orientées objet, depuis la conception jusqu'à la maintenance, est la création d'un langage abstrait compréhensible par l'homme et interprétable par les machines.

UML s'adresse à toutes les personnes chargées de la production, du déploiement et du suivi de logiciels( analystes, développeurs, chefs de projets, architectes ... ), mais peut également servir à la communication avec les clients et les utilisateurs du logiciel. Il s'adapte à tous les domaines d'application et a tous les supports. Il permet de construire plusieurs modèles d'un système, chacun mettant en valeur des aspects différents : fonctionnels, statiques, dynamique Et organisationnels. UML est devenu un langage incontournable dans les projets de développement.

# Chapitre

## Identification des besoins

*-Introduction.*

*1- Cahier de charge.*

*2- Dictionnaire de données.*

*3 -Diagramme de cas  
d'utilisation.*

*4 -La description des cas  
d'utilisation.*

*5 -Diagramme de séquence*

*-Conclusion.*

# 2

## Introduction

Dans ce chapitre nous allons identifier les besoins du système à réaliser. Au début nous allons définir le cahier de charges qui contient toutes les informations de ce projet. Puis, nous commencerons à identifier les acteurs qui interagiront avec le système, nous développerons un premier modèle UML de niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du système. Ensuite, nous identifions et nous décrivons les cas d'utilisation du système.

## 1. Cahier de charge

### 1.1. Présentation du projet :

La gestion de scolarité est une tâche critique et pointue parmi les fonctions de l'institut. Cette tâche est devenue un peu plus compliquée qu'avant avec le système LMD, car un étudiant peut étudier à cheval de plusieurs années (date). Les modules sont couronnées de plusieurs paramètres tel que le coefficient, le crédit l'unité etc. la délibération passe aussi par plusieurs étapes: PV de matière, PV d'unité, PV de semestre etc. et ils existent plusieurs types de PV: PV ordinaire, PV étudiants avec dettes etc. Donc, on a besoin d'une application informatique qui permet, une fois les données concernant les étudiants et les modules sont à saisir, de produire les différents PVs. Nous signalons ici l'existence déjà d'une application informatique qui est responsable de la tâche de délibération automatique. Cette application souffre d'un certain nombre d'anomalies et rend parfois la gestion scolaire automatique plus difficile à vérifier qu'une délibération qu'on peut faire manuellement.

### 1.2. Les besoins fonctionnels :

- établissement de la scolarité
- initialise toutes les informations de la scolarité
- calcul des moyennes
- Manipulation de la scolarité
- Recherche (étudiant, enseignant, note, moyenne ...)
- Impression (listes, relevés de notes...)

### 1.3. Les besoins techniques :

- **Convivialité et simplicité d'utilisation de l'application** : l'application doit avoir une interface graphique claire et bien documentée.
- **Le temps de réponse** : le temps de réponse doit être acceptable.

- **La sécurité** : l'utilisation du système est limitée à l'administrateur.

Le système doit permettre de :

- **S'authentifier** : l'administrateur se connecte au système par un mot de passe.

## 2. Dictionnaire de données :

Le dictionnaire de données permet de définir tous les termes et les concepts du projet.

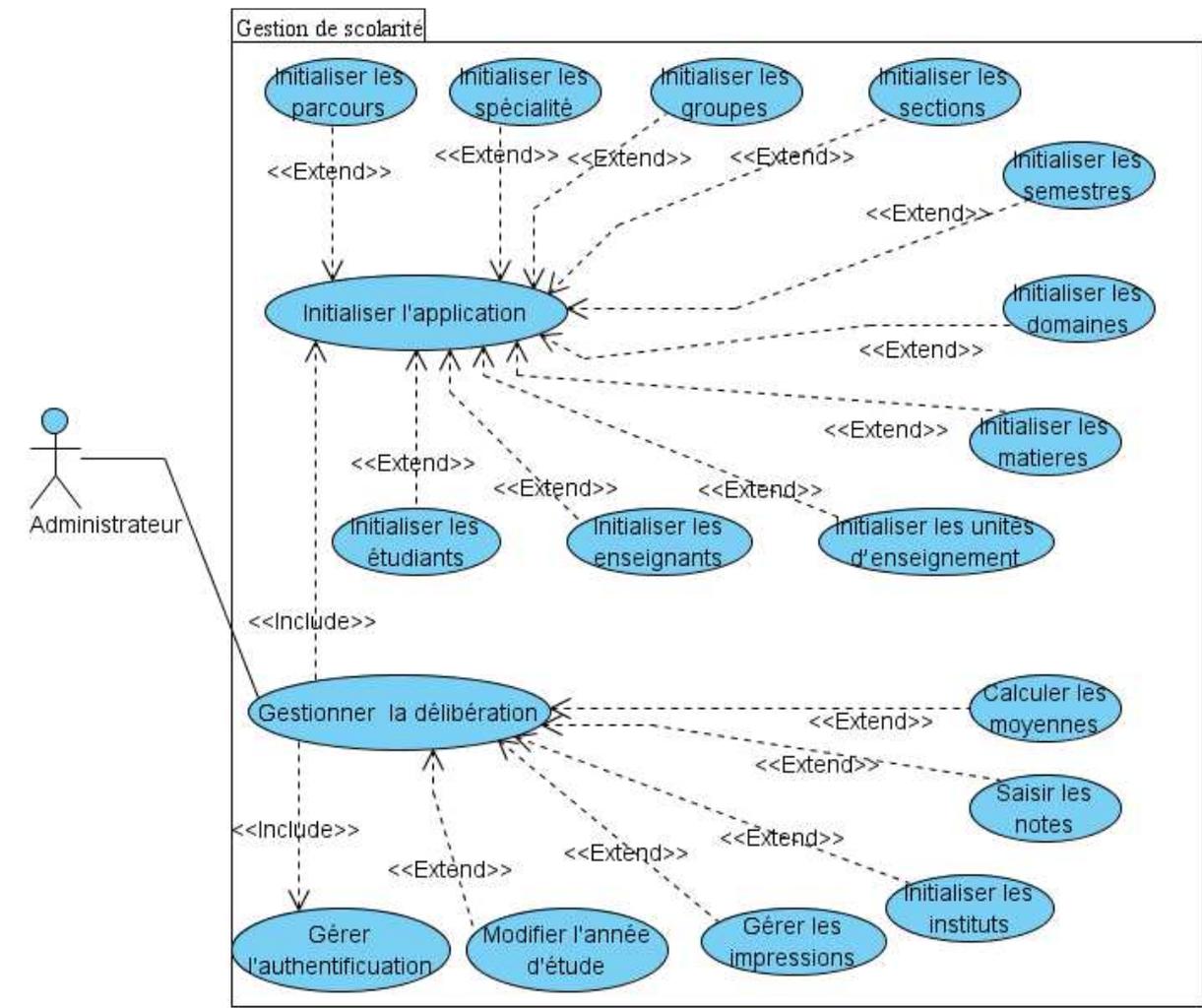
<i>Notion</i>	<i>Définition</i>
Institut	Définit le nom de l'institut et peut être composé d'un ou de plusieurs départements.
Système	Comprend l'ensemble des données d'un institut. comme les formations, administrateur, notes, coeffs....
Domaine	Définit les domaines de l'institut.
Parcours	Définit les parcours d'un domaine.
Spécialité	Définit les spécialités d'un parcours.
Section	Contient plusieurs groupes.
Groupe	Groupe d'étudiants.
Niveau	Détermine les niveaux d'étude.
Etudiant	Contient le numéro d'inscription, le nom, le prénom, le groupe, l'année de naissance, le domaine...
Enseignant	Contient le nom, le prénom, la ou les matière(s) d'enseignement(s).
Administrateur	Opère sur une base de données. il peut supprimer, modifier, rechercher des ressources.
Matière	Définit le crédit matière et le coefficient de la matière. A chaque matière correspond un Examen Final et/ou un Contrôles Continu.
<i>Coeff- matière</i>	Définit le coefficient d'une matière.
<i>crédit matière</i>	Définit le crédit d'une matière.
Les unités d'enseignement (UE)	Les unités d'enseignement sont de quatre types : Fondamental, Découverte, Spécialité ou Transversal. Une UE est dotée d'un

	Crédit et d'un Coefficient .L' UE Contient un ou plusieurs matières.
Coeff- unité	Définit le coefficient d'une unité.
crédit unité	Définit le crédit d'une unité.
Semestre	Contient un ou plusieurs unités d'enseignement.
Moyenne d'un semestre	Représente la moyenne d'un semestre.
Note	Représente la note d'un étudiant pour une matière.
Relevé de not	Contient les notes d'un étudiant plus sa moyenne.
PV d'unité	les moyennes et les crédits des étudiants pour une unité d'enseignement.
PV de matière	les moyennes et les crédits des étudiants pour une matière.
PV de semestre	les moyennes et les crédits des étudiants pour un semestre.
PV annelle	les moyennes et les crédits des étudiants pour une année.
Cas de discipline	Contient la sanction infligée à un étudiant traduit devant le conseil de discipline et sanctionné.

Action	Définition
Connexion	Permet à un utilisateur de se connecter au système.
Initialisation	Permet d'initialiser les informations.
Ajouter matière	Permet d'ajouter une matière.
Délibération	Permet de produire les différents PVs.
Supprimer matière	Permet de supprimer une matière.
Modifier matière	Permet de modifier le contenu d'une matière.
Ajouter note	Permet d'ajouter une note.
Supprimer note	Permet de supprimer une note.
Modifier note	Permet de modifier une note.
Ajouter unité	Permet d'ajouter une unité.

Supprimer unité	Permet de supprimer une unité.
Modifier unité	Permet de modifier une unité.
Imprimer le PV d'unité	Permet d'imprimer le PV d'unité.
Imprimer le PV de semestre	Permet d'imprimer le PV semestre.
Imprimer le PV annelle	Permet d'imprimer le PV annelle.
Imprimer le PV de matière	Permet d'imprimer le PV matière.
Saisir de notes	Permet la saisie des notes des étudiants.
Imprimer le relevé de note	Permet d'Imprimer le relevé de note.
Modifier étudiant	Permet de modifier un étudiant.
Ajouter étudiant	Permet d'ajouter un étudiant.
Supprimer étudiant	Permet de supprimer un étudiant.
Modifier enseignant	Permet de modifier un enseignant.
Ajouter enseignant	Permet d'ajouter un enseignant.
Supprimer enseignant	Permet de supprimer un enseignant.
Modifier parcours	Permet de modifier un parcours.
Ajouter parcours	Permet d'ajouter un parcours.
Supprimer parcours	Permet de supprimer un parcours.
Rechercher étudiant	Permet de rechercher un étudiant.

**3. Diagramme de cas d'utilisation:**



**Figure 2.1 :** Diagramme de cas d'utilisation.

**4. Les fiches descriptives des cas d'utilisations :**

**4.1. Gérer l'authentification :**

Cas d'utilisation	Gérer l'authentification.	
Acteur	L'administrateur.	
But	Vérifier l'autorisation d'accéder au système.	
Pré Conditions	L'utilisateur possède les informations d'accès.	
Post Conditions	L'utilisateur est authentifié par le système.	
Scénario	Etape	Action

Nominal	1	L'utilisateur choisit d'accéder au système.
	2	Le système demande à l'utilisateur d'entrer user name et un mot de passe.
	3	L'utilisateur saisit user name et le mot de passe et valide.
	4	Le système vérifie la validité du user name et du mot de passe et ouvre le système.
Scénario Alternatif	Etape	Action
	4	Le mot de passe est erroné: 4.1. Le système affiche un message d'erreur. 4.2. Le système propose à l'utilisateur de rentrer une nouvelle fois le user name et le mot de passe. 4.3. Reprise de l'enchaînement du scénario nominal au point 2.
Scenario Erreurs	User name ou mot de passe est erroné pour la 3 <sup>ème</sup> fois il sera déconnecté du système.	

#### 4.2. Gestionner la délibération:

Cas d'utilisation	Gestionner la délibération.	
Acteur	L'administrateur.	
But	Il permette à l'administrateur de gère la scolarité.	
Pré Conditions	L'administrateur authentifie.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	L'administrateur accède au système.
	2	Le système présente l'état actuel.
	3	3.1. Ajouter ou modifier l'année d'étude sous cas : Modifier l'année d'étude.  3.2. Initialiser : étudiant, enseignant, domaine, parcours, spécialité, semestre, unité d'enseignement, matière, section et groupe. sous cas : Initialiser l'application.  3.3. Ajouter, supprimer ou modifier les notes sous

		cas : Saisir les notes.
		3.4. Calculer des moyennes et des crédits sous cas : Calculer les moyennes.
		3.5. Ajouter ou modifier les notes sous cas : Initialiser les instituts.
		3.6. Imprimer les relèves de notes et les PVs sous cas : Gestionner les impressions.
Scénario Alternatif	Etape 3	Action L'administrateur annule l'accès à un sous cas et retour à 2 « l'état actuel ».
Scenario Erreurs	\	

### 4.3. Modifier l'année d'étude:

Cas d'utilisation	Modifier l'année d'étude	
Acteur	L'administrateur.	
But	Ajouter ou modifier une année d'étude.	
Pré Conditions	L'administrateur possède les informations d'une année d'étude.	
Post Conditions	Les informations ont été enregistrées.	
Scénario nominal	Etape	Action
	1	Le système présente l'état actuel de l'année d'étude.
	2	L'administrateur choisit une opération d'année d'étude ajouter ou modifier : <b>2.1. Cas d'ajout:</b> a- Le système demande de saisir l'année d'étude. b- L'administrateur saisi l'année d'étude et valide l'ajout. <b>2.2. Cas de modification:</b> a- Le système demande de saisir l'année d'étude. b- L'administrateur saisi l'année d'étude. c- Le système présente les informations actuelles de l'année d'étude. d- Système demande de saisir la nouvelle année d'étude. e- L'administrateur saisi les informations et valide la modification.

	3	Le système enregistre les modifications.
Scénario Alternatif	Etape	Action
	2	1. b. l'année d'étude existe déjà : affiche un message d'existence et retour à 2.1. 2. c. l'année d'étude n'existe pas : affiche un message d'absence et retour à 2.2.
Scenario Erreurs	Le cas où L'administrateur annule l'opération retour à 2 « l'état actuel » du cas : Gestionner la délibération.	

**4.4.Saisir les notes:**

Cas d'utilisation	Saisir les notes.	
Acteur	L'administrateur.	
But	Ajouter ou modifier une note.	
Pré Conditions	L'administrateur possède les notes.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	L'administrateur accède au système.
	2	Le système présente l'état actuel des notes.

	3	<p>L'administrateur choisie une opération : ajout ou modifié les notes.</p> <p><b>1. Cas d'ajout:</b></p> <p>a- Le système demande de sélectionner l'institut, le domaine, le niveau, le semestre, le code de l'unité d'enseignement, le parcours, la spécialité, le code de matière, l'année d'étude, la section, le numéro de groupe et le type de note.</p> <p>b- L'administrateur sélectionne l'institut, le domaine, le niveau, le semestre, le code de l'unité d'enseignement, le parcours, la spécialité, l'année d'étude, la section, le code de matière, le numéro de groupe et le type de note.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système présente la liste des étudiants et demande de saisir les notes des étudiants.</p> <p>f- L'administrateur saisie les notes et valide la saisie.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le numéro de groupe.</p> <p>b- L'administrateur saisie le numéro de groupe.</p> <p>c- Le système demande de sélectionner le type de note.</p> <p>d- L'administrateur sélectionne le type de note.</p> <p>e- Le système présente la liste des étudiants du groupe avec les notes.</p> <p>f- Le système demande de sélectionner le numéro d'étudiant.</p> <p>g- L'administrateur sélectionne le numéro d'étudiant.</p> <p>h- Le système demande de saisir la nouvelle note.</p> <p>i- L'administrateur saisie la nouvelle note et valide la modification.</p>
	4	Le système enregistre les modifications.

Scénario Alternatif	2. b. le numéro de groupe n'existe pas affiche un message d'absence et retour à 3.2.
Scénario D'erreurs	Le cas où l'administrateur annule l'opération retour à 2 « l'état actuel » du cas : Gérer la délibération.

#### 4.5.Initialiser les instituts :

Cas d'utilisation	Initialiser les instituts	
Acteur	L'administrateur.	
But	Ajouter ou modifier un Institut.	
Pré Conditions	L'administrateur possède les informations d institut.	
Post Conditions	Les informations ont été enregistrées.	
Scénario nominal	Etape	Action
	1	Le système présente l'état actuel de d institut.
	2	<p>L'administrateur choisit une opération d'institut « ajout ou modification ».</p> <p><b>1. Cas d'ajout:</b></p> <p>a- Le système demande de saisir le nom d'institut et les informations.</p> <p>b- L'administrateur saisie le nom et les informations de l'institut et valide l'ajout.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le nom d'un institut.</p> <p>b- L'administrateur saisie le nom d'institut.</p> <p>c- Le système présente les informations actuelles de l'institut.</p> <p>d- Système demande de saisir les nouvelles informations.</p> <p>e- L'administrateur saisie les informations et valide la modification.</p>
	3	Le système enregistre les modifications.
Scénario Alternatif	Etape	Action
	2	1. b. le nom d'institut existe déjà : affiche un message d'existence et retour à 2.1.

		2. c. le nom d'institut n'existe pas : affiche un message d'absence et retour à 2.2.
Scenari Erreurs	Le cas où L'administrateur annule l'opération retour à 2 le cas : Gestionner l'initialisation.	

#### 4.6.Calculer les moyennes :

Cas d'utilisation	Calculer les moyennes.	
Acteur	L'administrateur.	
But	Calcule les moyennes des unités, matières, semestres ou annuelle.	
Pré Conditions	L'administrateur possède les informations.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	L'administrateur accède au système.
	2	Le système présente l'état actuel.
	3	<p>a- Le système demande de sélectionner l'institut et le domaine, le niveau, le parcours, la spécialité</p> <p>b- L'administrateur sélectionne le nom de l'institut et le domaine, le niveau, le parcours, la spécialité.</p> <p>c- Le système demande de valider la sélection.</p> <p>e- L'administrateur valide la sélection.</p> <p>f- L'administrateur choisie une opération de calcule : PV matière, PV unité, PV semestre ou PV annule.</p> <p><b>1. Cas PV matière:</b></p> <p>a- Le système demande de sélectionner le semestre et l'unité d'enseignement.</p> <p>b- L'administrateur sélectionne le semestre et l'unité d'enseignement.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système demande de saisir le code d'une matière.</p> <p>f- L'administrateur saisi le code d'une matière.</p> <p>g- Le système présente les notes de contrôle continu, l'examen et le coefficient de contrôle continu et d'examen.</p>

		<p>h- L'administrateur demande de calculer les moyennes et les crédits de cette matière.</p> <p>i- Le système calcule les moyennes et les crédits de cette matière.</p> <p><b>2. Cas PV unité:</b></p> <p>a- Le système demande de sélectionner le semestre.</p> <p>b- L'administrateur sélectionne le semestre.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système demande de saisir le code d'unité d'enseignement.</p> <p>f- L'administrateur saisi le code d'unité d'enseignement.</p> <p>g- Le système présente les moyennes des matières de cette unité avec leurs coefficients.</p> <p>h- L'administrateur demande de calculer les moyennes et les crédits.</p> <p>i- Le système calcule les moyennes et les crédits.</p> <p><b>3. Cas PV semestre:</b></p> <p>a- Le système demande de saisir le numéro de semestre.</p> <p>b- L'administrateur saisie le numéro de semestre.</p> <p>c- Le système présente les moyennes des unités de ce semestre.</p> <p>d- L'administrateur demande de calculer les moyennes et les crédits.</p> <p>e- Le système calcule les moyennes et les crédits.</p> <p><b>4. Cas PV annuelle:</b></p> <p>a- Le système présente les moyennes des semestres de ce niveau.</p> <p>b- L'administrateur demande de calculer les moyennes et les crédits.</p> <p>c- Le système calcule les moyennes et les crédits.</p>
	4	Le système enregistre les modifications.

Scénario Alternatif	Etape 3	Action 1. f. le code de matière n'existe pas affiche un message d'absence et retour à 3.1. 2. f. le code d'unité n'existe pas affiche un message d'absence et retour à 3.2. 3. b. le code de semestre n'existe pas affiche un message d'absence et retour à 3.3.
Scenario D'erreurs	Le cas où l'administrateur annule l'opération retour à 2 « l'état actuel » du cas : Gestionner la délibération.	

#### 4.7. Gestionner les impressions:

Cas d'utilisation	Gâtionner les impressions	
Acteur	L'administrateur.	
But	Imprime les relevés de note et les PVs.	
Pré Conditions	L'administrateur possède les informations.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	<b>Etape</b>	<b>Action</b>
	1	Le système présente l'état actuel.
	2	a- Le système demande de sélectionner l'institut, le domaine, le niveau, le parcours, la spécialité. b- L'administrateur sélectionne le nom de l'institut et le domaine et le niveau, le parcours, la spécialité. c- Le système demande de valider la sélection. d- L'administrateur valide la sélection. e- L'administrateur choisie une opération d'un semestre imprimer les relevés de note ou les PVs:  <b>1. Cas d'impression des relevés de note:</b> a- Le système demande de saisir le numéro de section. b- L'administrateur saisie le numéro de section.

	<p>c- Le système présente les relevés de note de la section.</p> <p>d- L'administrateur demande d'imprimer les relevés de notes.</p> <p>e- Le système imprime les relevés de note.</p> <p><b>2. Cas d'impression des PVs de matière:</b></p> <p>a- Le système demande de sélectionner le semestre et l'unité d'enseignement.</p> <p>b- L'administrateur sélectionne le semestre, l'unité d'enseignement.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système demande de saisir le code de matière.</p> <p>f- L'administrateur saisi le code de matière.</p> <p>g- Le système présente les PVs de matière.</p> <p>h- L'administrateur demande d'imprimer les PVs matière.</p> <p>i- Le système imprime les PVs matière.</p> <p><b>3. Cas d'impression des PVs d'unité d'enseignement:</b></p> <p>a- Le système demande de sélectionner le semestre.</p> <p>b- L'administrateur sélectionne le semestre.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système demande de saisir le code d'unité d'enseignement.</p> <p>f- L'administrateur saisi le code d'unité d'enseignement.</p> <p>g- Le système présente les PVs d'unité d'enseignement.</p> <p>h- L'administrateur demande d'imprimer les PVs d'unité.</p> <p>i- Le système imprime les PV s d'unité.</p> <p><b>4. Cas d'impression des PVs semestre</b></p> <p>a- Le système demande de saisir le numéro de semestre.</p> <p>b- L'administrateur saisie le numéro le semestre.</p> <p>c- Le système présente les PVs de semestre.</p> <p>d- L'administrateur demande d'imprimer les PVs de semestre.</p>
--	---

		e- Le système imprime les PVs de semestre. <b>5. Cas d'impression des PVs annuelles</b> a- Le système présente les PVs annuelles. b- L'administrateur demande d'imprimer les PVs annuelles. c- Le système imprime les PVs annuelles.
	<b>3</b>	Le système enregistre les modifications.
Scénario Alternatif	/	
Scenario D'erreurs	Le cas où l'administrateur annule l'opération retour en 2 « l'état actuel » du cas : Gérer la délibération.	

#### 4.8. Gérer l'initialisation :

Cas d'utilisation	Initialiser l'application.	
Acteur	L'administrateur.	
But	Il permet à L'administrateur d'initialiser les informations concernant la scolarité.	
Pré Conditions	L'administrateur possède les informations.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	L'administrateur accède au système.
	2	Le système présente l'état actuel.
	3	a- Le système demande de sélectionner l'institut. b- L'administrateur sélectionne l'institut. c- Le système demande de valider la sélection. d- L'administrateur valide la sélection. e- Le système demande de sélectionner une Opération: <ul style="list-style-type: none"> <li>➤ Ajouter, supprimer ou modifier les étudiants sous cas : Initialiser les étudiants.</li> <li>➤ Ajouter, supprimer ou modifier les enseignants sous cas : Initialiser les enseignants.</li> <li>➤ Ajouter ou modifier domaine sous cas : Initialiser</li> </ul>

		<p>les domaines.</p> <ul style="list-style-type: none"> <li>➤ Ajouter ou modifier parcours sous cas : Initialiser les parcours.</li> <li>➤ Ajouter ou modifier spécialité sous cas : Initialiser les spécialités.</li> <li>➤ Ajouter ou modifier semestre sous cas : Initialiser les semestres.</li> <li>➤ Ajouter ou modifier unité d'enseignement sous cas : Initialiser les unités d'enseignement.</li> <li>➤ Ajouter ou modifier matière sous cas : Initialiser les matières.</li> <li>➤ Ajouter, supprimer, modifier section sous cas: Initialiser les sections.</li> <li>➤ Ajouter, supprimer ou modifier groupe sous cas: Initialiser les groupes.</li> </ul>
Scénario Alternatif	Etape	Action
	3	L'administrateur annule l'accès à un sous cas et retour à 2.
Scenario Erreurs	Le cas où L'administrateur annule l'opération retour à 2 « l'état actuel » du cas : Gestionner la délibération.	

#### 4.9. Initialiser les étudiants :

Cas d'utilisation	Initialiser les étudiants.	
Acteur	L'administrateur.	
But	Ajouter, supprimer ou modifier un étudiant.	
Pré Conditions	L'administrateur possède les informations d'un étudiant.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	Le système présente l'état actuel d'étudiant.
	2	L'administrateur choisit une opération de Initialiser les

	<p>étudiants ajouter, modifier, supprimer:</p> <p><b>1. Cas d'ajout:</b></p> <p>a- Le système demande de sélectionner le niveau, le domaine, le parcours, la spécialité, la section et le groupe.</p> <p>b- L'administrateur sélectionne le niveau, le domaine, le parcours, la spécialité, la section et le groupe.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système demande de saisir le numéro et les informations d'un étudiant.</p> <p>f- L'administrateur saisie le numéro et les informations d'un étudiant et valide l'ajout.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le numéro d'étudiant.</p> <p>b- L'administrateur saisie le numéro d'étudiant.</p> <p>c- Le système présente les informations actuelles d'étudiant.</p> <p>d- Le système demande de saisir les nouvelles informations d'étudiant.</p> <p>e- L'administrateur saisie les informations d'étudiant et valide la modification.</p> <p><b>3. Cas de suppression:</b></p> <p>a- Le système demande de saisir le numéro d'étudiant.</p> <p>b- L'administrateur saisie le numéro d'étudiant.</p> <p>c- Le système présente les informations actuelles</p>
--	---

		d'étudiant. d- L'administrateur valide la suppression.
	3	Le système enregistre les modifications.
Scénario	Etape	Action
Alternatif	2	<p>1. e. le numéro d'étudiant existe déjà : affiche un message d'existence et retour à 2.1.</p> <p>1. g. avertissement : s'il y a un autre étudiant à ajouter dans le même groupe retour à 1.e sinon s'il y a un autre étudiant à ajouter dans un autre groupe retour à 1.a sinon retour à 2.</p> <p>2. c. le numéro d'étudiant n'existe pas : affiche un message d'absence et retour à 2.2.</p> <p>3. c. le numéro d'étudiant n'existe pas : affiche un message d'absence et retour à 3.2.</p>
Scenario	Le cas où L'administrateur annule l'opération retour à 2 le cas :	
Erreurs	Initialiser l'application.	

#### 4.10.Initialiser les enseignants :

Cas d'utilisation	Initialiser les enseignants.
Acteur	L'administrateur.
But	Ajouter, modifier ou supprimer un enseignant.
Pré Conditions	L'administrateur possède les informations d'un enseignant.
Post Conditions	Les informations ont été enregistrées.

Scénario	Etape	Action
Nominal	1	Le système présente l'état actuel d'enseignant.
	2	<p>L'administrateur choisit une opération de gestion d'enseignant « ajout, modification ou suppression ».</p> <p><b>1. Cas d'ajout:</b></p> <p>a- Le système demande de saisir le code d'enseignant, nom, prénom, grade, et matière</p> <p>b- L'administrateur saisie le code, nom, prénom, matière et valide l'ajout.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le code d'enseignant.</p> <p>c- L'administrateur saisi le code d'enseignant.</p> <p>d- Le système présente les informations actuelles d'enseignant.</p> <p>e- Système demande de saisir les nouvelles informations.</p> <p>f- L'administrateur saisie les informations et valide la modification.</p> <p><b>3. Cas de suppression:</b></p> <p>a- Le système demande de saisir le code d'enseignant.</p> <p>b- L'administrateur saisi le code d'enseignant.</p> <p>c- Le système présente les informations actuelles d'enseignant.</p> <p>d- L'administrateur valide la suppression.</p>
	3	Le système enregistre les modifications.
Scénario Alternatif	Etape	Action
	2	<p>1. b. le code d'enseignant existe déjà : affiche un message d'existence et retour à 2.1.</p> <p>2. c. le numéro d'enseignant n'existe pas : affiche un message d'absence et retour à 2.2.</p> <p>3. c. le numéro d'enseignant n'existe pas : affiche un</p>

		message d'absence et retour à 3.2.
Scenario Erreurs	Le cas où L'administrateur annule l'opération retour à 2 le cas : Initialiser l'application.	

#### 4.11.Initialiser les domaines :

Cas d'utilisation	Initialiser les domaines.	
Acteur	L'administrateur.	
But	Ajouter ou modifier un domaine.	
Pré Conditions	L'administrateur possède les informations de domaine.	
Post Conditions	Les informations ont été enregistrées.	
Scénario nominal	Etape	Action
	1	Le système présente l'état actuel de domaine.
	2	<p>L'administrateur choisit une opération de domaine « ajout ou modification ».</p> <p><b>1. Cas d'ajout:</b></p> <p>a- Le système demande de saisir le code et les informations de domaine.</p> <p>b- L'administrateur saisie le code et les informations de domaine et valide l'ajout.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le code de domaine.</p> <p>b- L'administrateur saisi le code de domaine.</p> <p>c- Le système présente les informations actuelles de domaine.</p> <p>d- Système demande de saisir les nouvelles informations.</p> <p>e- L'administrateur saisi les informations et valide la modification.</p>

	3	Le système enregistre les modifications.
Scénario Alternatif	Etape	Action
	2	1. b. le code de domaine existe déjà : affiche un message d'existence et retour à 2.1. 2. c. le code de domaine n'existe pas : affiche un message d'absence et retour à 2.2.
Scenario Erreurs	Le cas où L'administrateur annule l'opération retour à 2 le cas : Initialiser l'application.	

#### 4.12.Initialiser le parcours :

Cas d'utilisation	Initialiser le parcours.	
Acteur	L'administrateur.	
But	ajouter, modifier un parcours.	
Pré Conditions	L'administrateur possède les informations des parcours.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	Le système présente l'état actuel de parcours.
	2	L'administrateur choisit une opération de parcours « ajout ou modification ».  <b>1. Cas d'ajout:</b> a- Le système demande de sélectionner le domaine. b- L'administrateur sélectionne un domaine. c- Le système demande de valider la sélection. d- L'administrateur valide la sélection. e- Le système demande de saisir le nom et les informations de parcours. f- L'administrateur saisie le nom et les informations de parcours et valide l'ajout.

		<p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le nom de parcours.</p> <p>b- L'administrateur saisie le nom de parcours.</p> <p>c- Le système présente les informations actuelles de parcours.</p> <p>d- Système demande de saisir les nouvelles informations.</p> <p>e- L'administrateur saisie les informations et valide la modification.</p>
	3	Le système enregistre les modifications.
Scénario Alternatif	Etape	Action
	2	<p>1. f. le nom de parcours existe déjà : affiche un message d'existence et retour à 2.1.</p> <p>2. c. le nom de parcours n'existe pas : affiche un message d'absence et retour à 2.2.</p>
Scenario Erreurs	Le cas où L'administrateur annule l'opération retour à 2 le cas : Initialiser l'application.	

#### 4.13.Initialiser les spécialités :

Cas d'utilisation	Initialiser les spécialités.	
Acteur	L'administrateur.	
But	Ajouter ou modifier une spécialité.	
Pré Conditions	L'administrateur possède les informations des spécialités.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	Le système présente l'état actuel d'une spécialité.
	2	<p>L'administrateur choisit une opération de parcours « ajout ou modification ».</p> <p><b>1. Cas d'ajout:</b></p> <p>a- Le système demande de sélectionner le domaine et le parcours.</p> <p>b- L'administrateur sélectionne un domaine et un</p>

		<p>parcours.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système demande de saisir le nom et les informations de spécialité.</p> <p>d- L'administrateur saisie le nom et les informations d'une spécialité et valide l'ajout.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le nom d'une spécialité.</p> <p>b- L'administrateur saisie le nom de la spécialité.</p> <p>c- Le système présente les informations actuelles de la spécialité.</p> <p>d- Le système demande de saisir les nouvelles informations.</p> <p>d- L'administrateur saisie les informations et valide la modification.</p>
	3	Le système enregistre les modifications.
Scénario Alternatif	Etape	Action
	2	<p>1. f. le nom de la spécialité existe déjà : affiche un message d'existence et retour à 2.1.</p> <p>2. c. le nom de la spécialité n'existe pas : affiche un message d'absence et retour à 2.2.</p>
Scénario Erreurs	Le cas où L'administrateur annule l'opération retour en 2 du cas : Initialiser l'application.	

#### 4.13.Initialiser le semestre :

Cas d'utilisation	Initialiser le semestre.	
Acteur	L'administrateur.	
But	Ajouter ou modifier un semestre.	
Pré Conditions	L'administrateur possède les informations d'un semestre.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	Le système présente l'état actuel d'un semestre.

	2	<p>L'administrateur choisie une opération d'un semestre « ajout ou modification ».</p> <p><b>1. Cas d'ajout:</b></p> <p>a- Le système demande de sélectionner le domaine et le niveau, le parcours, la spécialité.</p> <p>b- L'administrateur sélectionne le domaine et le niveau, le parcours, la spécialité.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système demande de saisir le numéro et les informations d'un semestre.</p> <p>f- L'administrateur saisie le numéro et les informations d'un semestre et valide l'ajout.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le numéro de semestre.</p> <p>b- L'administrateur saisie le numéro de semestre.</p> <p>c- Le système présente les informations actuelles de semestre.</p> <p>d- Le système demande de saisir les nouvelles informations de semestre.</p> <p>d- L'administrateur saisie les nouvelles informations de semestre et valide la modification.</p>
	3	Le système enregistre les modifications.
Scénario	Etape	Action
Alternatif	2	<p>1. f. le numéro de semestre existe déjà : affiche un message d'existence et retour à 2.1.</p> <p>2. c. le nom de spécialité n'existe pas : affiche un message d'absence et retour à 2.2.</p>
Scenarior D'erreurs	Le cas où L'administrateur annule l'opération retour à 2 le cas : Gestionner l'initialisation.	

#### 4.14. Initialiser les unités d'enseignement :

Cas d'utilisation	Initialiser les unités d'enseignement.
-------------------	--

Acteur	L'administrateur.	
But	Ajouter ou modifier l'unité d'enseignement.	
Pré Conditions	L'administrateur possède les informations d'une unité d'enseignement.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	Le système présente l'état actuel d'unité d'enseignement.
	2	<p>L'administrateur choisit une opération d'unité d'enseignement « ajout ou modification ».</p> <p><b>1. Cas d'ajout:</b></p> <p>a- Le système demande de sélectionner le domaine, le niveau, le parcours, la spécialité et le semestre.</p> <p>b- L'administrateur sélectionne le domaine, le niveau, le parcours, la spécialité et le semestre.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système demande de saisir le code et les informations d'unité d'enseignement.</p> <p>f- L'administrateur saisie le code et les informations d'unité d'enseignement et valide l'ajout.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le code d'unité d'enseignement.</p> <p>b- L'administrateur saisie le code d'unité d'enseignement.</p> <p>c- Le système présente les informations actuelles de l'unité d'enseignement.</p> <p>d- Le système demande de saisir les nouvelles informations de l'unité d'enseignement.</p> <p>e- L'administrateur saisie les nouvelles informations de l'unité d'enseignement et valide la modification.</p>

	3	Le système enregistre les modifications.
Scénario Alternatif	Etape	Action
	2	1. f. le code d'unité d'enseignement existe déjà : affiche un message d'existence et retour à 2.1. 2. c. le code de l'unité d'enseignement n'existe pas : affiche un message d'absence et retour à 2.2. 2. e. avertissement : la modification de l'unité d'enseignement modifie toutes les informations de cette unité d'enseignement si oui modifié si non retour à 2.1.
Scenario Erreurs	Le cas où L'administrateur annule l'opération retour à 2 le cas : Initialiser l'application.	

#### 4.15. Initialiser les matières :

Cas d'utilisation	Initialiser les matières.	
Acteur	L'administrateur.	
But	Ajouter ou modifier une matière.	
Pré Conditions	L'administrateur possède les informations d'une matière.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	Le système présente l'état actuel de matière.
	2	L'administrateur choisit une opération de matière « ajout ou modification ». <b>1. Cas d'ajout:</b> a- Le système demande de sélectionner le domaine, le niveau, le parcours, la spécialité, le semestre et l'unité d'enseignement. b- L'administrateur sélectionne le domaine, le parcours, la spécialité, le niveau, le semestre et l'unité d'enseignement. c- Le système demande de valider la sélection. d- L'administrateur valide la sélection. e- Le système demande de saisir le code et les informations de matière.

		<p>f- L'administrateur saisie le code et les informations de matière et valide l'ajout.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le code de matière.</p> <p>b- L'administrateur saisi le code de matière.</p> <p>c- Le système présente les informations actuelles de matière.</p> <p>d- Le système demande de saisir les nouvelles informations de matière.</p> <p>e- L'administrateur saisie les nouvelles informations de matière et valide la modification.</p>
	3	Le système enregistre les modifications.
Scénario Alternatif	Etape	Action
	2	<p>1. f. le code de matière existe déjà : affiche un message d'existence et retour à 2.1.</p> <p>2. c. le code de matière n'existe pas : affiche un message d'absence et retour à 2.2.</p> <p>2. e. avertissement : la modification de la matière modifie toutes les informations de cette matière si oui modifier si non retour à 2.1.</p>
Scenario Erreurs	Le cas où L'administrateur annule l'opération retour à 2 le cas : Initialiser l'application.	

**4.16. Initialiser les sections :**

Cas d'utilisation	Initialiser les sections.	
Acteur	L'administrateur.	
But	Ajouter, supprimer ou modifier une section.	
Pré Conditions	L'administrateur possède les informations d'une section.	
Post Conditions	Les informations ont été enregistrées.	
Scénario	Etape	Action

Nominal	1	Le système présente l'état actuel de section.
	2	<p>L'administrateur choisie une opération de section « ajout, modification ou suppression ».</p> <p><b>1. Cas d'ajout:</b></p> <p>a- Le système demande de sélectionner le niveau, le domaine, le parcours et la spécialité.</p> <p>b- L'administrateur sélectionne le niveau, le domaine, le parcours et la spécialité.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p> <p>e- Le système demande de saisir le code et les informations de section.</p> <p>f- L'administrateur saisie le code et les informations de section et valide l'ajout.</p> <p><b>2. Cas de modification :</b></p> <p>a- Le système demande de saisir le code de section.</p> <p>b- L'administrateur sélectionne le code de section.</p> <p>c- Le système présente les informations actuelles de section.</p> <p>d- Le système demande de saisir les nouvelles informations de section.</p> <p>e- L'administrateur saisie les informations de section et valide la modification.</p> <p><b>3. Cas de suppression:</b></p> <p>a- Le système demande de saisir le code de section.</p> <p>b- L'administrateur saisi le code de section.</p> <p>c- Le système présente les informations actuelles de section.</p> <p>d- L'administrateur valide la suppression.</p>
	3	Le système enregistre les modifications.
Scénario	Etape	Action

Alternatif	2	<p>1. f. le code de la section existe déjà : affiche un message d'existence et retour à 2.1.</p> <p>2. c. le code de la section n'existe pas : affiche un message d'absence et retour à 2.2.</p> <p>3. c. le code de la section n'existe pas : affiche un message d'absence et retour à 2.3.</p> <p>3. d. avertissement : la suppression de la section supprime toutes les informations de cette section dans le système si oui supprimé sinon retour à 2.3.</p>
Scénario Erreurs	Le cas où L'administrateur annule l'opération retour à 2 le cas : Initialiser l'application.	

#### 4.17. Initialiser les groupes :

Cas d'utilisation	Initialiser les groupes.	
Acteur	L'administrateur.	
But	Ajouter ou supprimer, modifier un groupe.	
Pré Conditions	L'administrateur possède les informations d'un groupe.	
Post Conditions	Les informations ont été enregistrées.	
Scénario Nominal	Etape	Action
	1	Le système présente l'état actuel de groupe.
	2	<p>L'administrateur choisie une opération de groupe « ajout, modification ou suppression ».</p> <p><b>1. Cas d'ajoute:</b></p> <p>a- Le système demande de sélectionner le niveau, le domaine, le parcours, la spécialité et la section.</p> <p>b- L'administrateur sélectionne le niveau, le domaine, le parcours, la spécialité et la section.</p> <p>c- Le système demande de valider la sélection.</p> <p>d- L'administrateur valide la sélection.</p>

		<p>e- Le système demande de saisir le numéro et les informations de groupe.</p> <p>f- L'administrateur saisie le numéro et les informations de groupe et valide l'ajout.</p> <p><b>2. Cas de modification:</b></p> <p>a- Le système demande de saisir le numéro d'un groupe.</p> <p>b- L'administrateur saisie le numéro d'un groupe.</p> <p>c- Le système présente les informations actuelles d'un groupe.</p> <p>d- Le système demande de saisir les nouvelles informations de groupe.</p> <p>e- L'administrateur saisie les informations de groupe et valide la modification.</p> <p><b>3. Cas de suppression:</b></p> <p>a- Le système demande de saisir le numéro d'un groupe.</p> <p>b- L'administrateur saisie le numéro d'un groupe.</p> <p>c- Le système présente les informations de groupe.</p> <p>d- L'administrateur valide la suppression.</p>
	3	Le système enregistre les modifications.
Scénario Alternatif	Etape	Action
	2	<p>1. f. le numéro groupe existe déjà : affiche un message d'existence et retour à 2.1.</p> <p>2. c. le numéro de groupe n'existe pas : affiche un message d'absence et retour à 2.2.</p> <p>3. c. le numéro de groupe n'existe pas : affiche un message d'absence et retour à 2.3.</p> <p>3. c. avertissement : la suppression de groupe supprime toutes les informations du groupe dans le système si oui supprimé sinon retour à 2.3.</p>
Scenari Erreurs	Le cas où L'administrateur annule l'opération retour à 2 le cas : Initialiser l'application.	

## 5. Les diagrammes de séquence système :

### 5.1. Gérer l'authentification :

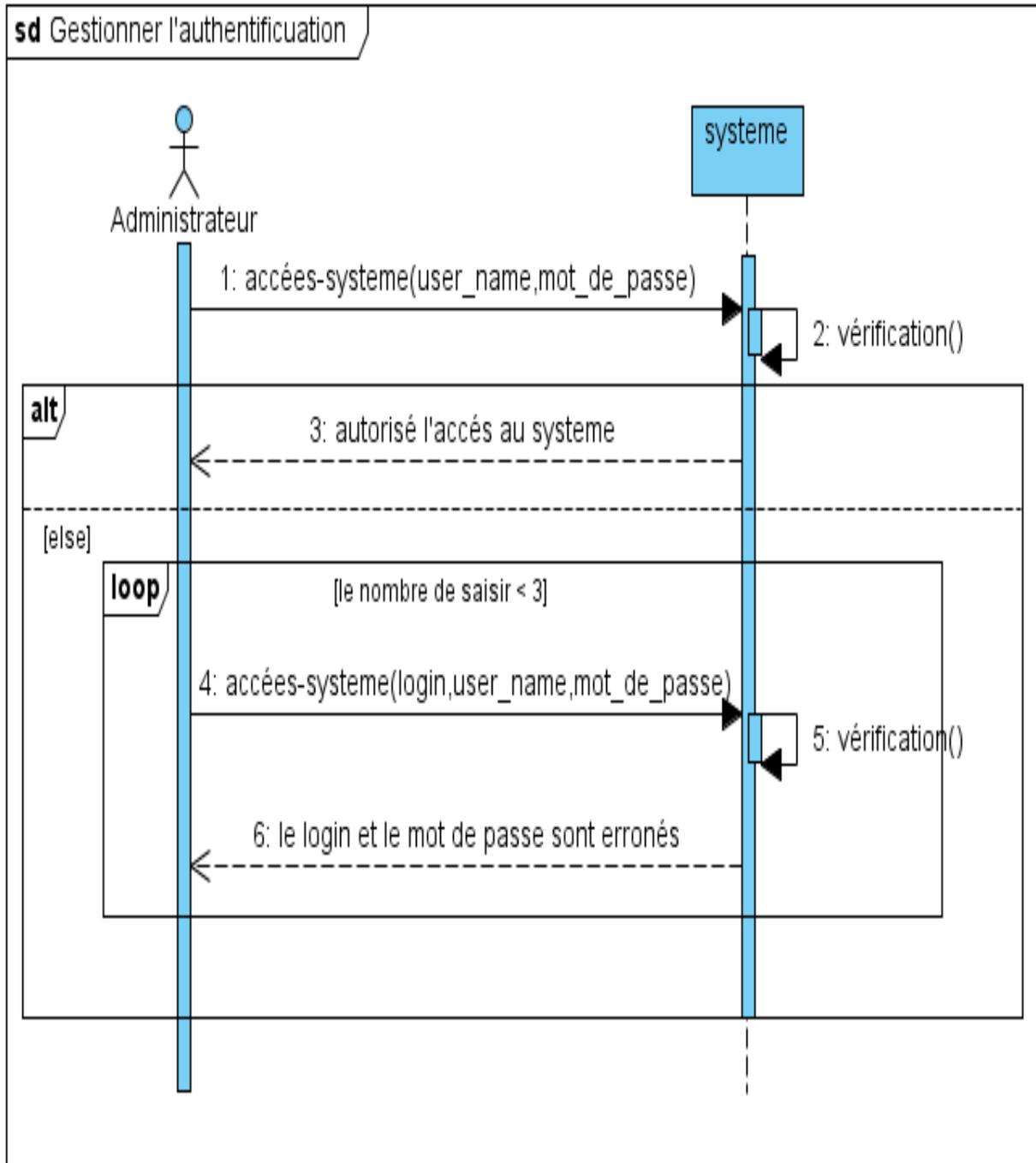


Figure 2 .2: Diagramme de : Gérer d'authentification

5.2.Gestionner la délibération:

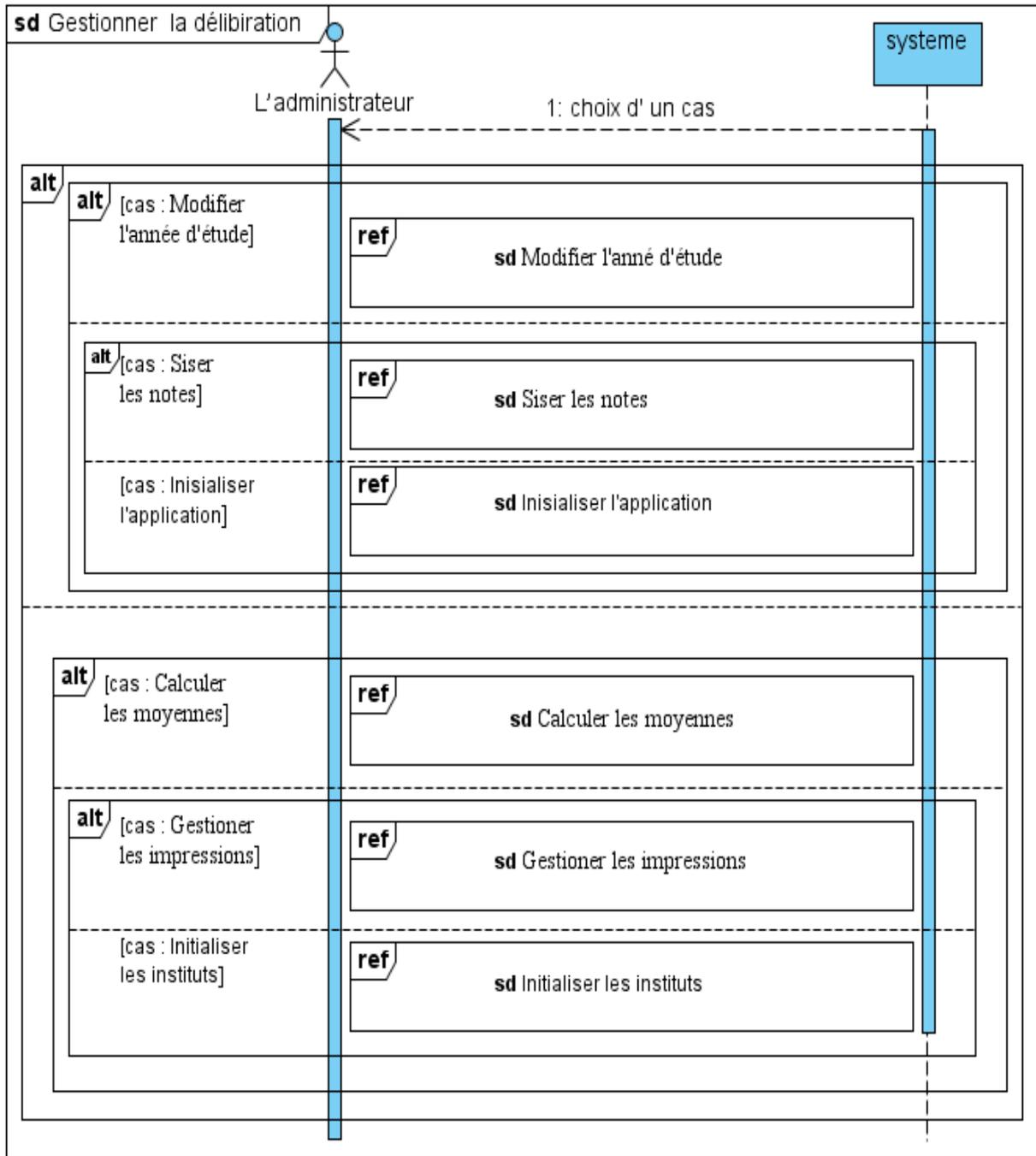


Figure 2 .3 : Diagramme de classes participantes : Gestionner la délibération.

5.3. Modifier l'année d'étude:

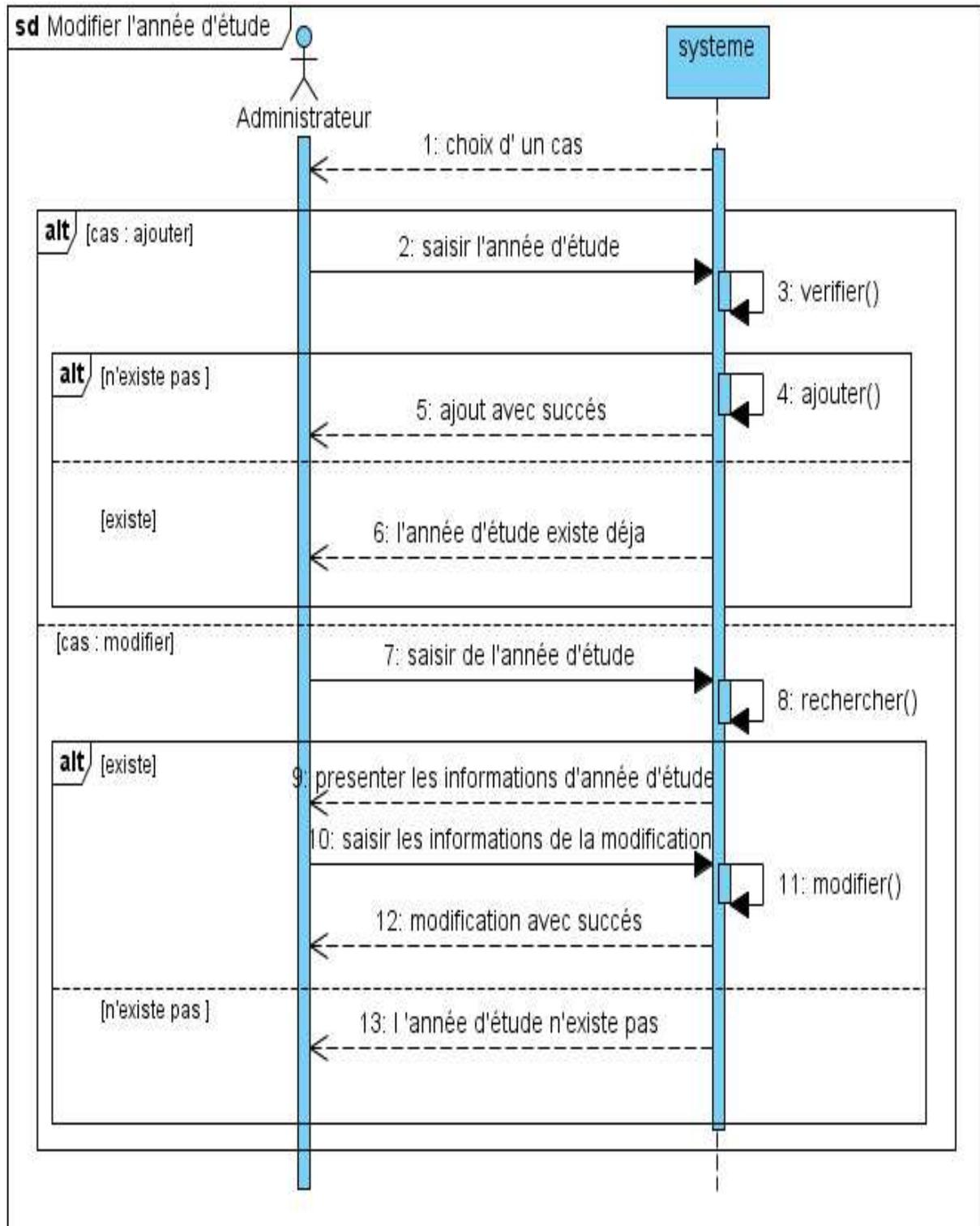


Figure 2 .4 : Diagramme de classes participantes : modifier l'année d'étude.

5.4.Initialiser les instituts :

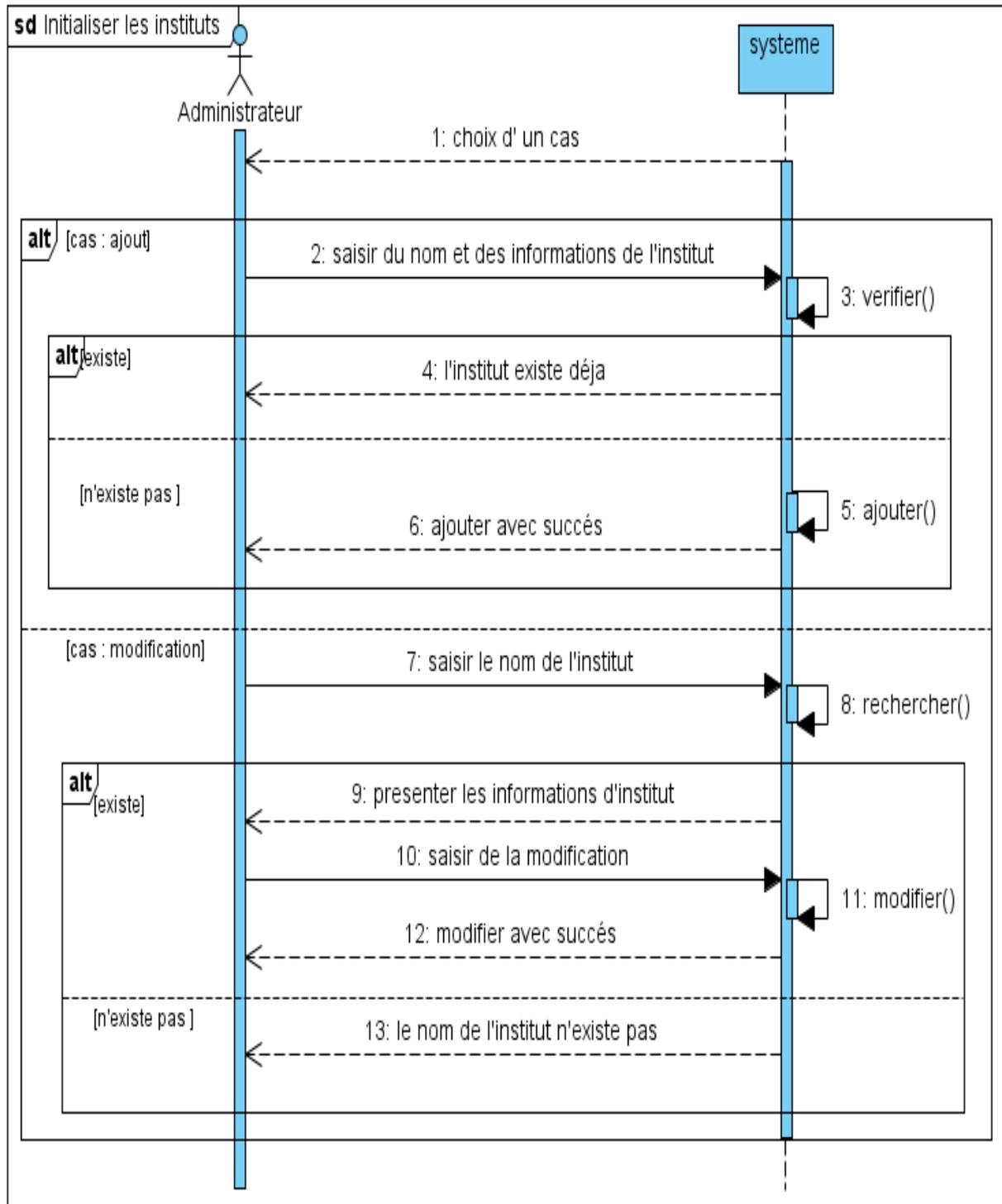
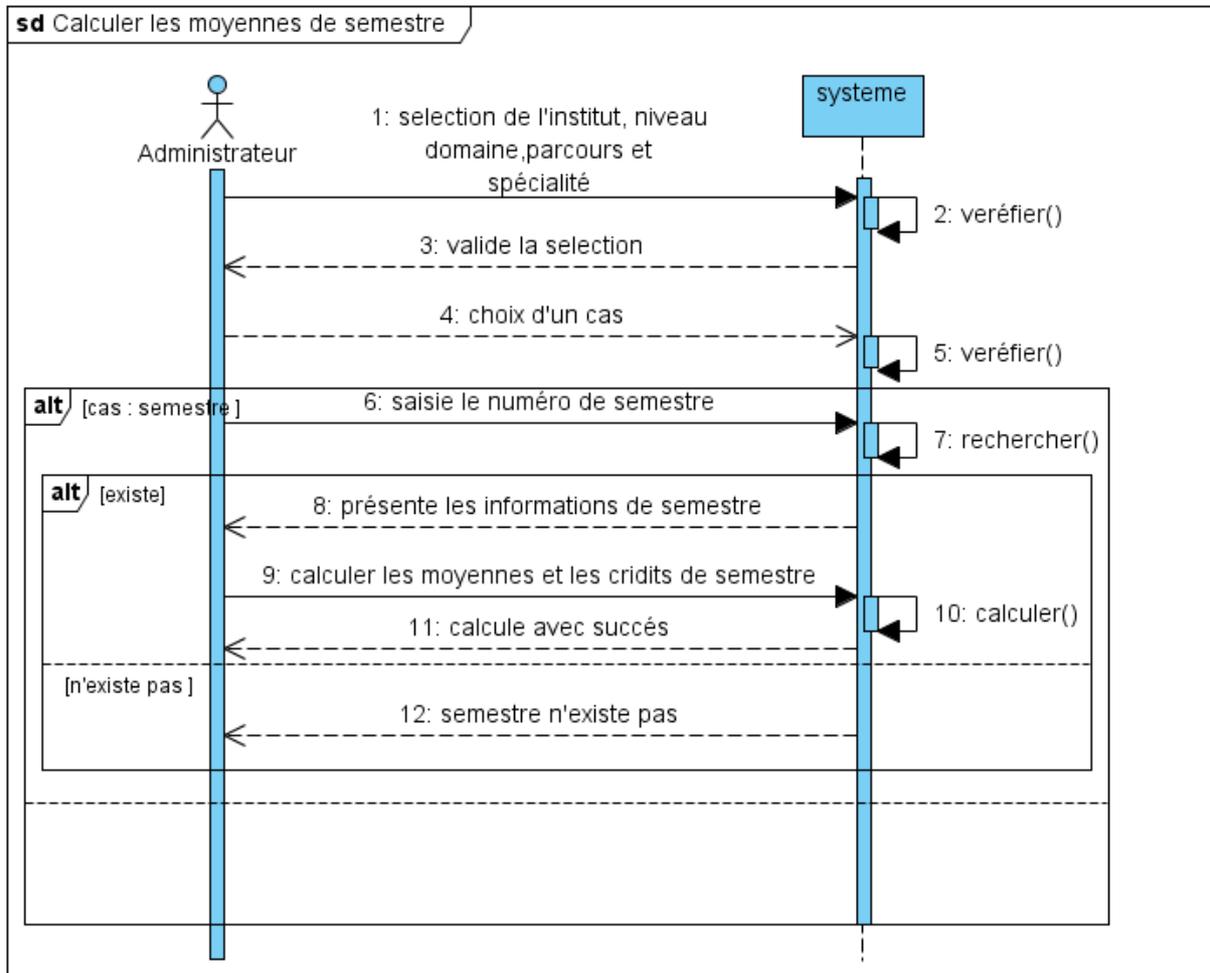


Figure 2.5 : Diagramme de classes participantes : initialiser l’institut..

## 5.5. Calculer les moyennes :

### 5.5.1. Calculer les moyens de semestre :



*Figure 2.6: Diagramme de classes participantes : calculer la moyenne < semestre >*

5.5.2.Calculer les moyennes de matières:

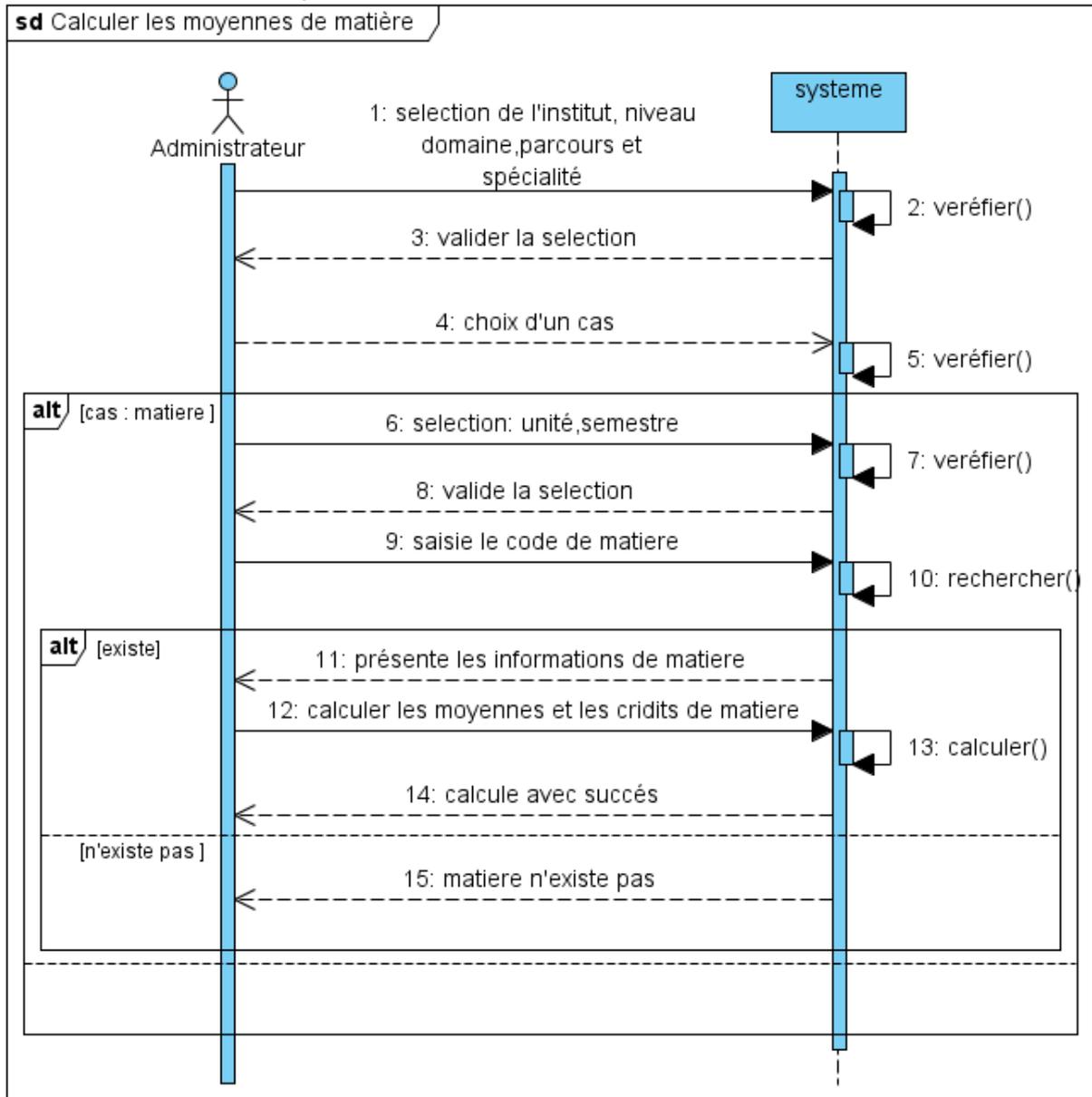


Figure 2.7: Diagramme de classes participantes : calculer les moyennes < Matières >.

5.5.3.Calculer les moyennes annuelles:

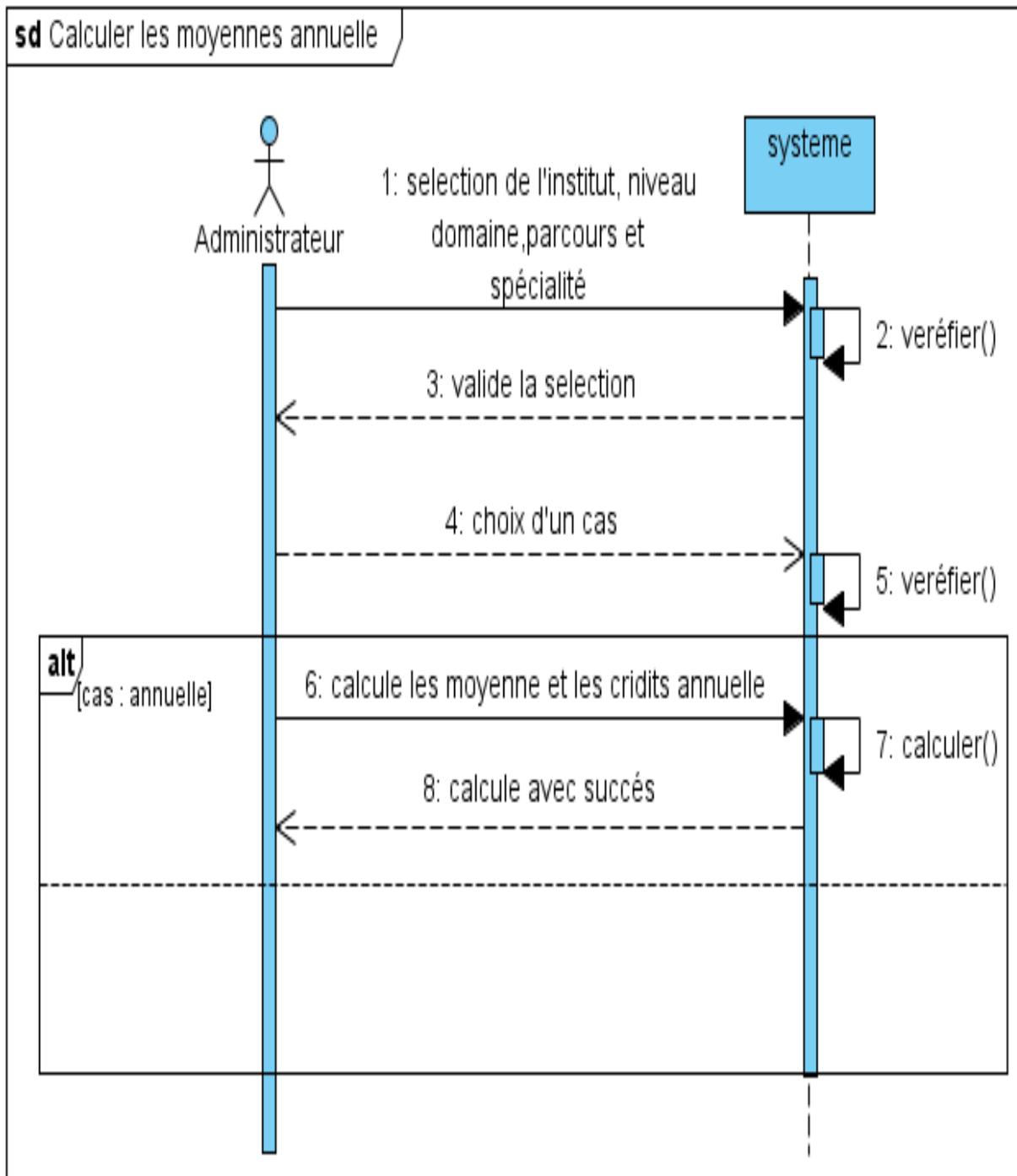


Figure 2.8: Diagramme de classes participantes : calculer les moyennes.

< Annuelles >

5.5.4.Calculer les moyennes d'unit enseignant :

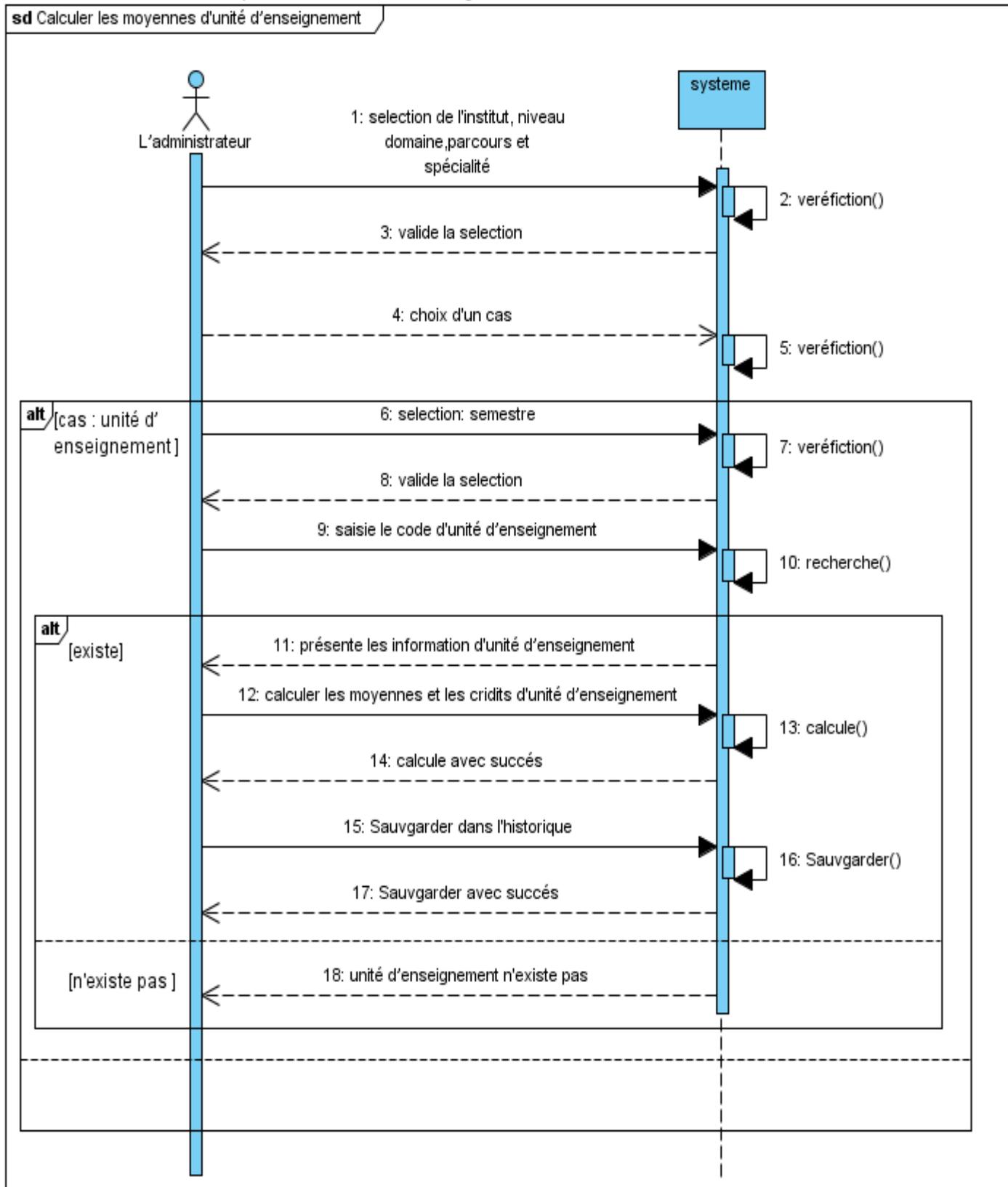
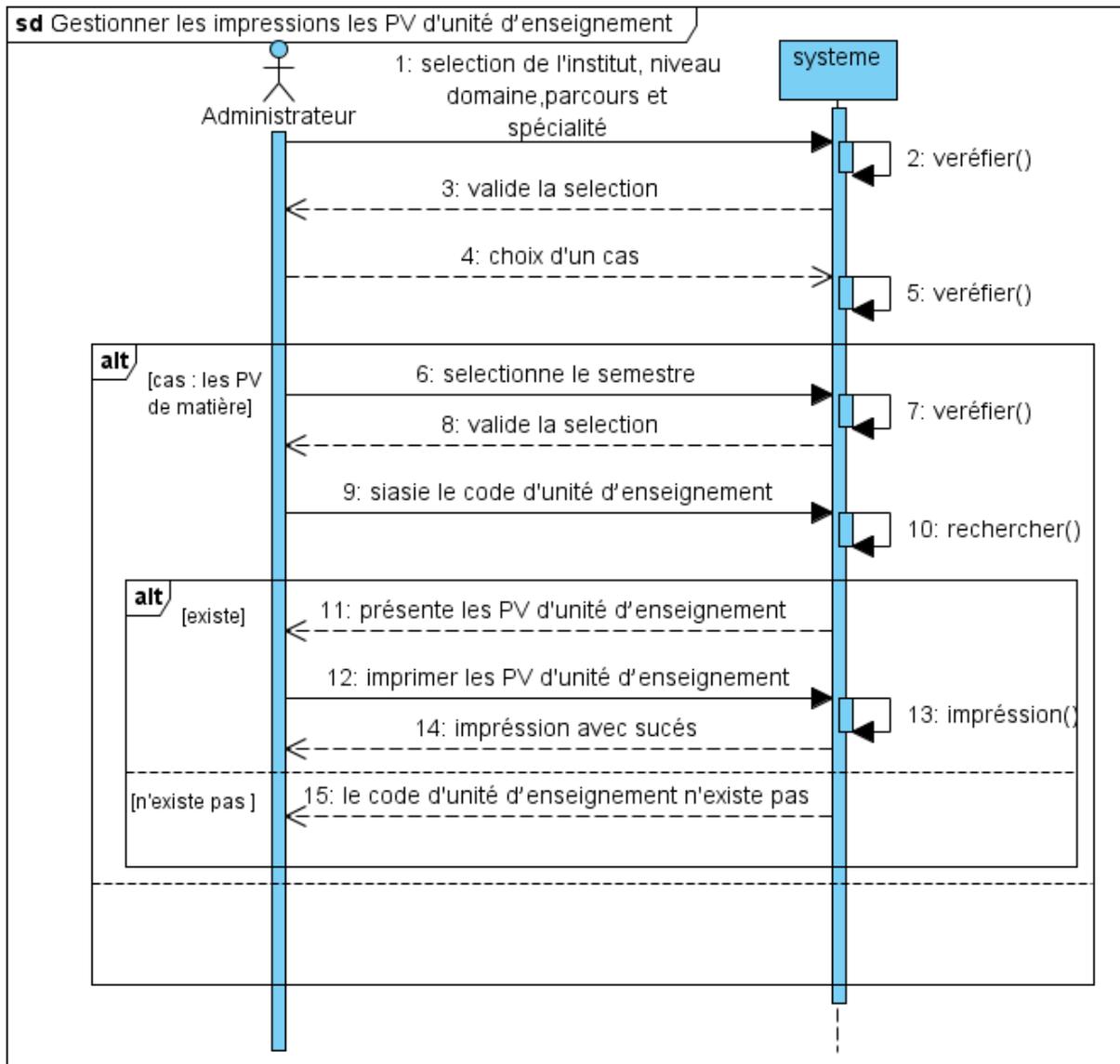


Figure 2.9: Diagramme de classes participantes : calculer les moyennes < D'unit enseignant >.

**5.6.Gérer les impressions :**

**5.6.1.Cas d'impression des PV unité :**



*Figure 2.10: Diagramme de classes participantes : gérer l'impression  
< Des PV unité >*

5.6.2.Cas d'impression des PV semestre, annuelle :

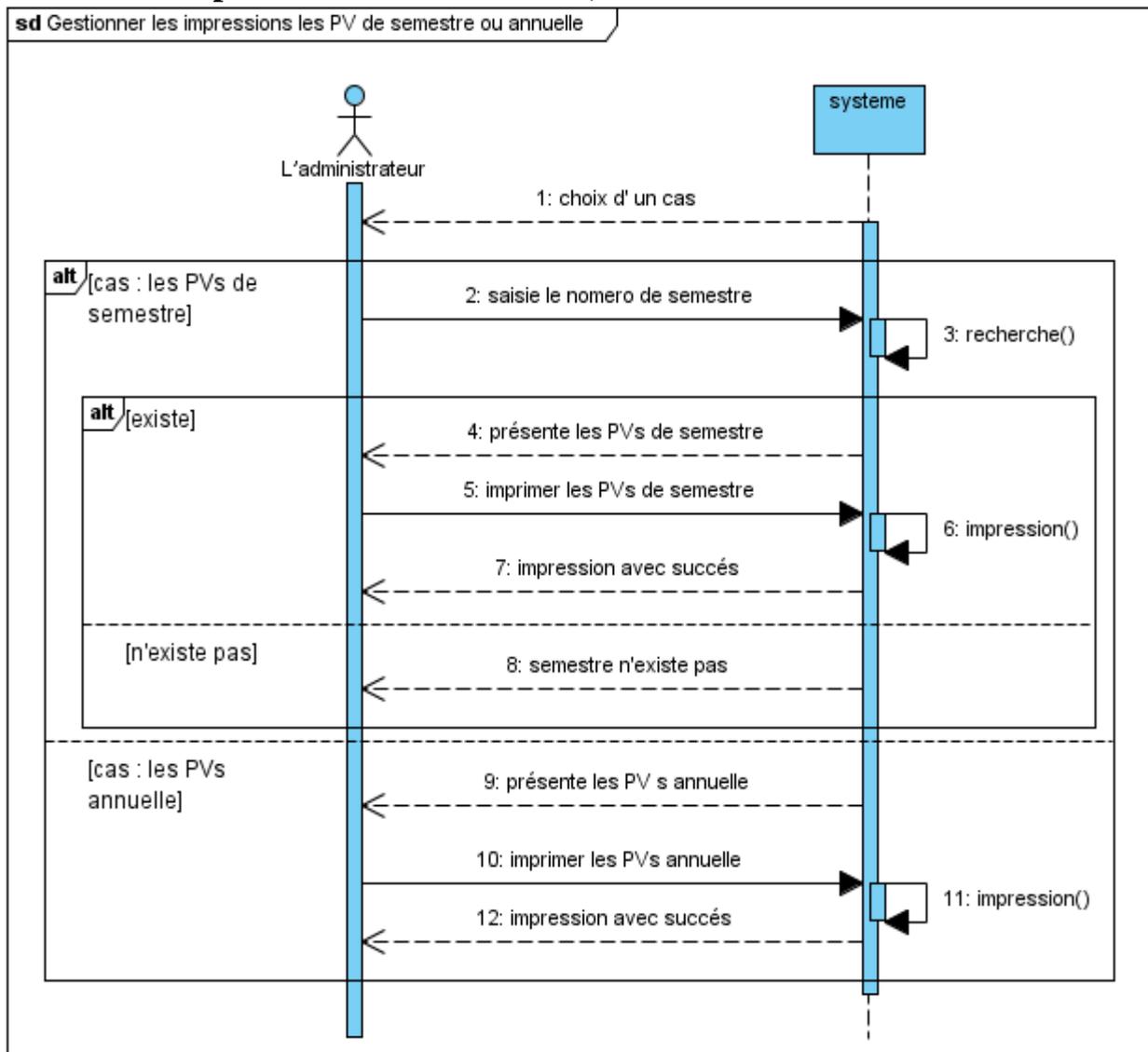


Figure 2.11: Diagramme de classes participantes : gérer l'impression  
 < PV semestre, annuelle >

5.6.3.Cas d'impression des PV de matière :

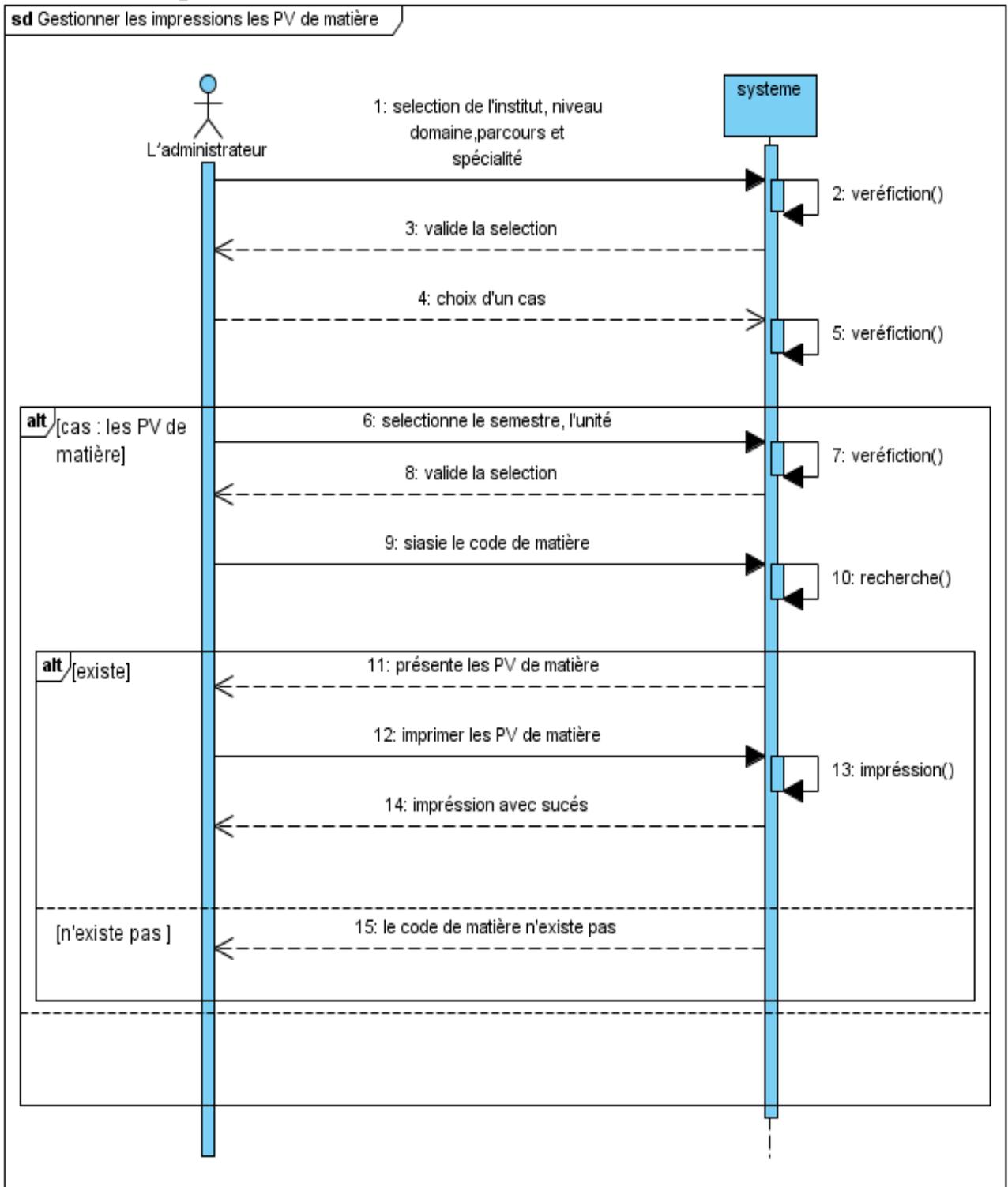


Figure 2.12: Diagramme de classes participantes : gérer l'impression < PV de matière >.

5.6.4.Cas d'impression des PV de relevé note :

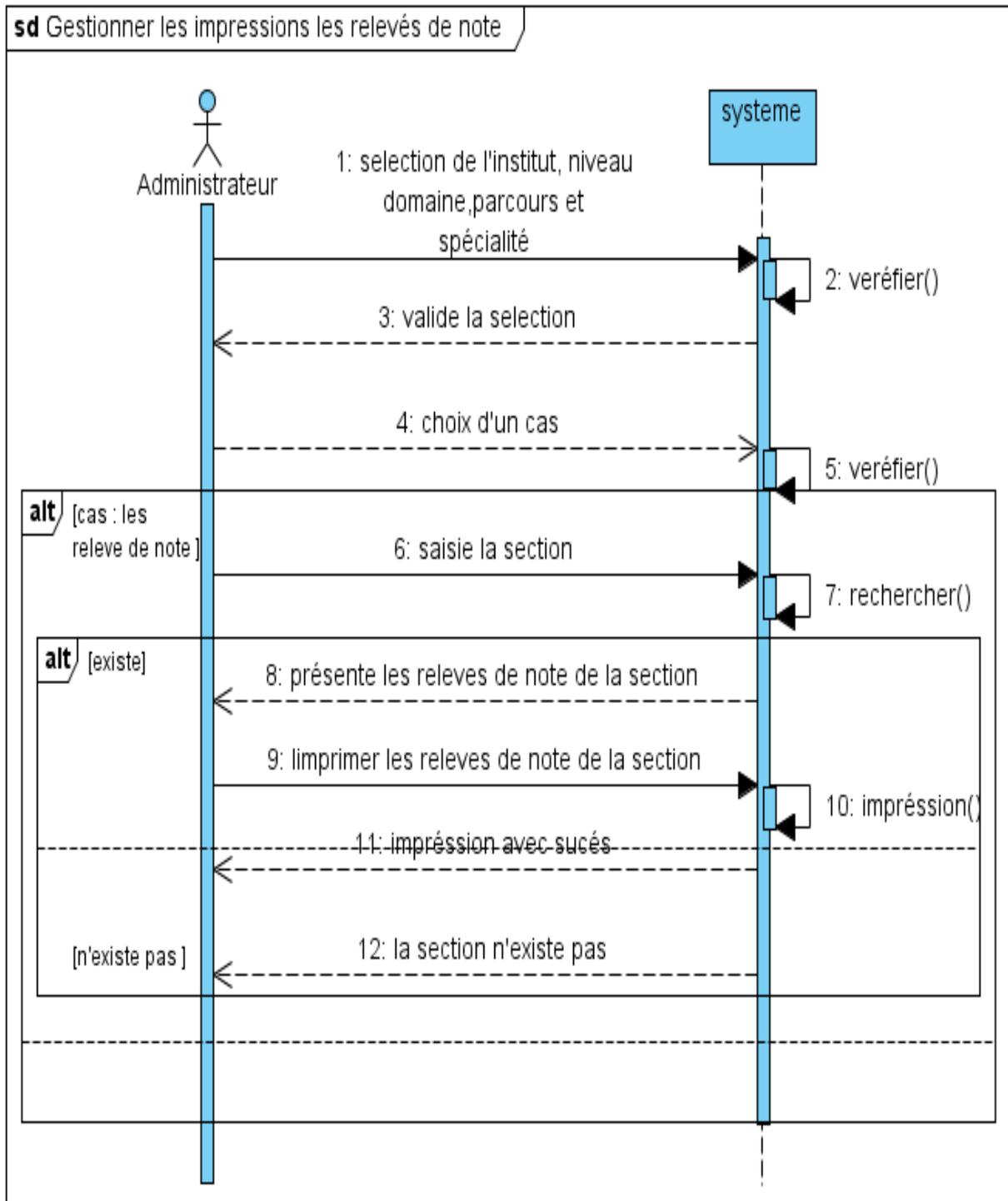


Figure 2.13: Diagramme de classes participantes : gérer l'impressions <PV de relevé note >

5.6.5.Initialiser l’application :

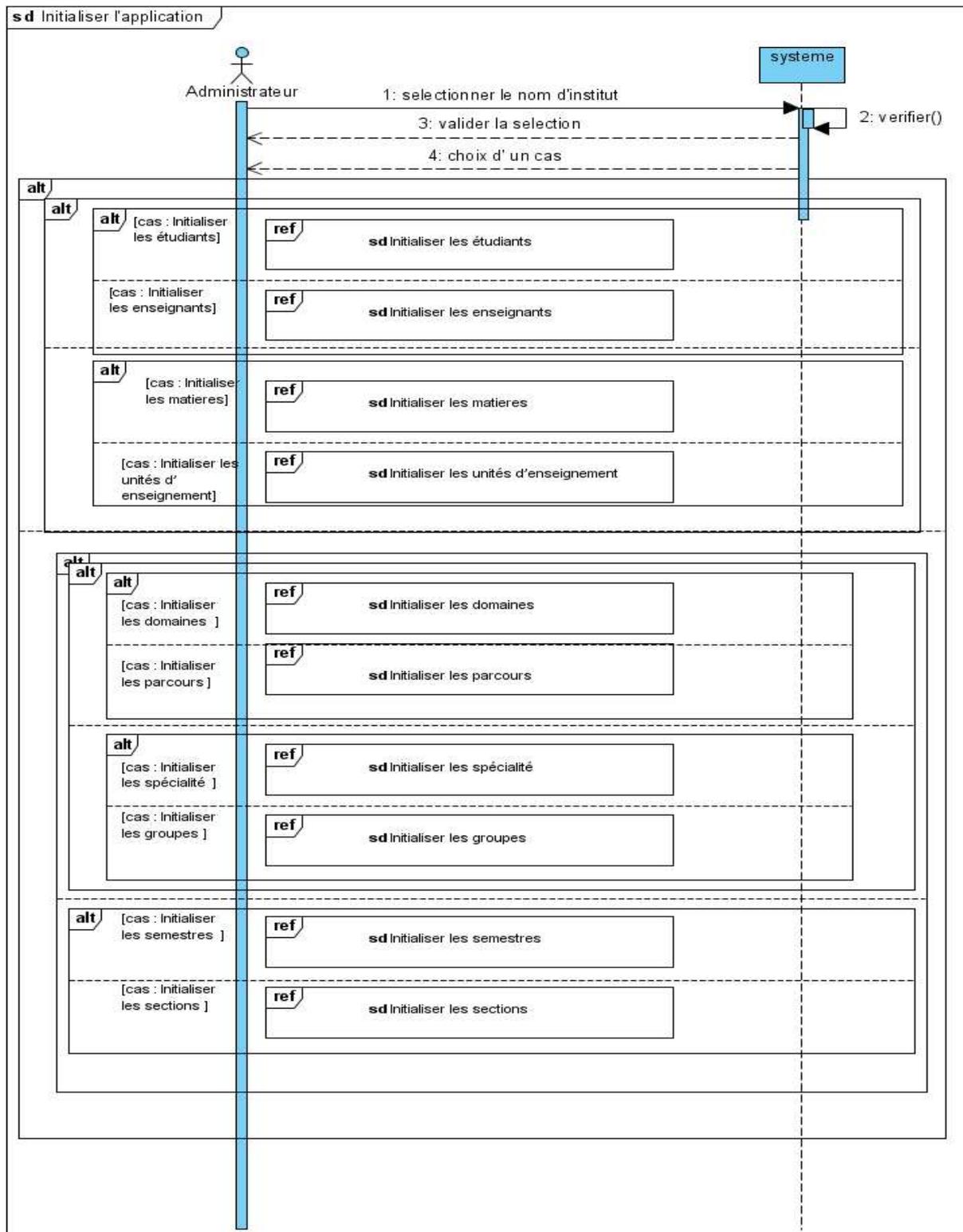


Figure 2.14: Diagramme de classes participantes : initialiser l’application.

5.7. Initialiser les sections :

5.7.1. Cas d'ajouté :

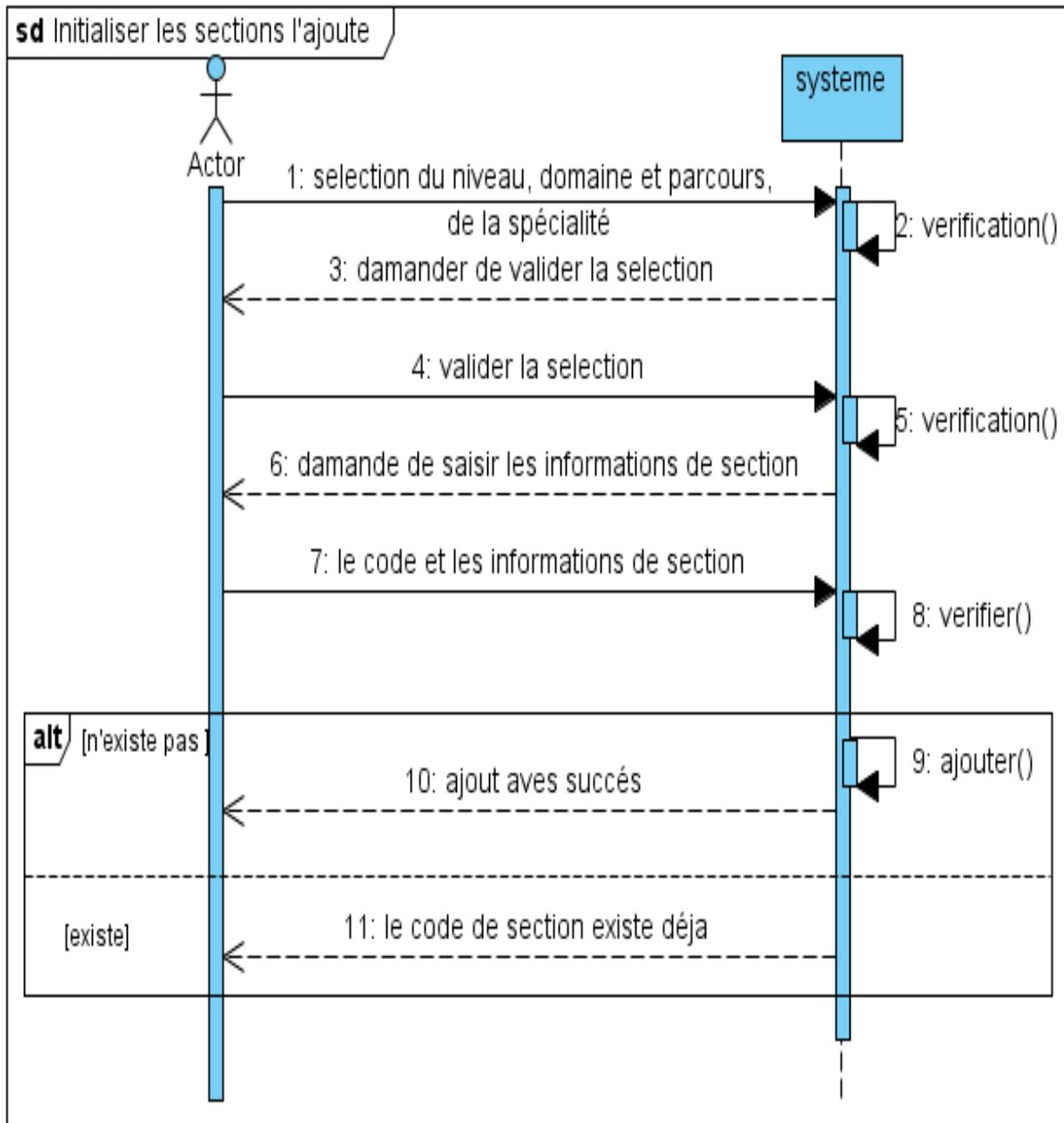
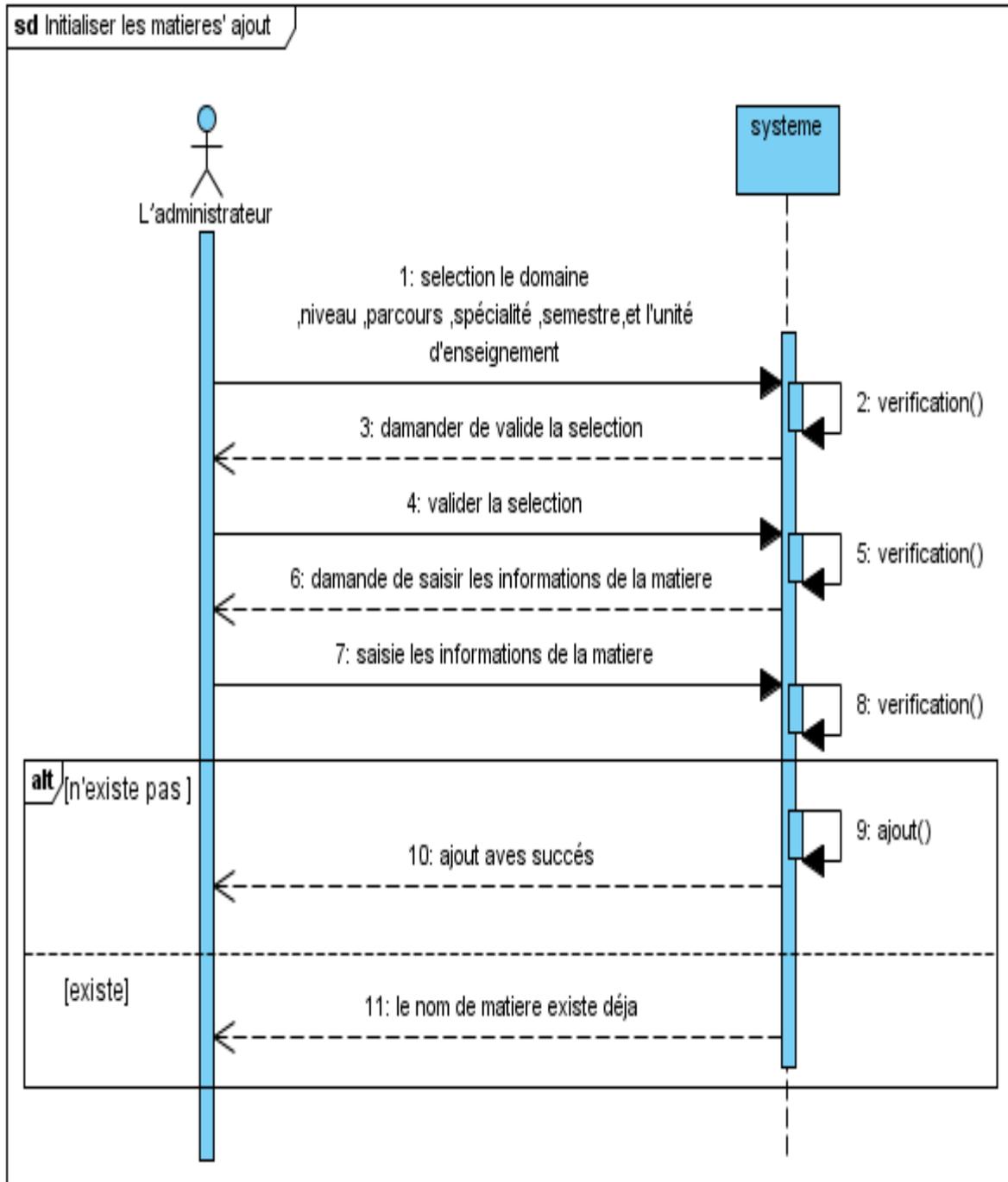


Figure 2.15: Diagramme de classes participantes : initialiser les sections.

**5.8.Initialiser les matières :**

**5.8.1.Cas d'ajouter:**



*Figure 2.16: Diagramme de classes participantes : initialiser les sections.*

5.8.2.Cas de modification :

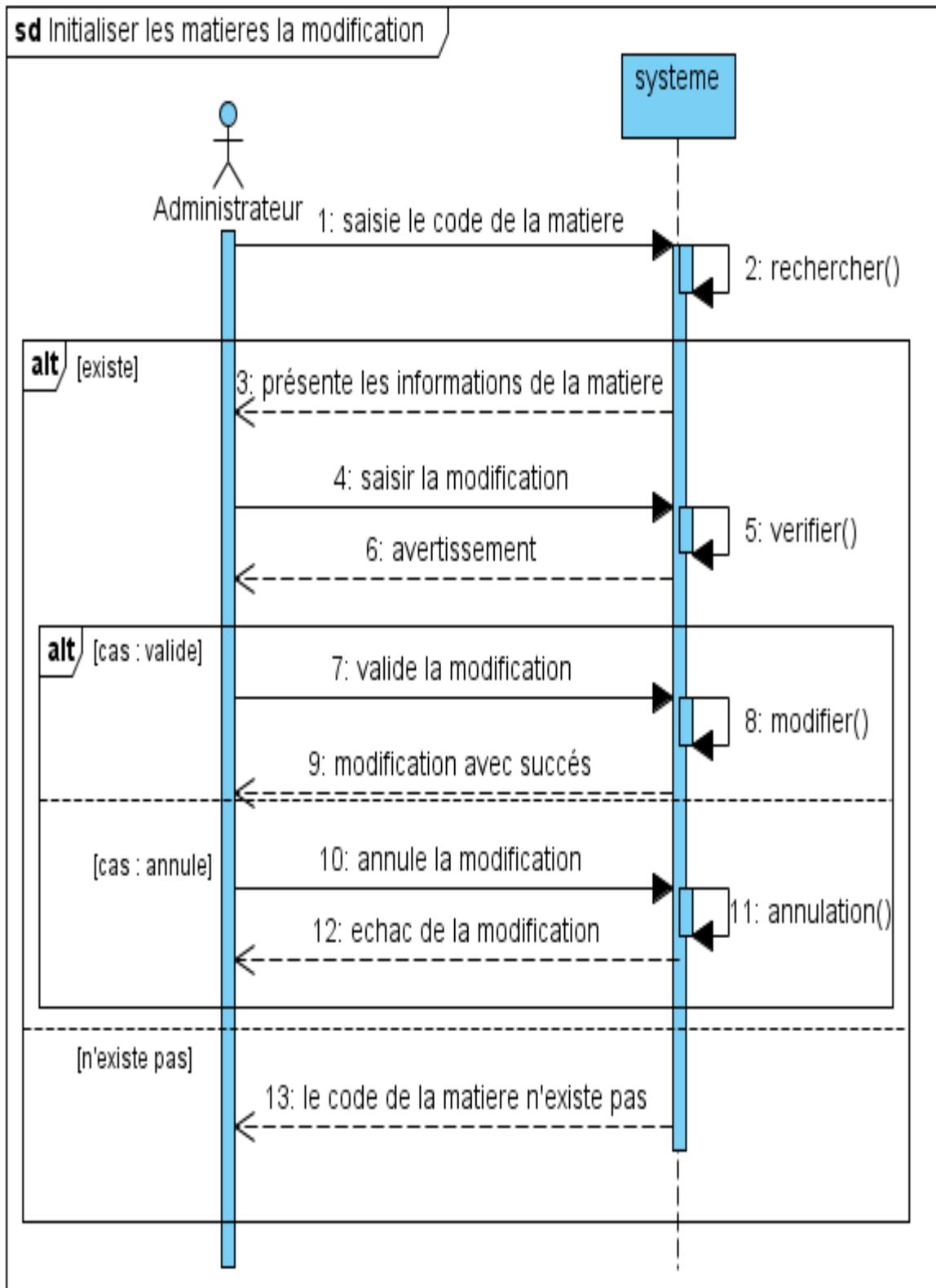


Figure 2.17: Diagramme de classes participantes : initialiser les sections.

5.9.Initialiser les etudiant :

5.9.1.Cas d'ajouter :

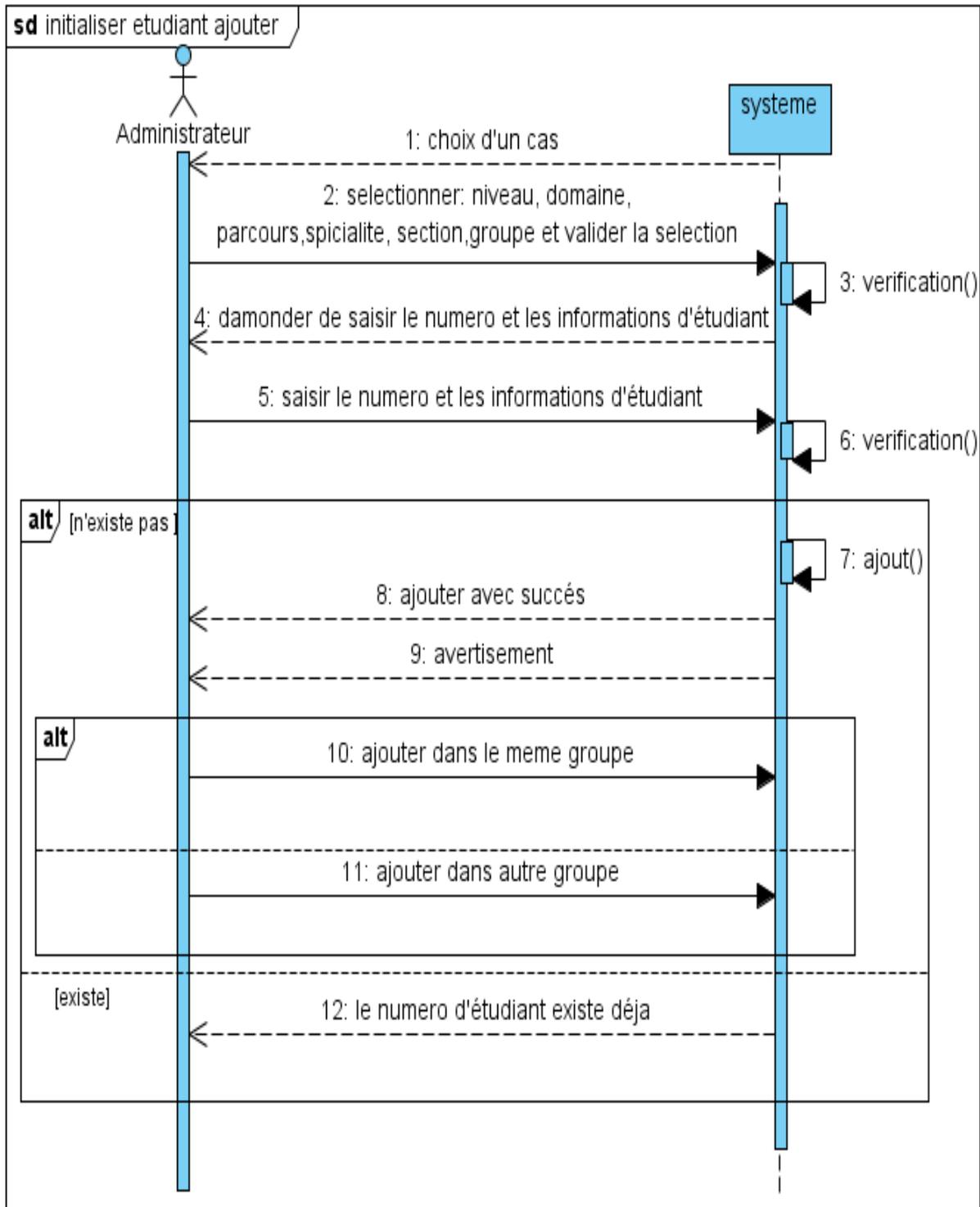


Figure 2.18: Diagramme de classes participantes : initialiser les étudiants <cas d'ajouter>.

5.9.2.Cas de modification ou suppression :

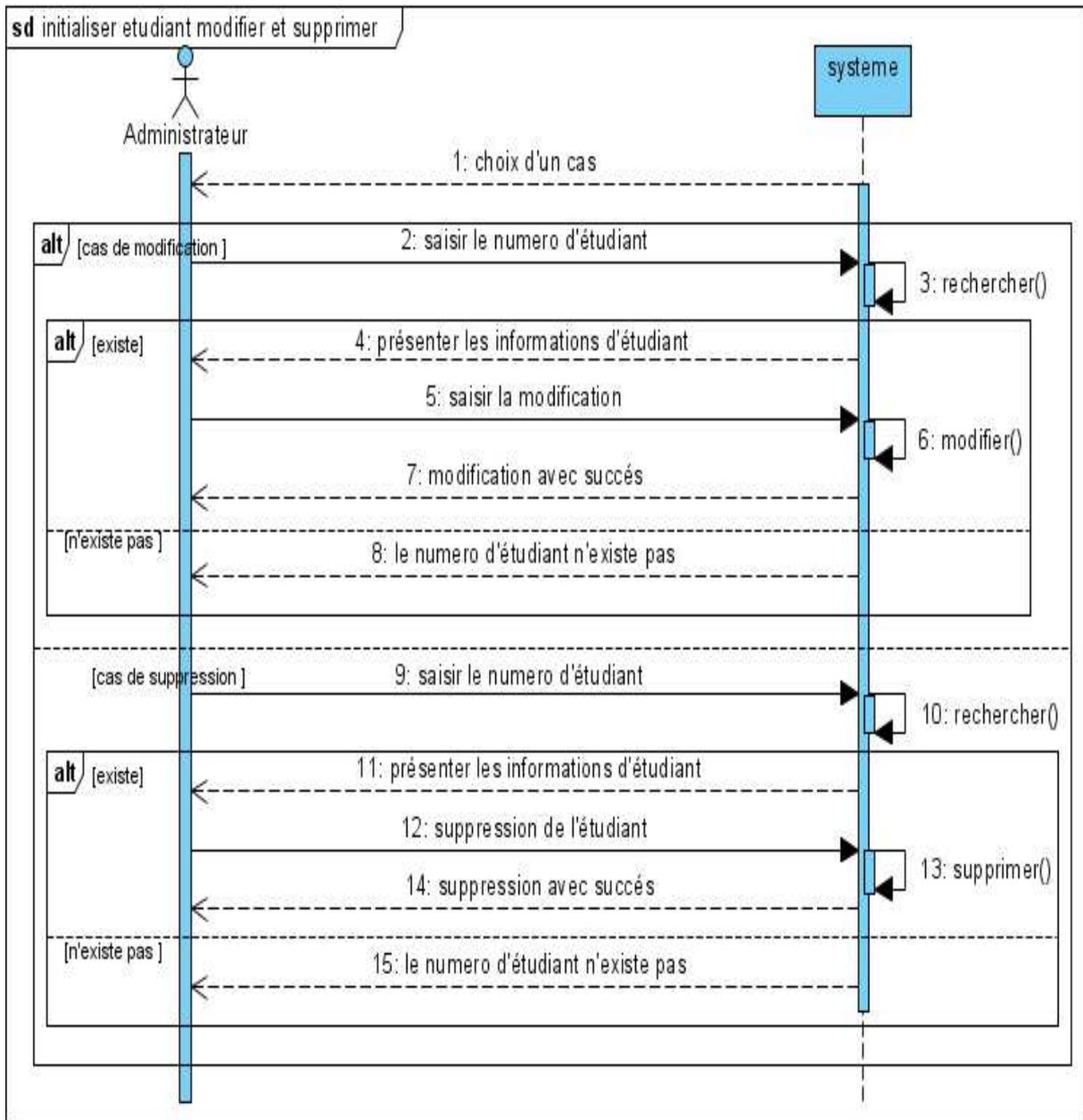
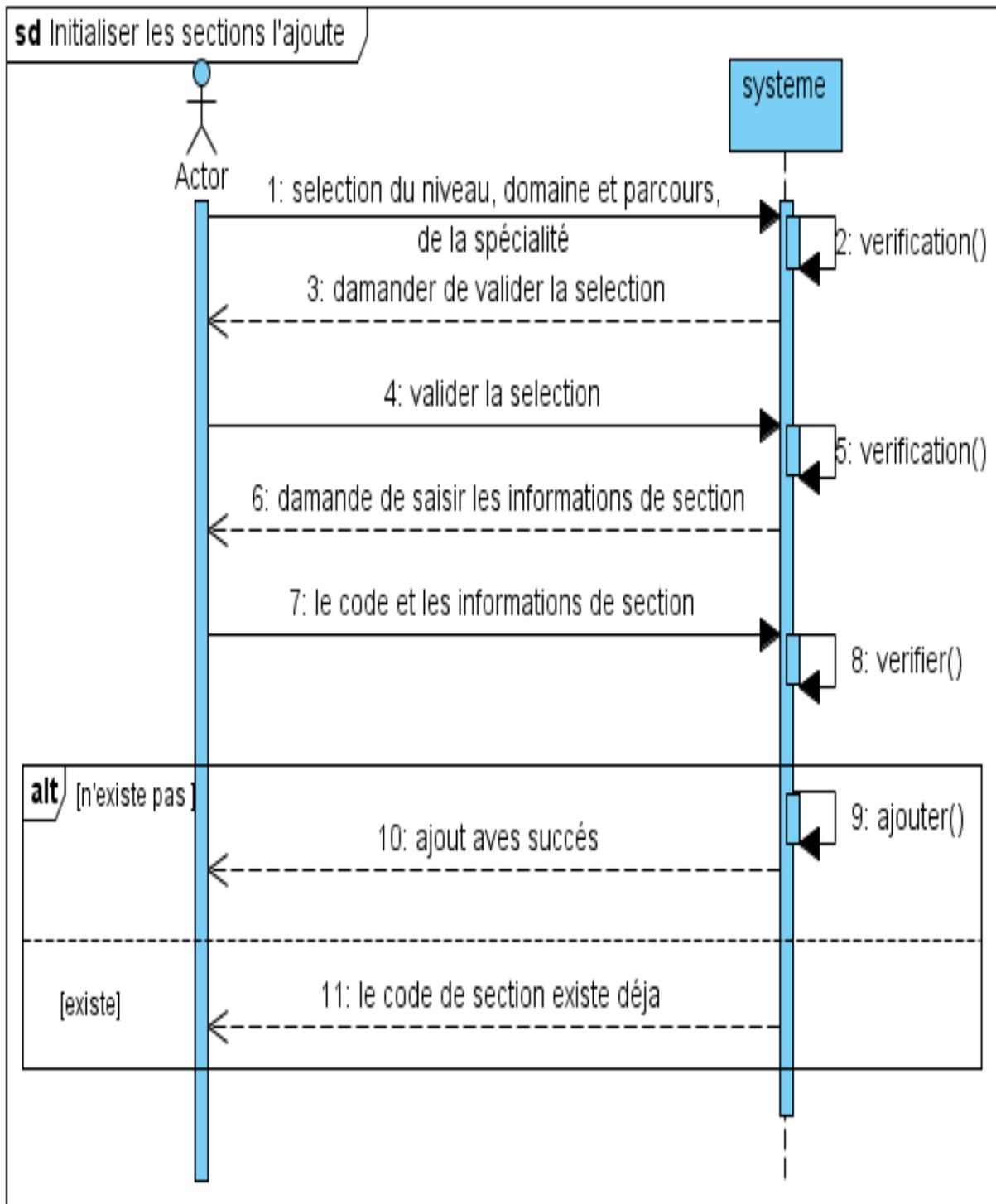


Figure 2.19: Diagramme de classes participantes : initialiser les étudiant<cas modifications ou suppression>.

## 5.10.Initialiser les sections:

## 5.10.1Cas d'ajouter :



*Figure 2.20: Diagramme de classes participantes : initialiser les sections<cas d'ajouter>.*

5.10.2.Cas de modification :

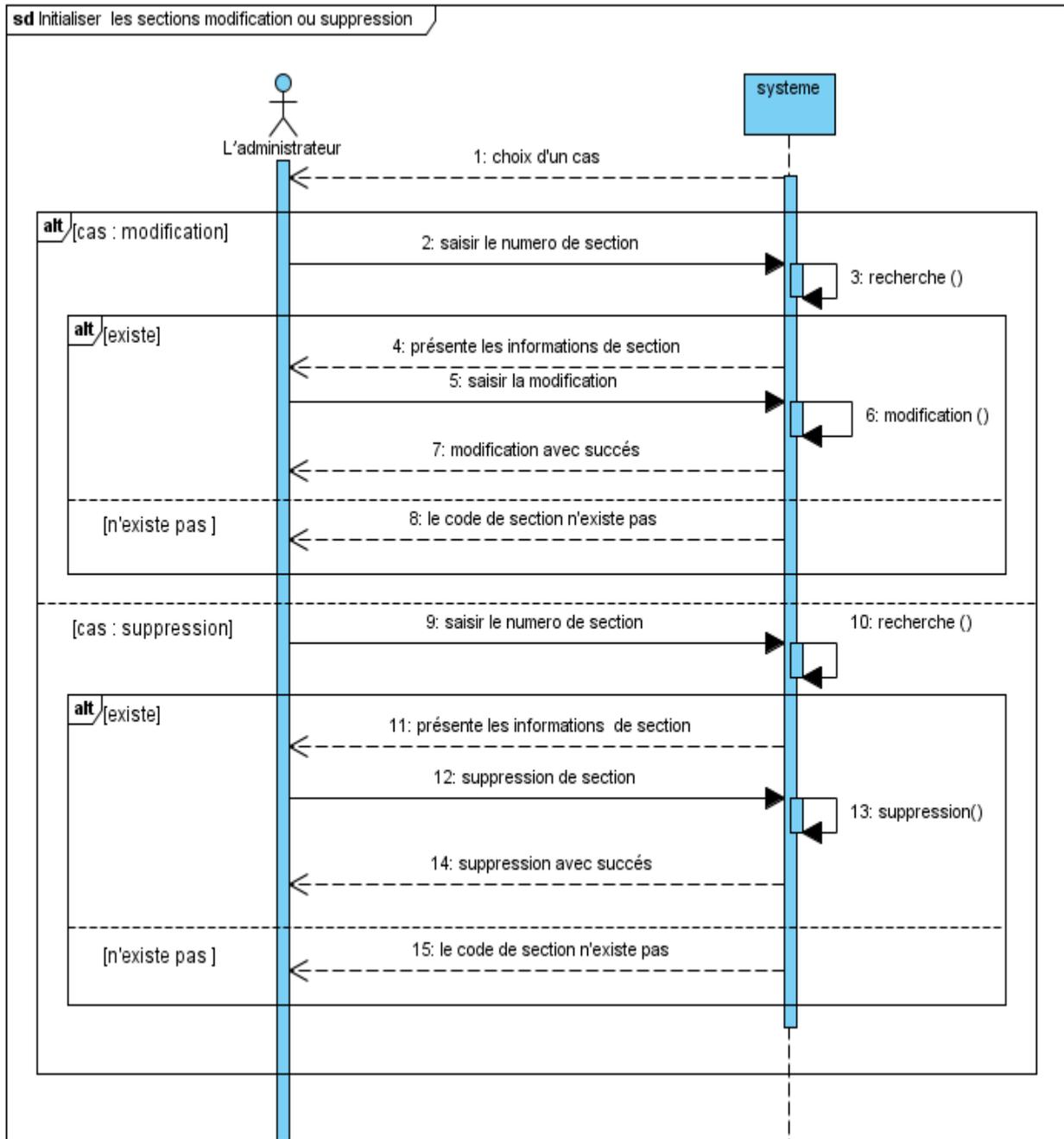


Figure 2.21: Diagramme de classes participantes : initialiser les sections<cas modifications ou suppression>.

**Conclusion :**

La phase d'identification des besoins nous a permis de modéliser le contexte du système considéré comme une boîte noire en:

- Répertoriant les interactions entre acteurs et système.
- Représentant les interactions sur le diagramme de contexte dynamique.

Ainsi l'identification de besoin nous a permis aussi de modéliser les cas d'utilisations et les diagrammes de séquences système en prennent en considération que le système a toujours vu comme une boîte noire.

Ensuite cette phase prépare la phase d'analyse qui est l'objet de chapitre suivant.

# Chapitre

## Phase d'analyse

### *- Introduction*

*1- Identification les concepts du domaine.*

*2- Affectation des attributs aux classes.*

*3- Notre modèle de domaine.*

*4-Diagrammes de classe participante.*

*-conclusion*

3

## Introduction

Dans ce chapitre nous allons présenter l'analyse du domaine qui permet d'élaborer la première version du diagramme de classes appelée modèle du domaine. Nous apprendrons à identifier les concepts du domaine à partir de l'expression initiale des besoins de notre étude de cas. Les résultats sont présentés dans les diagrammes de classes UML nommés diagrammes des classes participantes. Nous avons identifié aussi les principales classes IHM ainsi que celles qui décrivent la cinématique de l'application [3].

### ***1. Identification des concepts du domaine***

Nous allons prendre les cas d'utilisation un par un et nous posons pour chacune la question suivante : quel sont les concepts métier qui participent à ce cas d'utilisation ?

#### **• *S'authentifier***

➤ *Administrateur*

#### **• *Gestion de la scolarité***

➤ *Administrateur*

➤ *Etudiant*

➤ *Enseignant*

➤ *Niveau*

➤ *Institut*

➤ *Domaine*

➤ *Parcours*

➤ *Spécialité*

➤ *Semestre*

➤ *Matière*

➤ *Unité d'enseignement*

➤ *Section*

➤ *Groupe*

➤ *Note*

➤ *Cas de discipline*

➤ *Année d'étude*

**• Gestion d'institut**

- *Administrateur*
- *Institut*

**• Année d'étude (gestion de l'année d'étude)**

- *Administrateur*
- *Année d'étude*

**• Initialisation (initialisation des composants)**

- *Administrateur*
- *Etudiant*
- *Enseignant*
- *Institut*
- *Domaine*
- *Parcours*
- *Spécialité*
- *Semestre*
- *Matière*
- *Unité d'enseignement*
- *Section*
- *Groupe*

**• Gestion d'étudiants**

- *Administrateur*
- *Institut*
- *Niveau*
- *Domaine*
- *Parcours*
- *Spécialité*
- *Section*
- *Groupe*
- *Etudiant*

**•Gestion d'enseignants**

- *Administrateur*
- *Enseignant*
- *Matière*
- *Note*

**•Gestion du Domaines**

- *Administrateur*
- *Institut*
- *domaine*

**•Gestion de Parcours**

- *Administrateur*
- *Institut*
- *Domaine*
- *parcours*

**• Gestion de spécialités**

- *Administrateur*
- *Institut*
- *Domaine*
- *Parcours*
- *spécialité*

**• Gestion de Semestres**

- *Administrateur*
- *Institut*
- *niveau*
- *Domaine*
- *Parcours*
- *spécialité*
- *semestre*

**• Gestion d'Unité d'enseignements**

- *Administrateur*
- *Institut*
- *niveau*
- *Domaine*
- *Parcours*
- *spécialité*
- *semestre*
- *Unité d'enseignement*

• ***Gestion de matières***

- *Administrateur*
- *Institut*
- *niveau*
- *Domaine*
- *Parcours*
- *spécialité*
- *semestre*
- *Unité d'enseignement*
- *matière*

• ***Gestion de Sections***

- *Administrateur*
- *Institut*
- *niveau*
- *Domaine*
- *Parcours*
- *spécialité*
- *semestre*
- *section*

• ***Gestion de Groupes***

- *Administrateur*
- *Institut*

- *Niveau*
- *Domaine*
- *Parcours*
- *spécialité*
- *semestre*
- *section*
- *groupe*

• ***Gestion de notes***

- *Administrateur*
- *Etudiant*
- *Note*
- *Institut*
- *Domaine*
- *Niveau*
- *Semestre*
- *Unité d'enseignement*
- *Matière*
- *Section*
- *groupe*
- *Etudiant*
- *Parcours*
- *Spécialité*
- *Année d'étude*

• ***Gestion de Délibération***

- *Administrateur*
- *Institut*
- *Domaine*
- *Parcours*
- *Spécialité*
- *Niveau*

- *Semestre*
- *Etudiant*
- *Unité d'enseignement*
- *Matière*
- *Note*
- *Enseignant*
- *PV unité*
- *PV matière*
- *PV annelle*
- *PV semestre*
- ***Gestion de Cas de discipline***
  - *Administrateur*
  - *Etudiant*
  - *Cas de discipline*
- ***Impression (PV, relevée de note)***
  - *Administrateur*
  - *Etudiant*
  - *Institut*
  - *Domaine*
  - *Niveau*
  - *Section*
  - *matière*
  - *Unité d'enseignement*
  - *semestre*
  - *année*
  - *PV unité*
  - *PV matière*
  - *PV annelle*
  - *PV semestre*
  - *spécialité*

- *parcours*
- *relèves de note*
- ***Gestion de l'historique***
  - *Administrateur*
  - *Institut*
  - *Domaine*
  - *Niveau*
  - *Année d'étude*
  - *matière*
  - *Unité d'enseignement*
  - *Semestre*
  - *Etudiant*
  - *Relevé de note*
  - *PV unité*
  - *PV matière*
  - *PV annelle*
  - *PV semestre*

## ***2. Affectation des attributs aux classes***

- Etudiant {num\_etud, nom\_etud, pren\_etud, dat\_ness, nationalité, adresse, sexe, p\_père, -nom\_mère, p\_mère} .
- Enseignant {cod\_ens, nom\_ens, pren\_ens, grad, cod\_mat}
- Niveau {niv} .
- Institut {nom\_inst, nbr\_dom}.
- Domaine {cod\_dom, nom\_dom, nbr\_par}.
- Parcours {nom\_par, nbr\_spé} .
- Spécialité {nom\_spé, nbr\_sem}.
- Semestre {num\_sem, nbr\_UE} .
- Unité d'enseignement {cod\_UE, nom\_UE, coef\_UE, cred\_UE, type\_UE, nbr\_mat}.
- Matière {cod\_mat, nom\_mat, coeff\_mat, coeff\_EX, coeff\_CC, cred\_mat} .
- Section {cod\_sec, nbr\_gro} .
- Groupe {num\_gro, nbr\_etud}.
- Note {num\_not, not, typ\_not}.

### 3. Notre modèle de domaine :

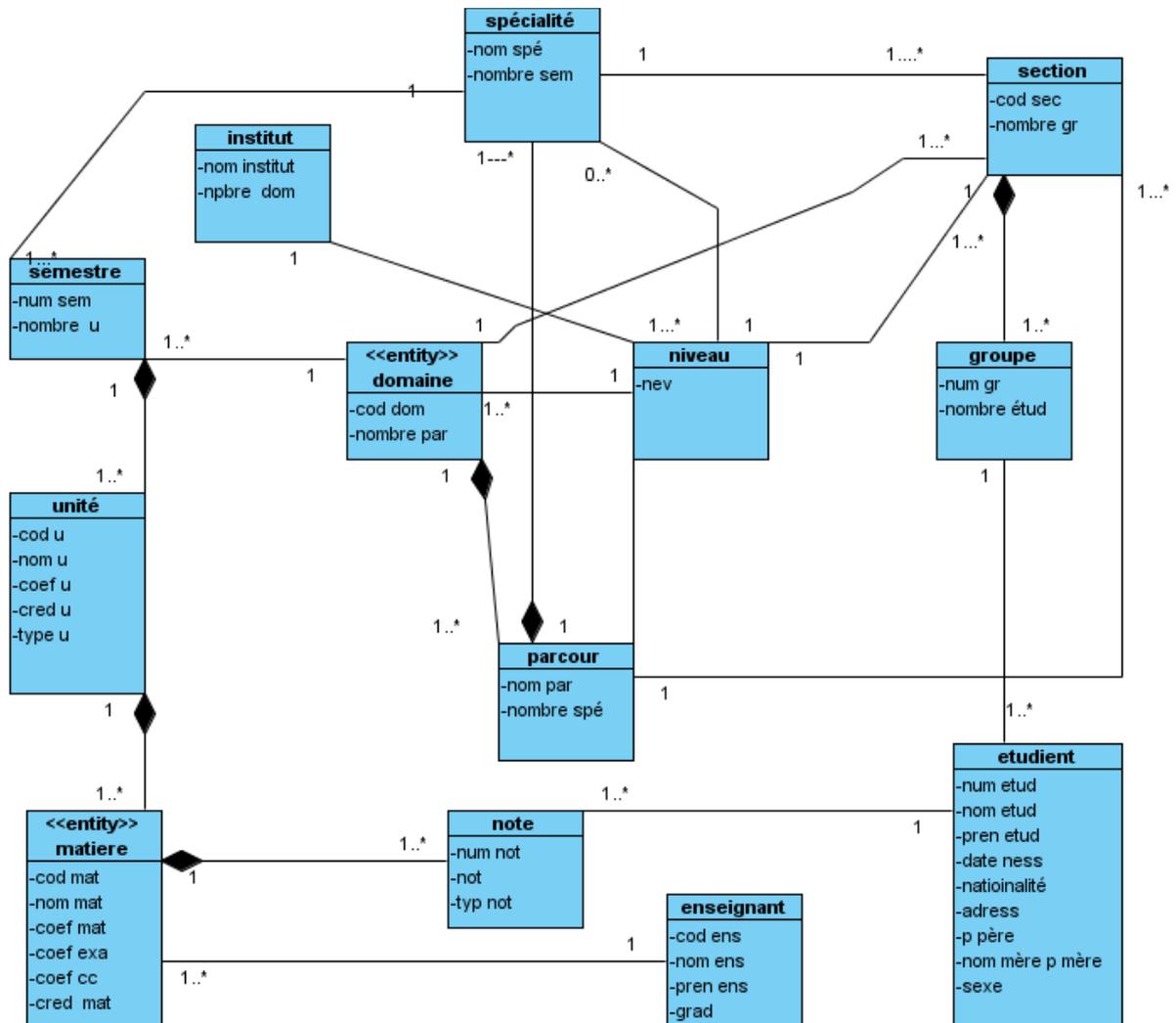


Figure 3.1 : Le modèle de domaine.

## 4. Diagrammes de classe participante :

### 4.1. Définition :

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre les classes des cas d'utilisation, et le modèle du domaine.

Il n'est pas souhaitable que les utilisateurs interagissent directement avec les instances des classes des domaines par le biais de l'interface graphique. En effet, le modèle du domaine doit être indépendant des utilisateurs et de l'interface graphique. De même, l'interface graphique du logiciel doit pouvoir évaluer sans répercussion sur le cœur de l'application [9].

4.2.Gestionner l'authentification :

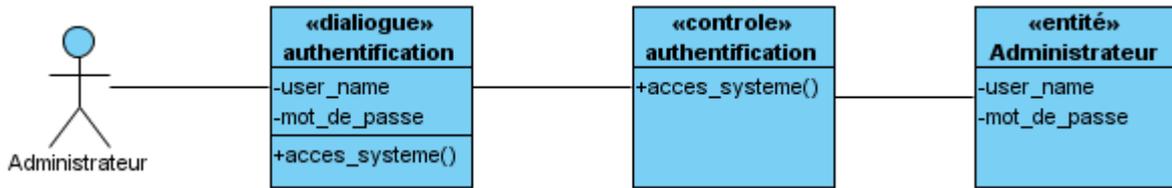


Figure 3 .2 : Diagramme de classes participantes : Gestionner d'authentification

4.3. Gestionner l'année d'étude:

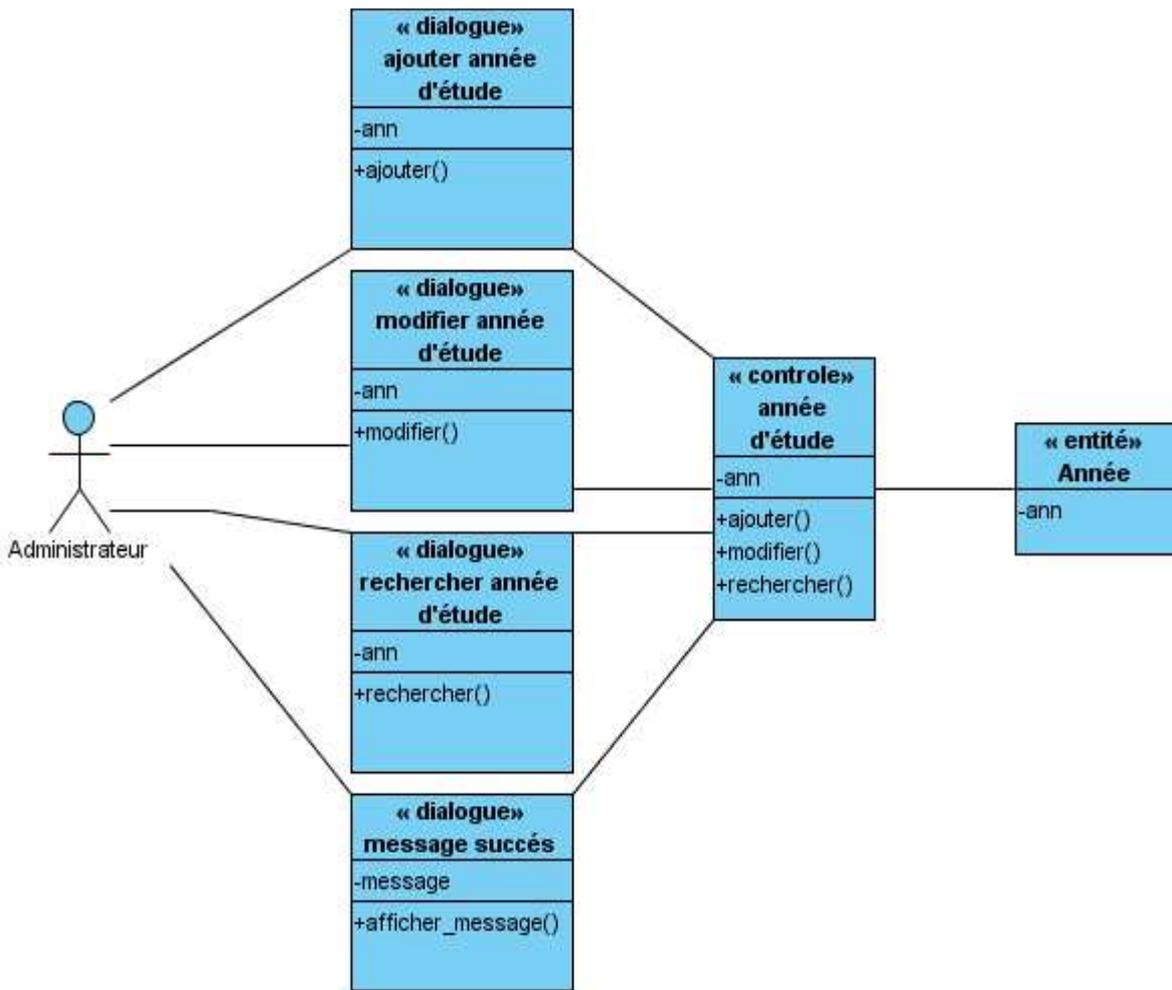


Figure 3 .3 : Diagramme de classes participantes : Gestionner l'année d'étude.

4.4. Initialiser les instituts :

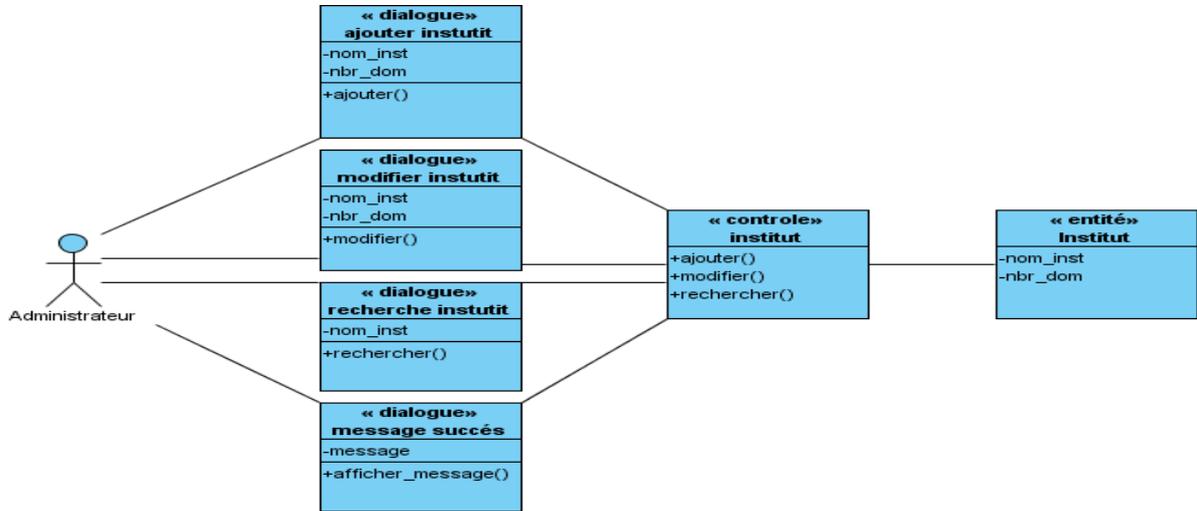


Figure 3.4 : Diagramme de classes participantes : Initialiser les instituts.

4.5. Saisir les notes :

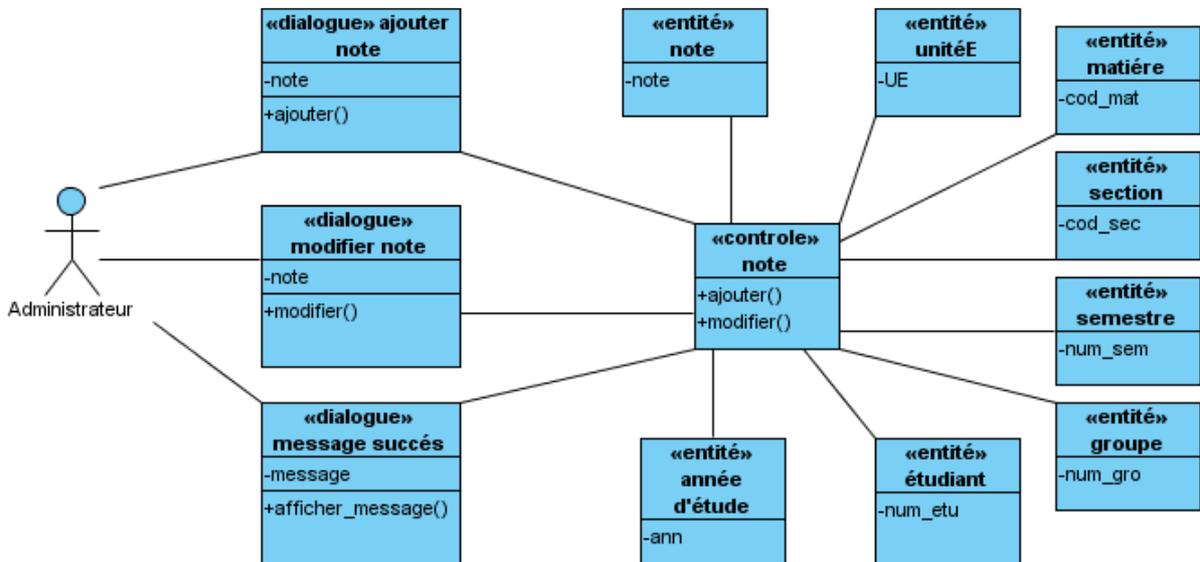


Figure 3.5 : Diagramme de classes participantes : saisir les notes.

4.6. Initialiser les étudiants :

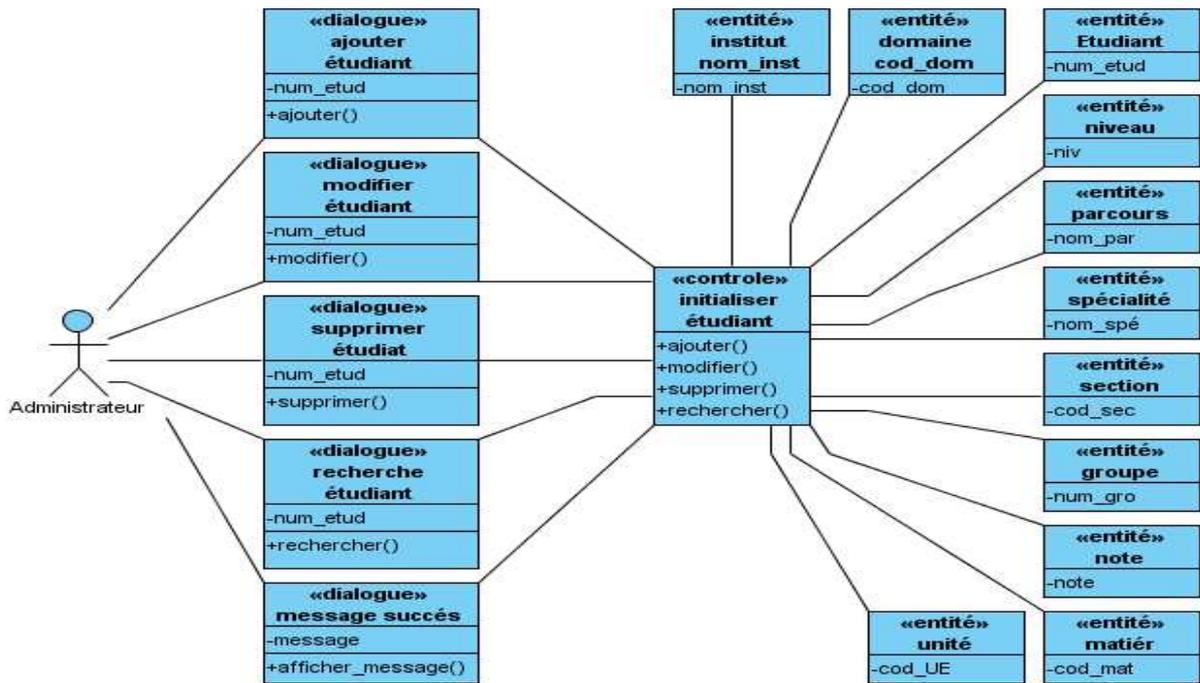


Figure 3 .6: Diagramme de classes participantes : initialiser les étudiants

4.7. Calculer les moyennes:

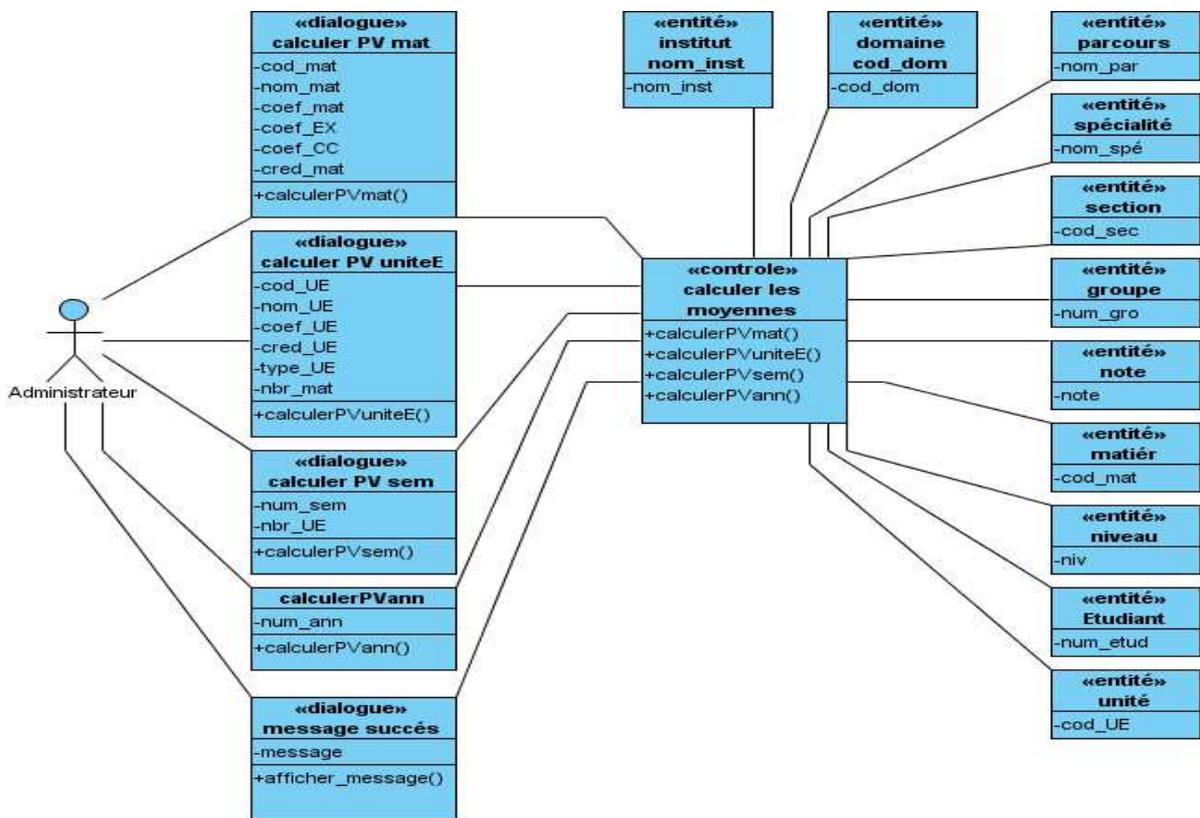
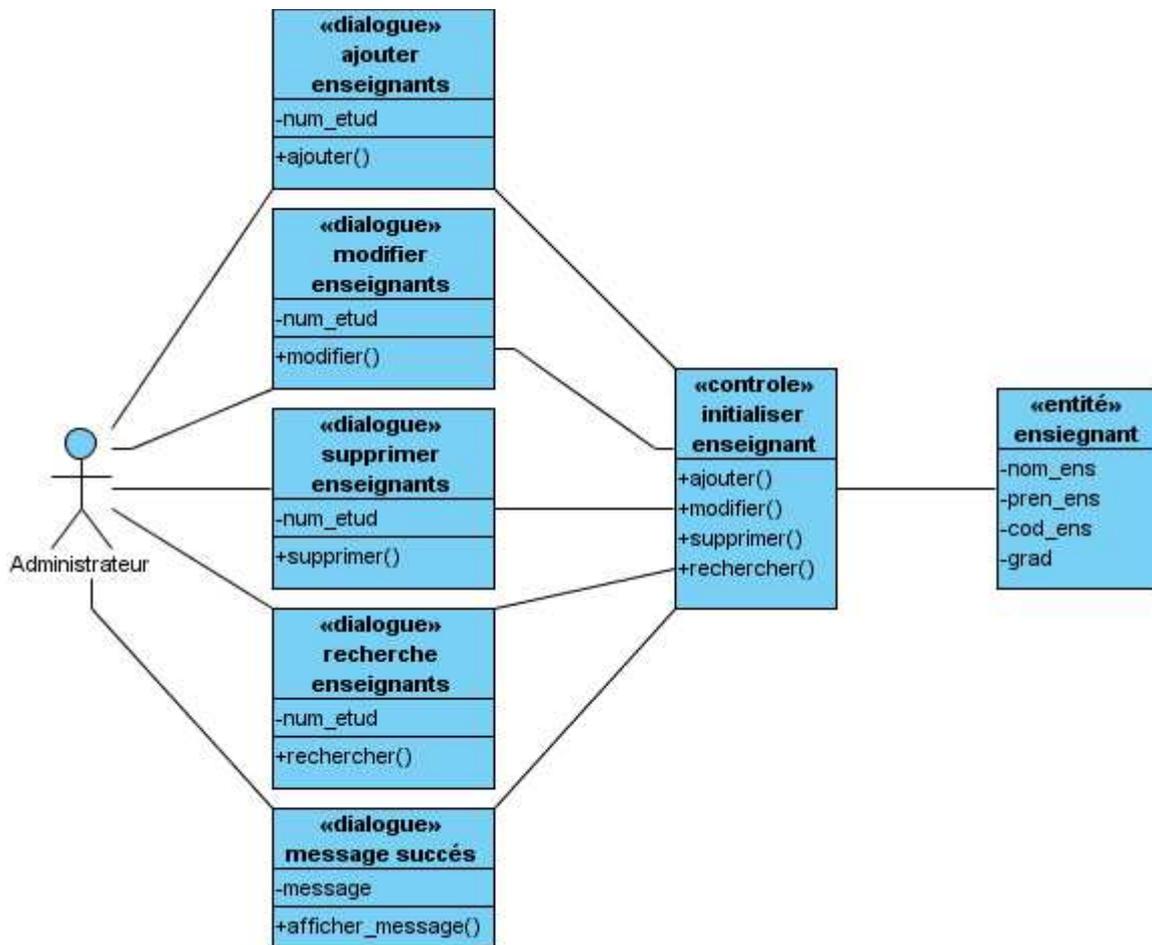


Figure 3 .7 : Diagramme de classes participantes : calculer les moyennes.

#### 4.8. Initialiser l'enseignant:



**Figure 3.8 :** Diagramme de classes participantes : l'enseignant.

#### Conclusion :

La phase d'analyse nous a permis de modéliser les concepts de domaine en:

- Représentant les différentes entités de domaine.
- Représentant les relations entre les classes d'entités, les classes de contrôles et les classes de dialogues.

Ensuite cette phase prépare la phase de conception qui est l'objet de chapitre suivant.

# Chapitre

## Phase de conception

*-Introduction.*

*1-Diagramme d'interaction.*

*2-les Diagramme d'interaction.*

*3-Diagramme de classes de  
conception.*

*-Conclusion.*



## Introduction

Dans ce chapitre nous allons représenter la vue dynamique du système, qui permet d'élaborer les interactions entre les objets, ainsi que les messages échangés à travers les diagrammes d'interaction (diagramme de séquence).

Le diagramme d'interaction met l'accent sur l'organisation structurelle des objets qui envoient et reçoivent des messages. Le diagramme de séquence est un diagramme d'interaction mettant l'accent sur la chronologie de l'envoi des messages. Les diagrammes d'interaction permettent d'établir un lien entre les diagrammes de cas d'utilisation et les diagrammes de classes : ils montrent comment des objets (i.e. des instances de classes) communiquent pour réaliser une certaine fonctionnalité. Ils apportent ainsi un aspect dynamique à la modélisation du système.

Pour produire un diagramme d'interaction, il faut focaliser l'attention sur un sous-ensemble d'éléments du système et étudier leur façon d'interagir pour décrire un comportement particulier [6].

### 1. Diagramme d'interaction :

#### 1.1. Définition :

Le diagramme d représente des échanges de messages entre les objets. Il permet de mieux visualiser la séquence des messages par une lecture de haut en bas.

Le diagramme de séquence permet de représenter un processus de façon simplifiée, en se centrant sur le Workflow et les échanges entre acteurs ou avec le système d'information, voir le système informatique. On pourra donc l'utiliser soit pour représenter un processus existant, soit pour modéliser des variantes de processus à partir d'un processus de référence [8].

### 2. Les Diagrammes d'interaction :

2.1.Gérer l'authentification :

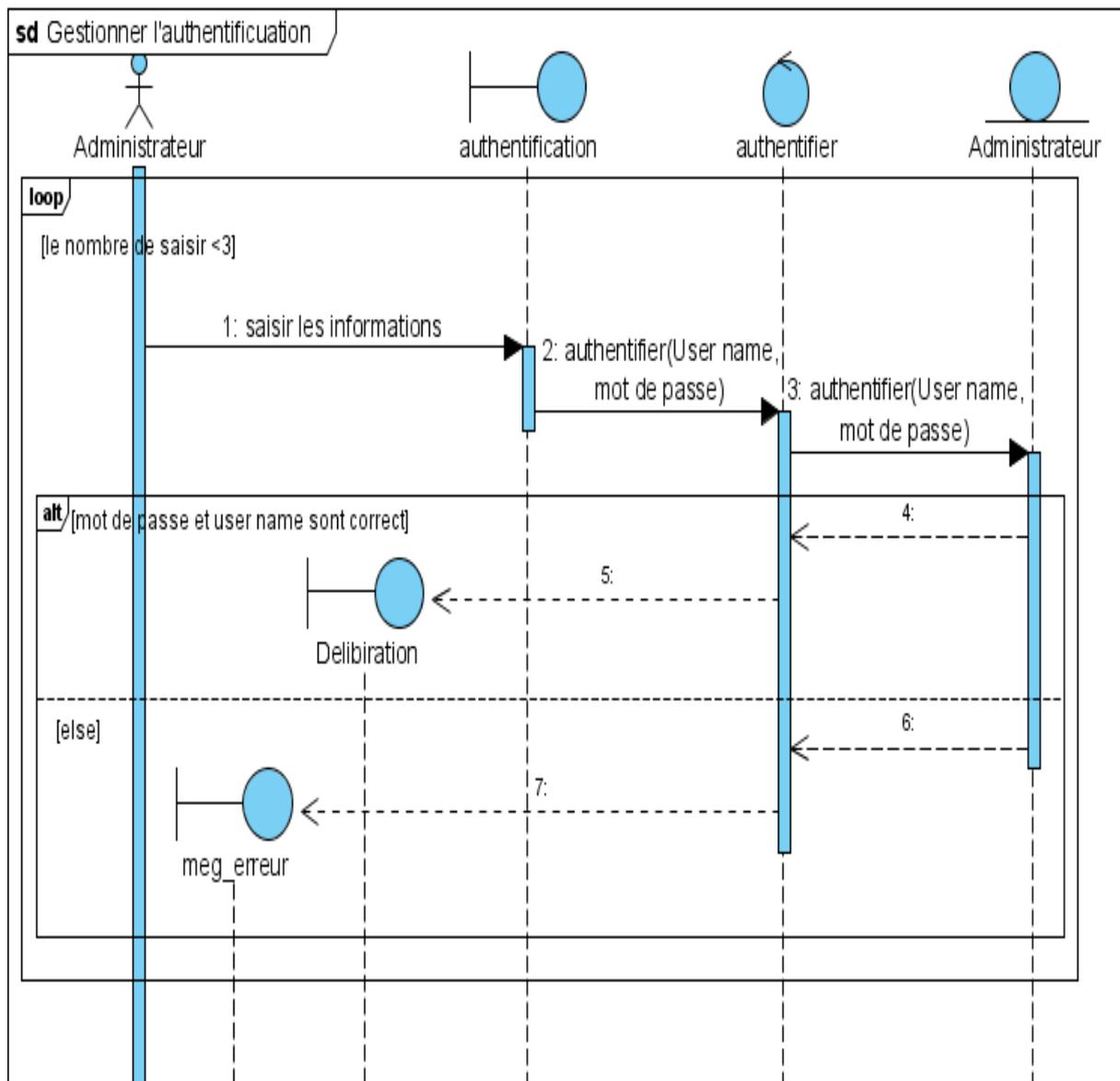


Figure 4 .1 : Diagramme d'interaction : gérer l'authentification.

2.2. Initialiser les étudiants :

2.2.1. Cas d'ajouter :

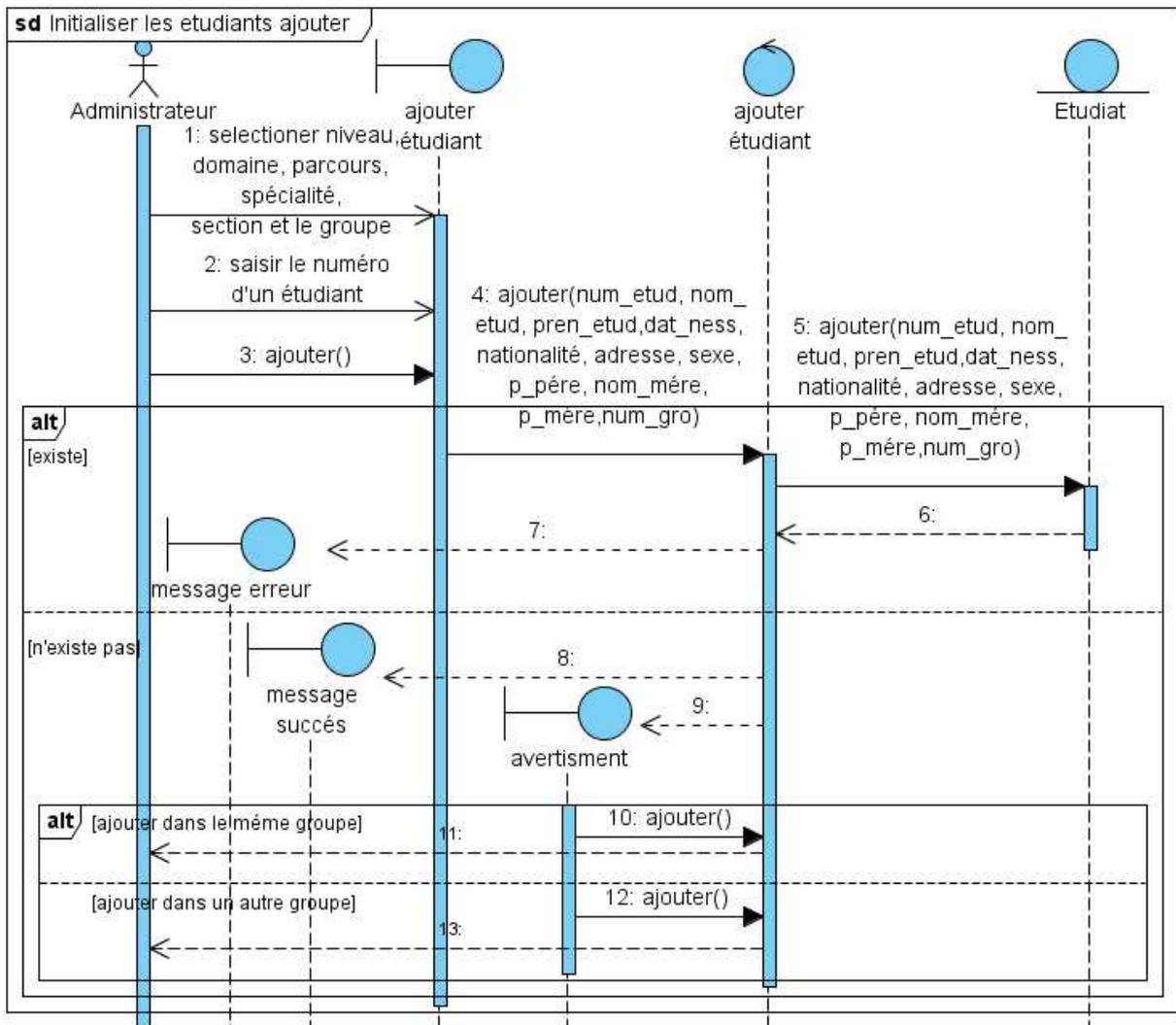


Figure 4.2 : Diagramme d'interaction : initialiser l'étudiant.

<Cas d'ajouter>

2.2.2. Cas de modification:

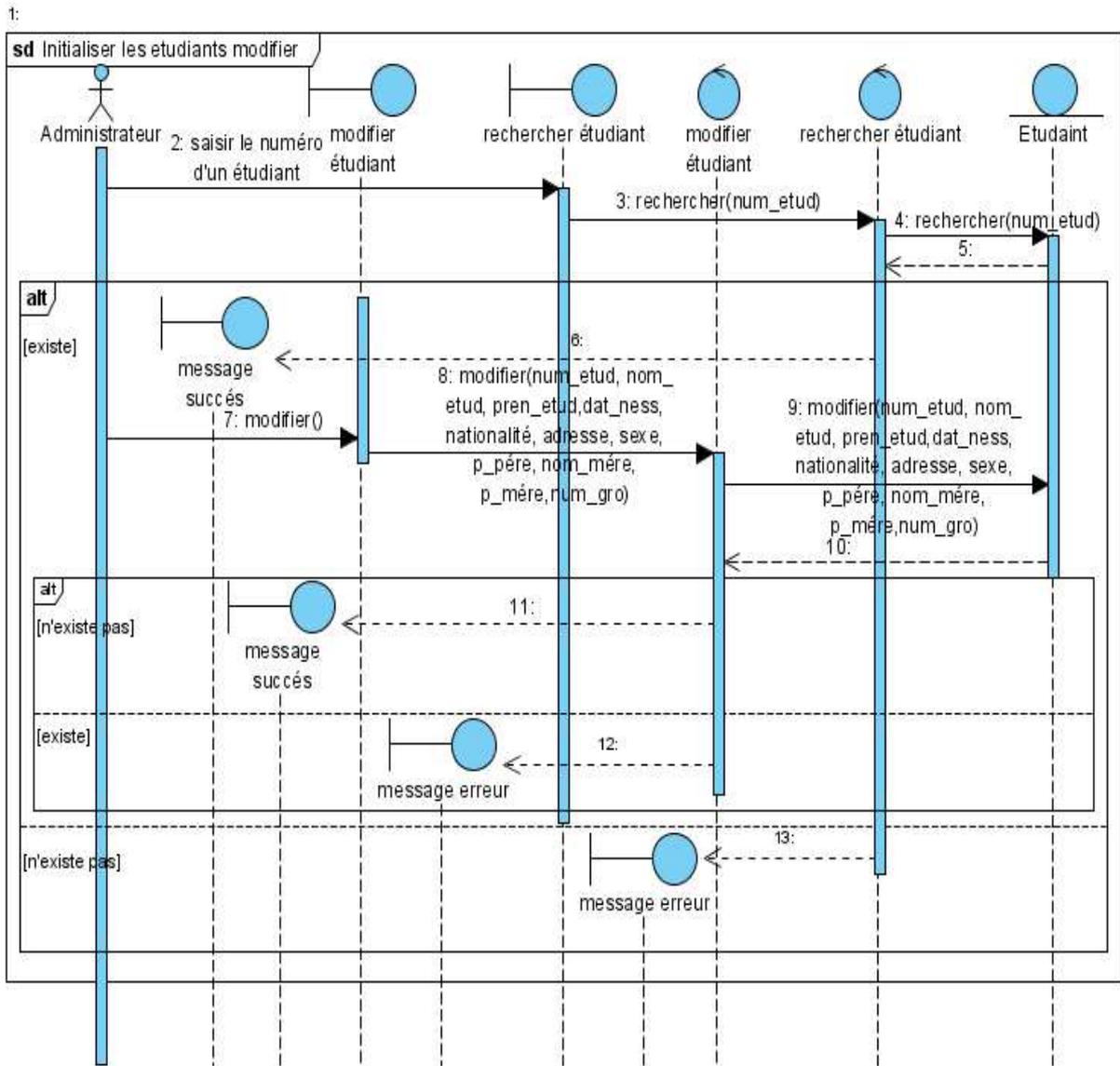


Figure 4.3 : Diagramme d'interaction : initialiser l'étudiant.

<Cas de modification>

2.2.3. Cas de suppression :

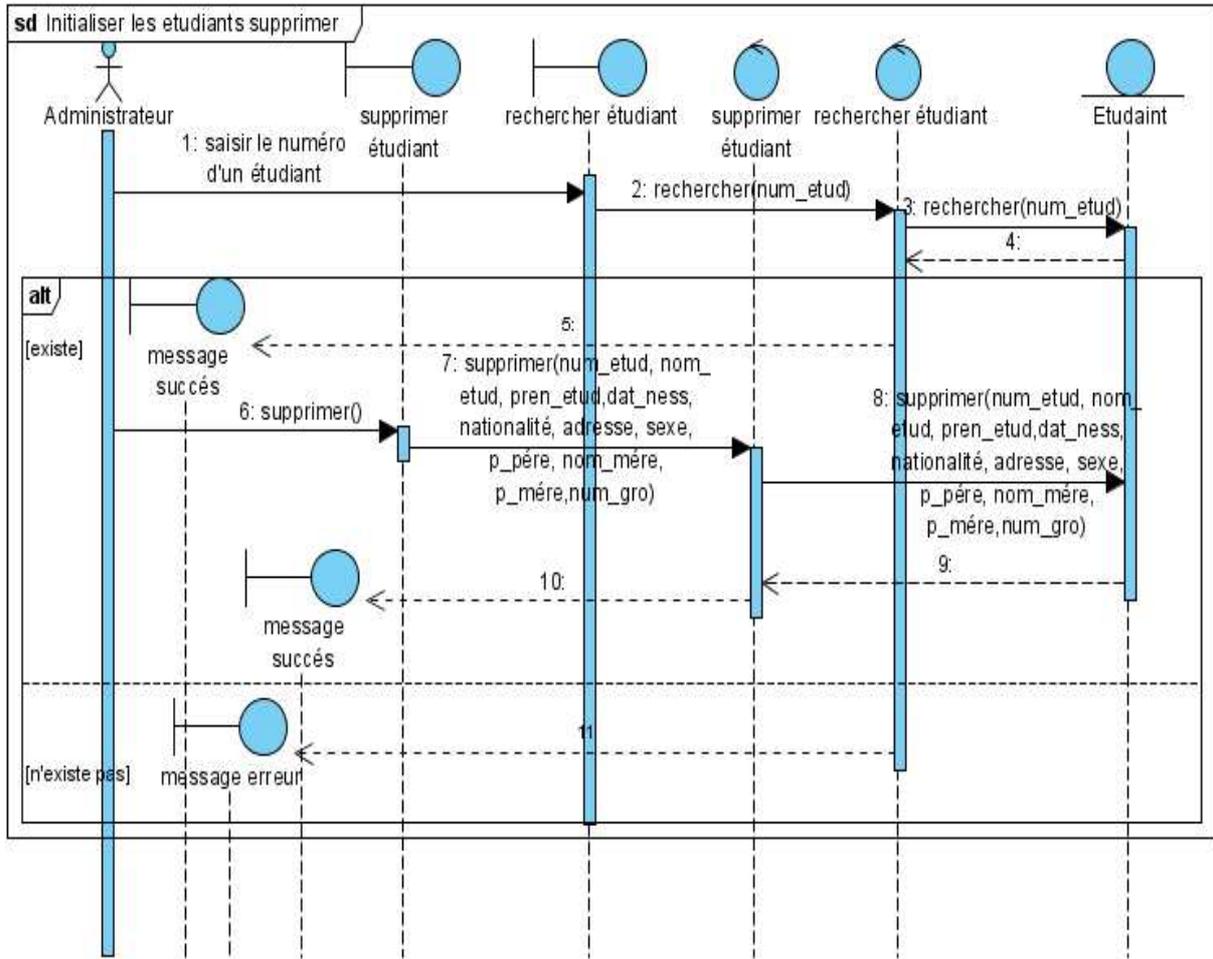


Figure 4.4 : Diagramme d'interaction : initialiser l'étudiant.

<Cas suppression>

2.3. Initialiser les enseignants :

2.3.1. Cas d'ajouter :

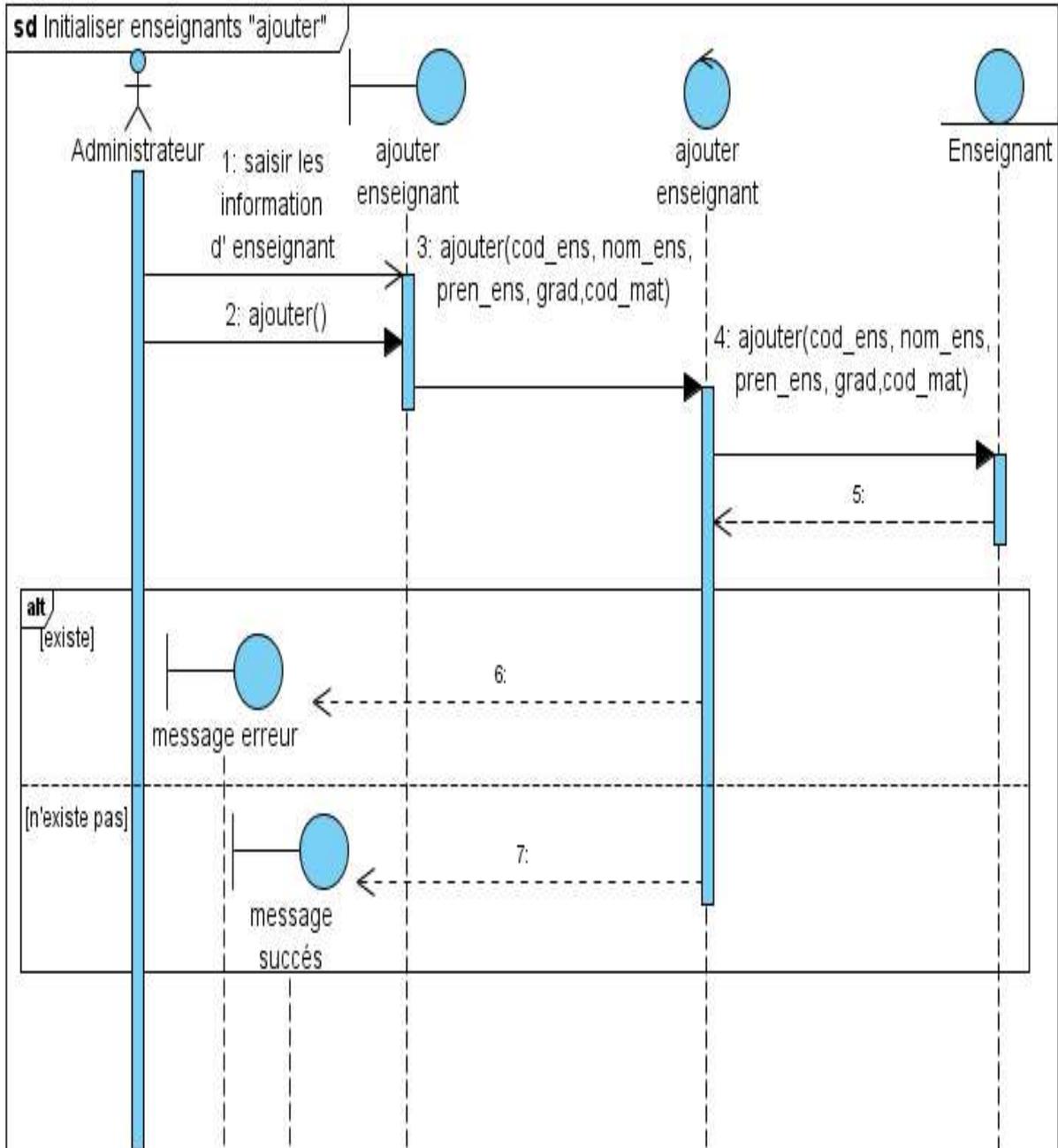


Figure 4.5 : Diagramme d'interaction : initialiser l'enseignant.

< Cas d'ajouter >

2.3.2. Cas de modification :

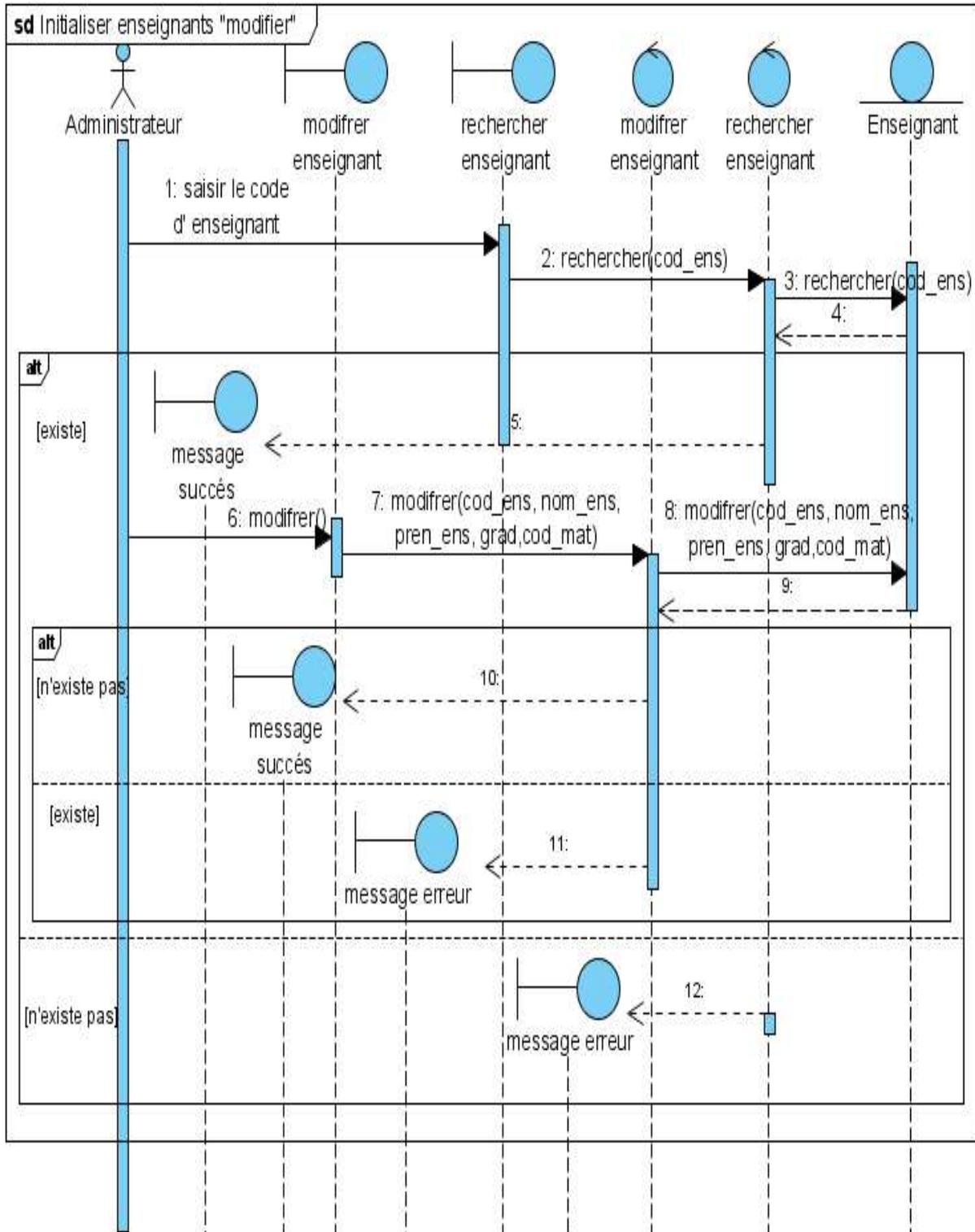


Figure 4.6 : Diagramme d'interaction : initialiser l'enseignant.

<Cas de modification>

2.3.3. Cas de suppression :

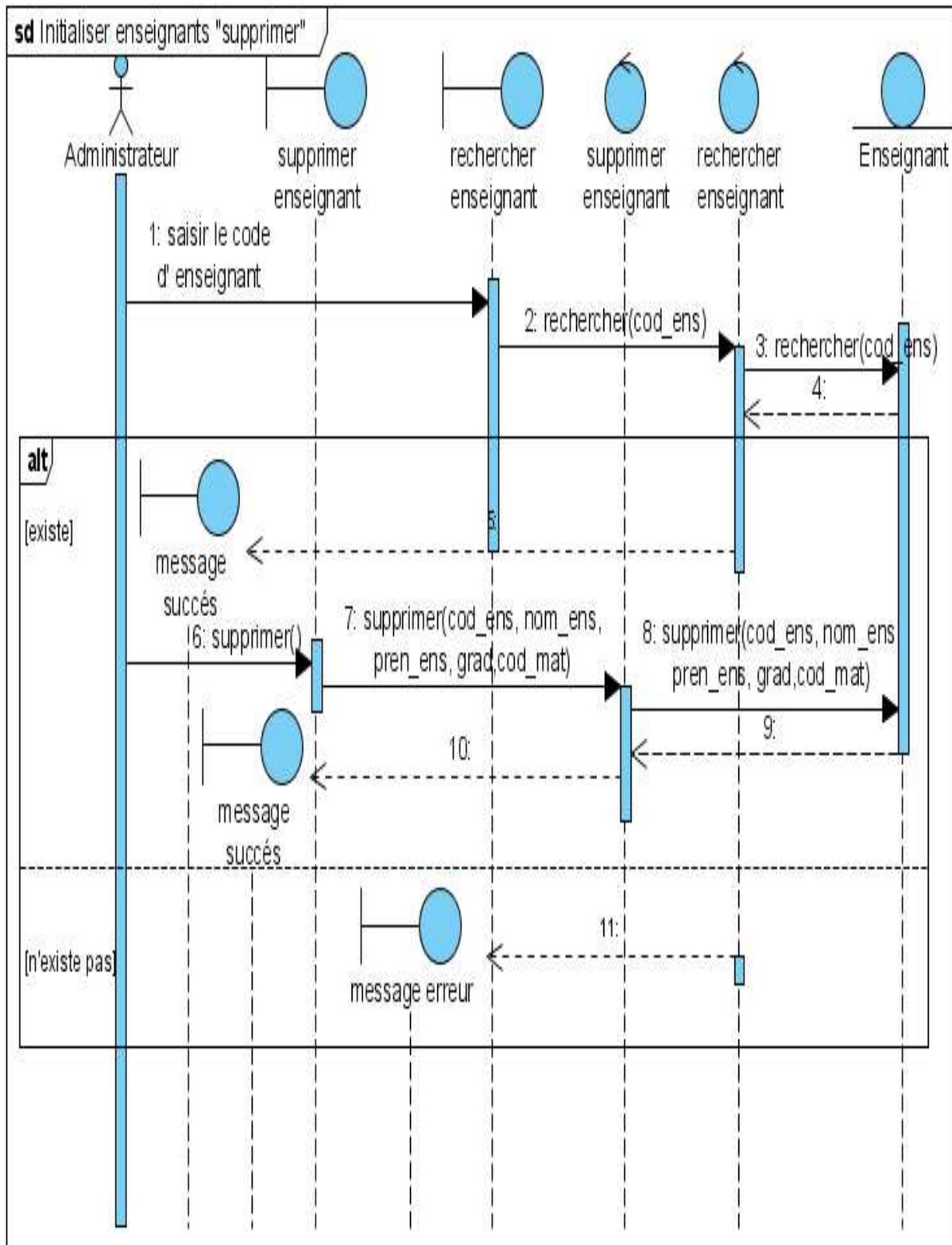


Figure 4.7 : Diagramme d'interaction : initialiser l'enseignant

<Cas de suppression>

2.4. Initialiser l'institut :

2.4.1. Cas d'ajouter :

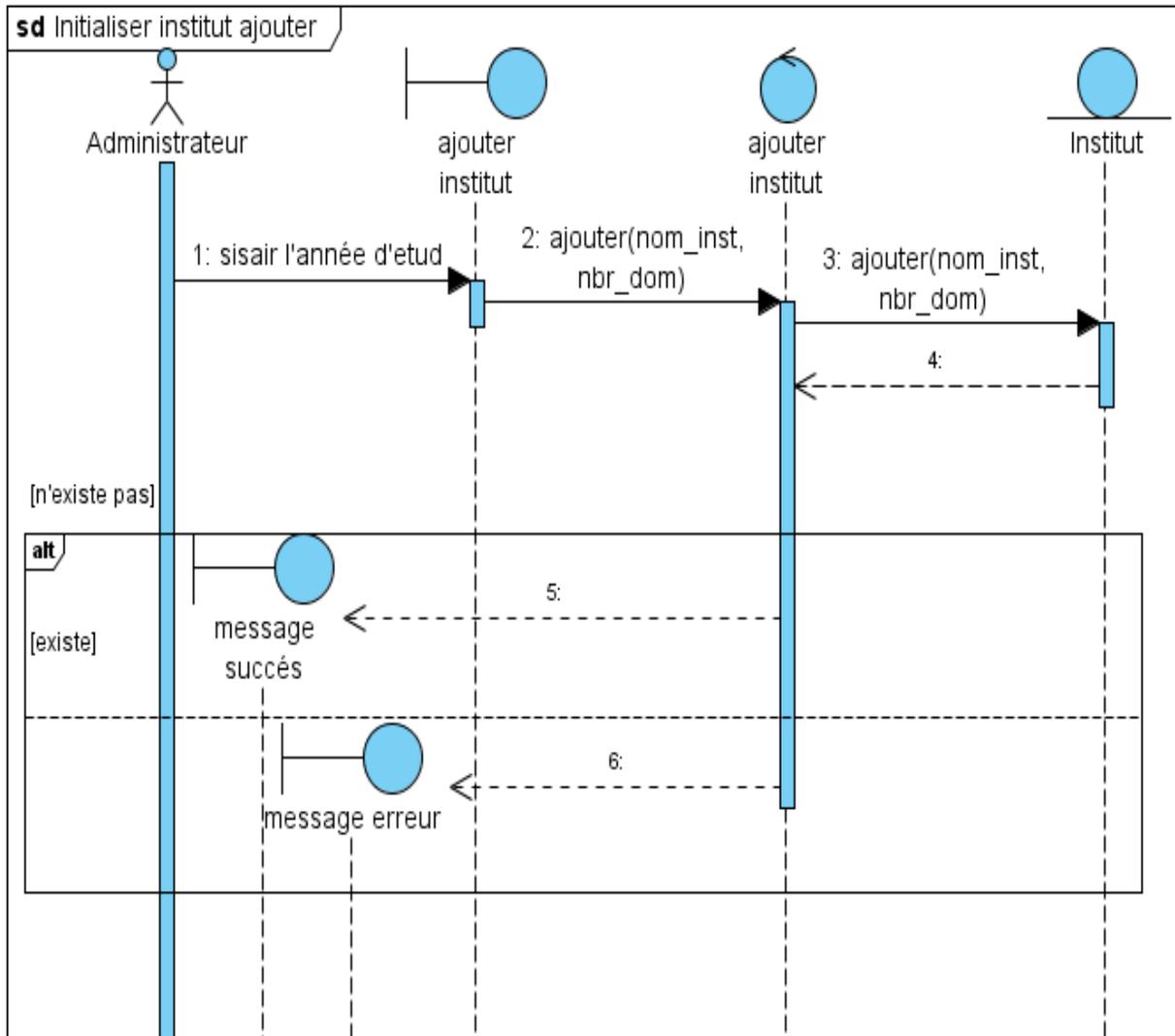


Figure 4.8 : Diagramme d'interaction : initialiser l'institut.

<Cas d'ajouter>

2.4.2. Cas de modification:

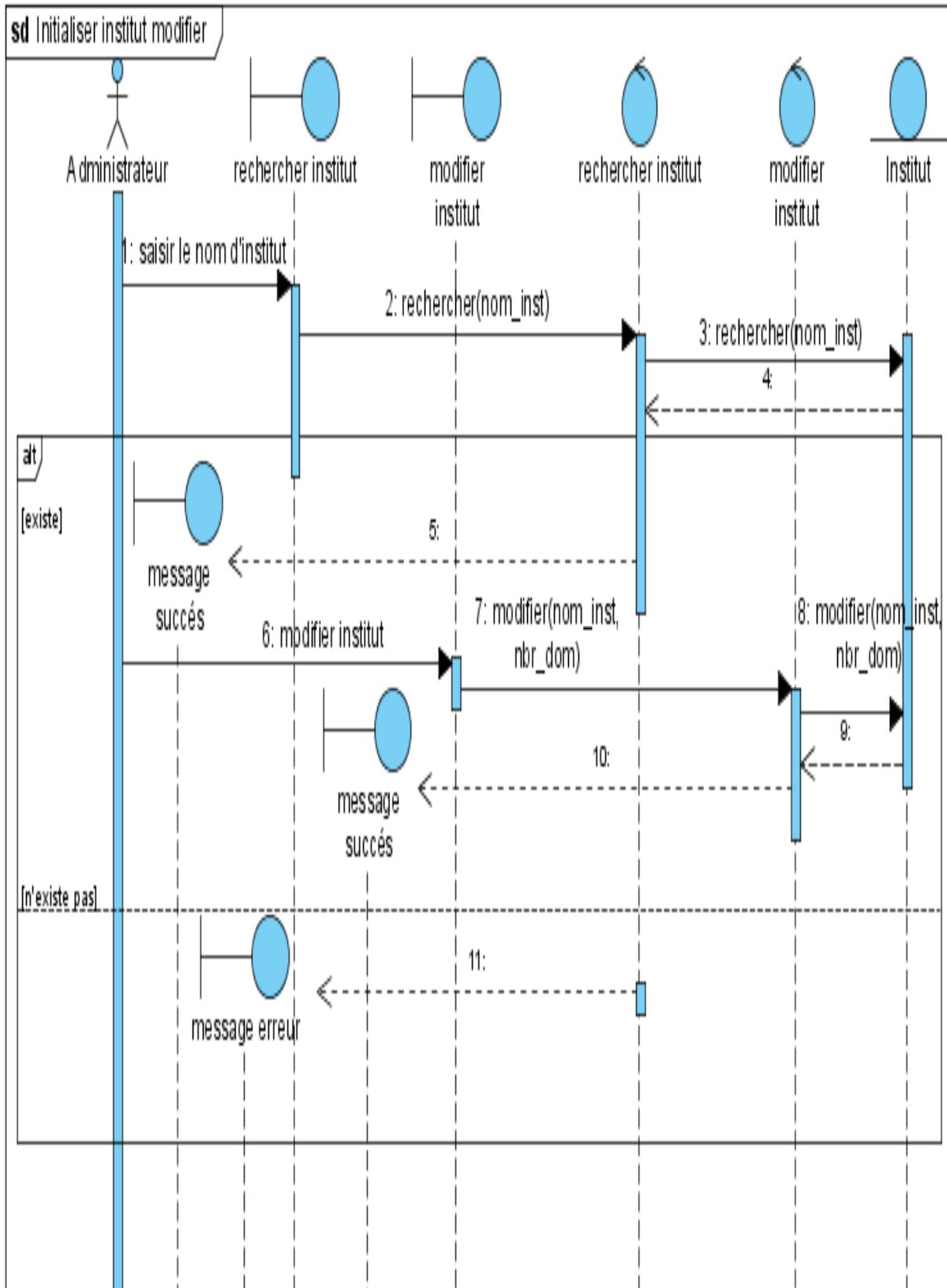


Figure 4.9 : Diagramme de interaction : initialiser l'institut.<cas de modification>

2.5. Saisir les notes :

2.5.1. Cas d'ajouter :

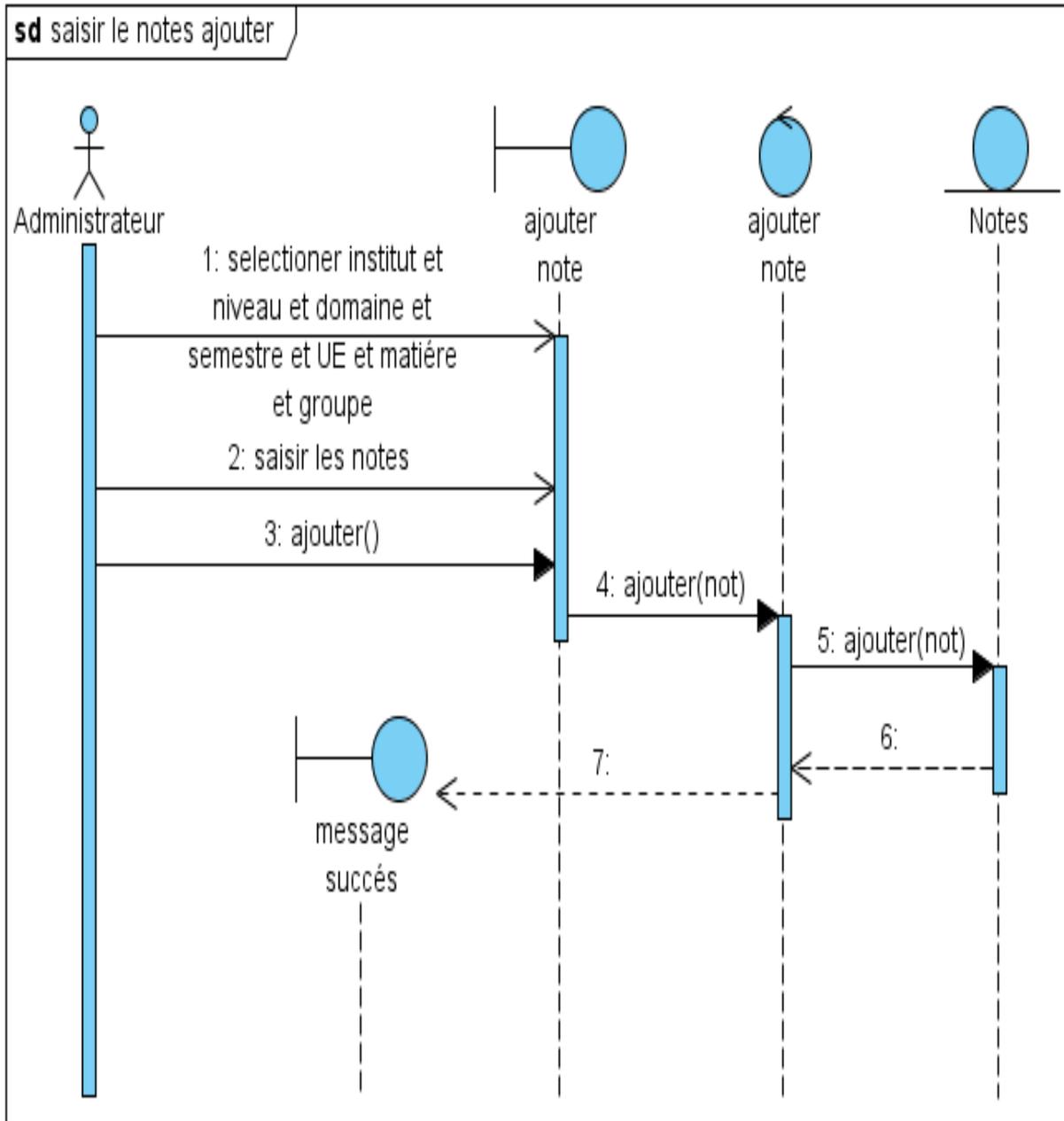


Figure 4 .10: Diagramme d'interaction : saisir les notes.<cas d'ajouter>

2.5.2. Cas de modification :

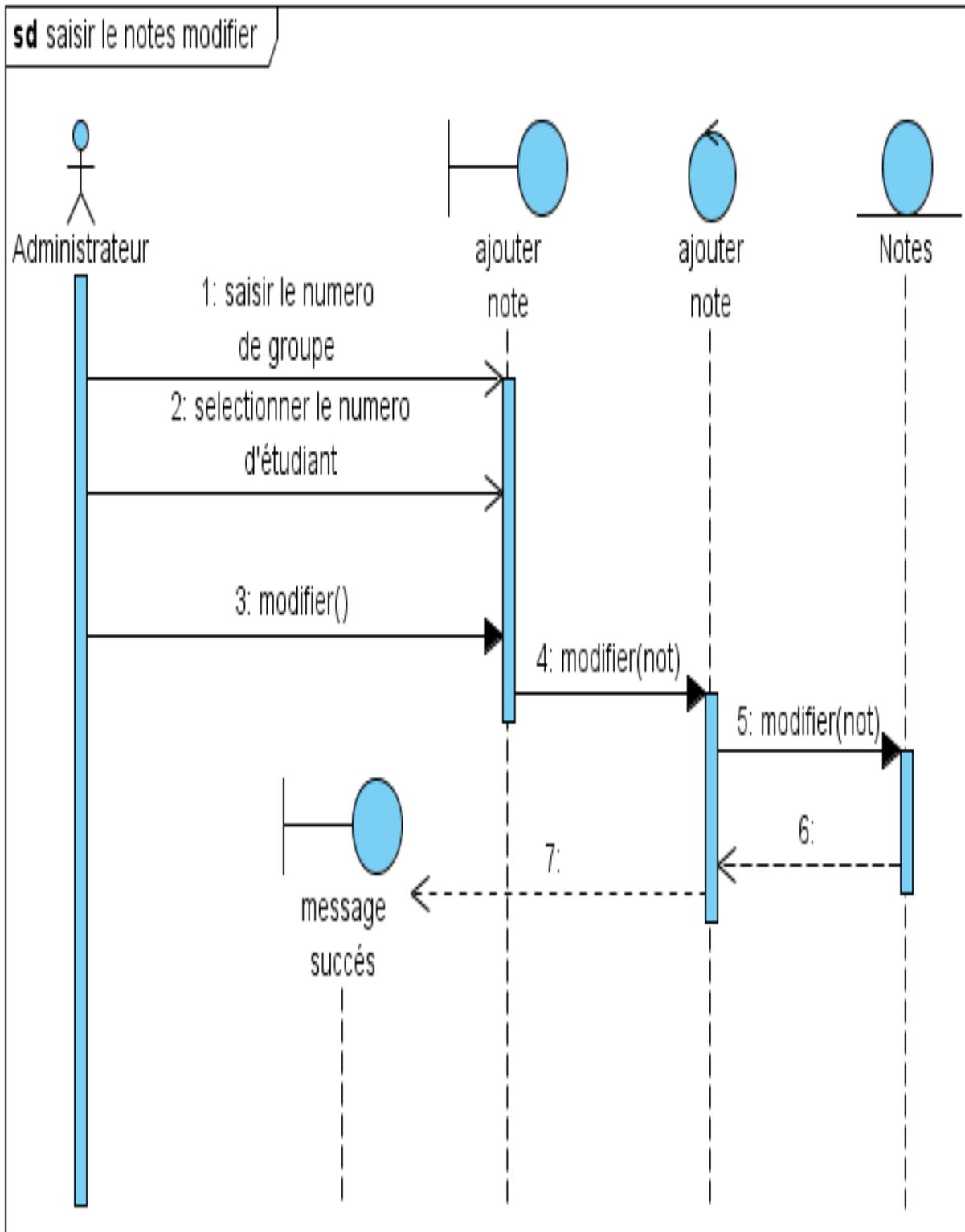


Figure 4 .11: Diagramme d'interaction : saisir les notes <cas de modification>

2.6. Calculer les moyennes :

2.6.1. Calculer PV matière :

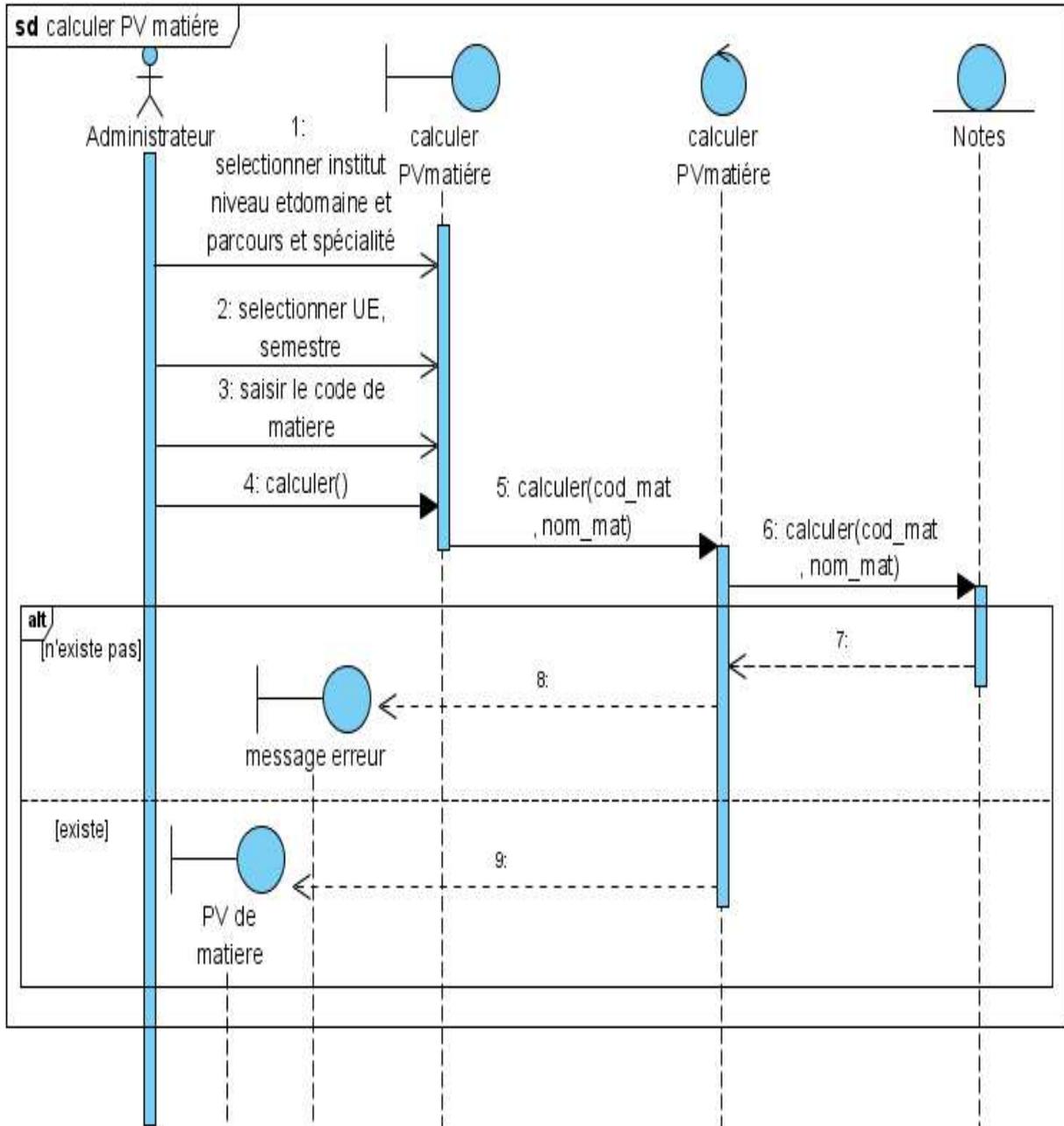


Figure 4 .12: Diagramme d'interaction : Calculer les moyennes.

<PV matière>

2.6.2. Calculer PV unité d'enseignant:

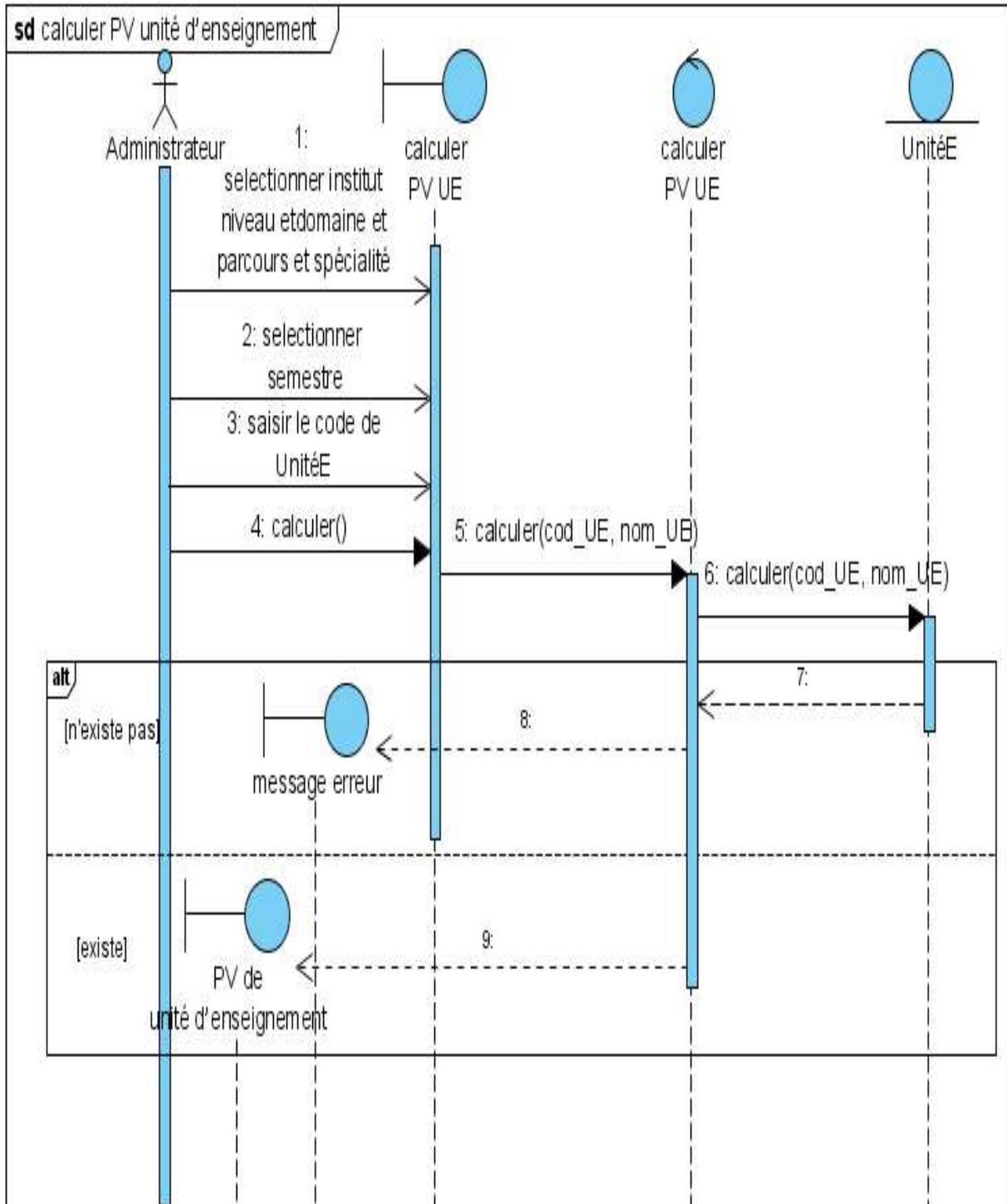


Figure 4.13 : Diagramme d'interaction : Calculer les moyennes.

<PV unité d'enseignant>

2.6.3. Calculer PV semestre :

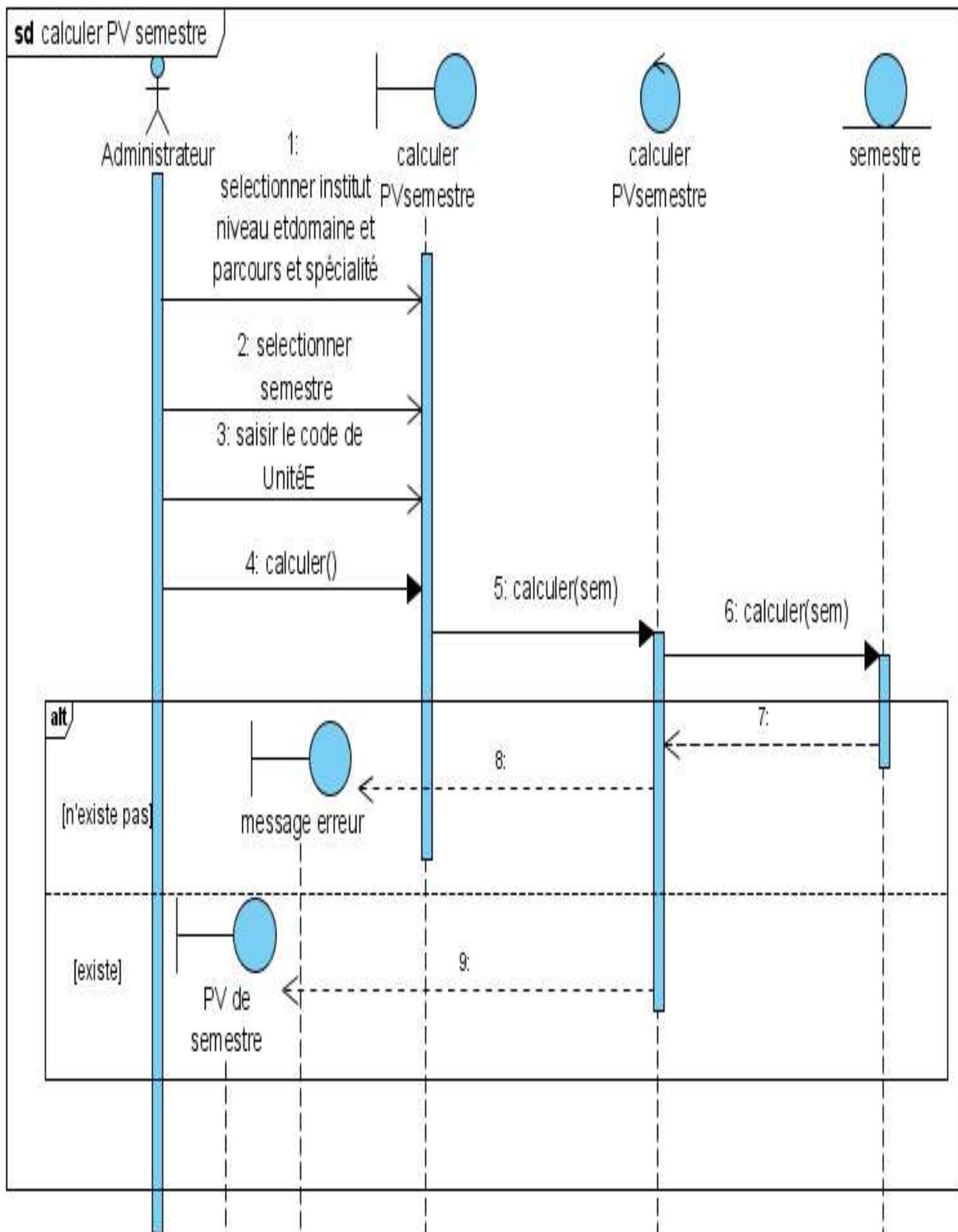


Figure 4 .14: Diagramme d'interaction : Calculer les moyennes.

<PV semestre>

2.6.3. Calculer PV annuelle :

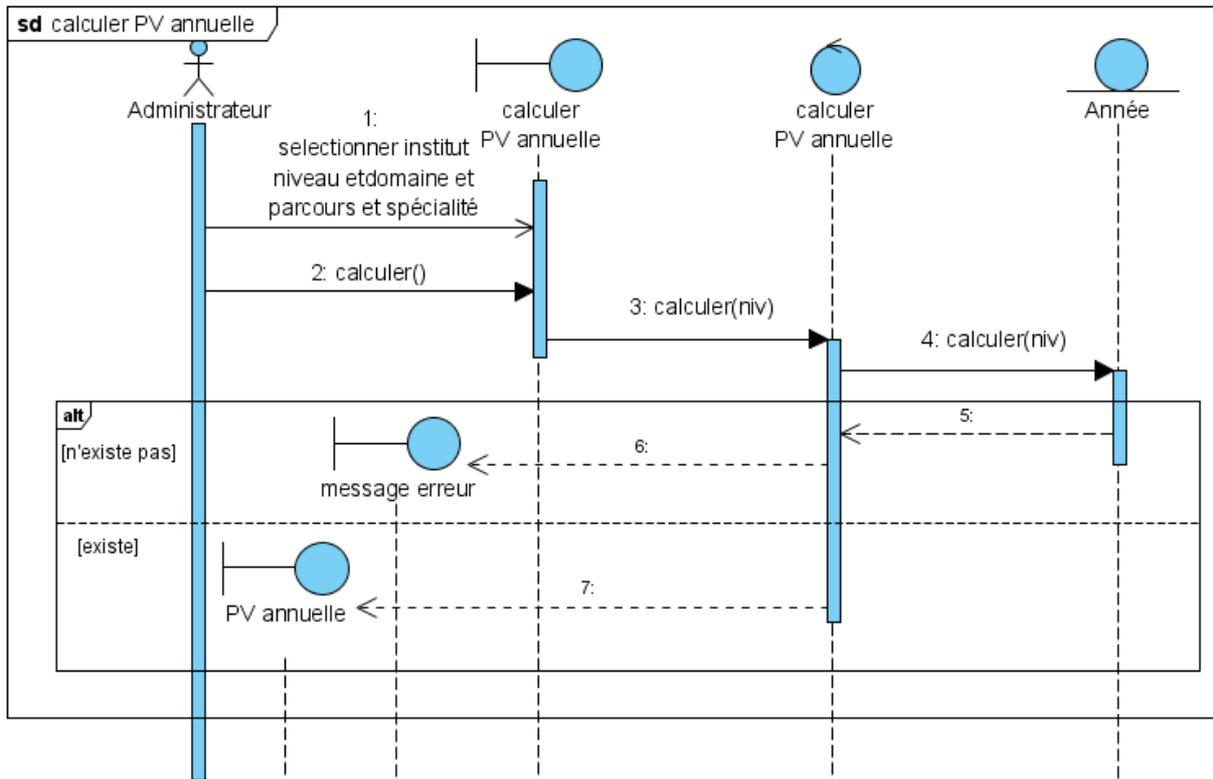


Figure 4.15 : Diagramme d'interaction : Calculer les moyennes.

<PV annuelle>

3. Diagramme de classes de conception :

3.1. gérer l'authentification :

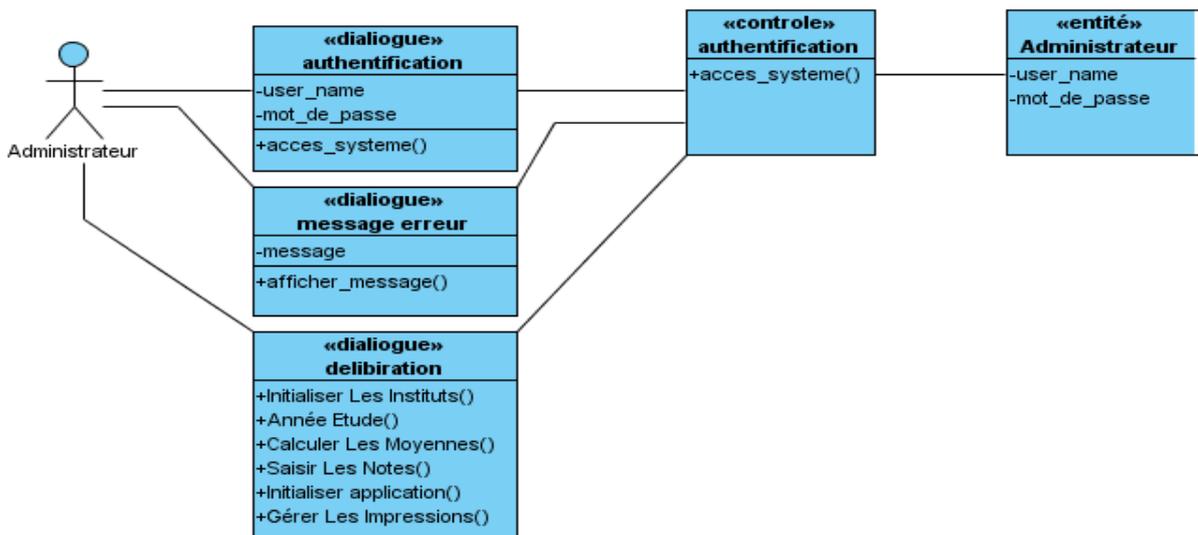


Figure 4.16: Diagramme de classes de conception : gérer d'authentification

### 3.2. Gérer l'année d'étude:

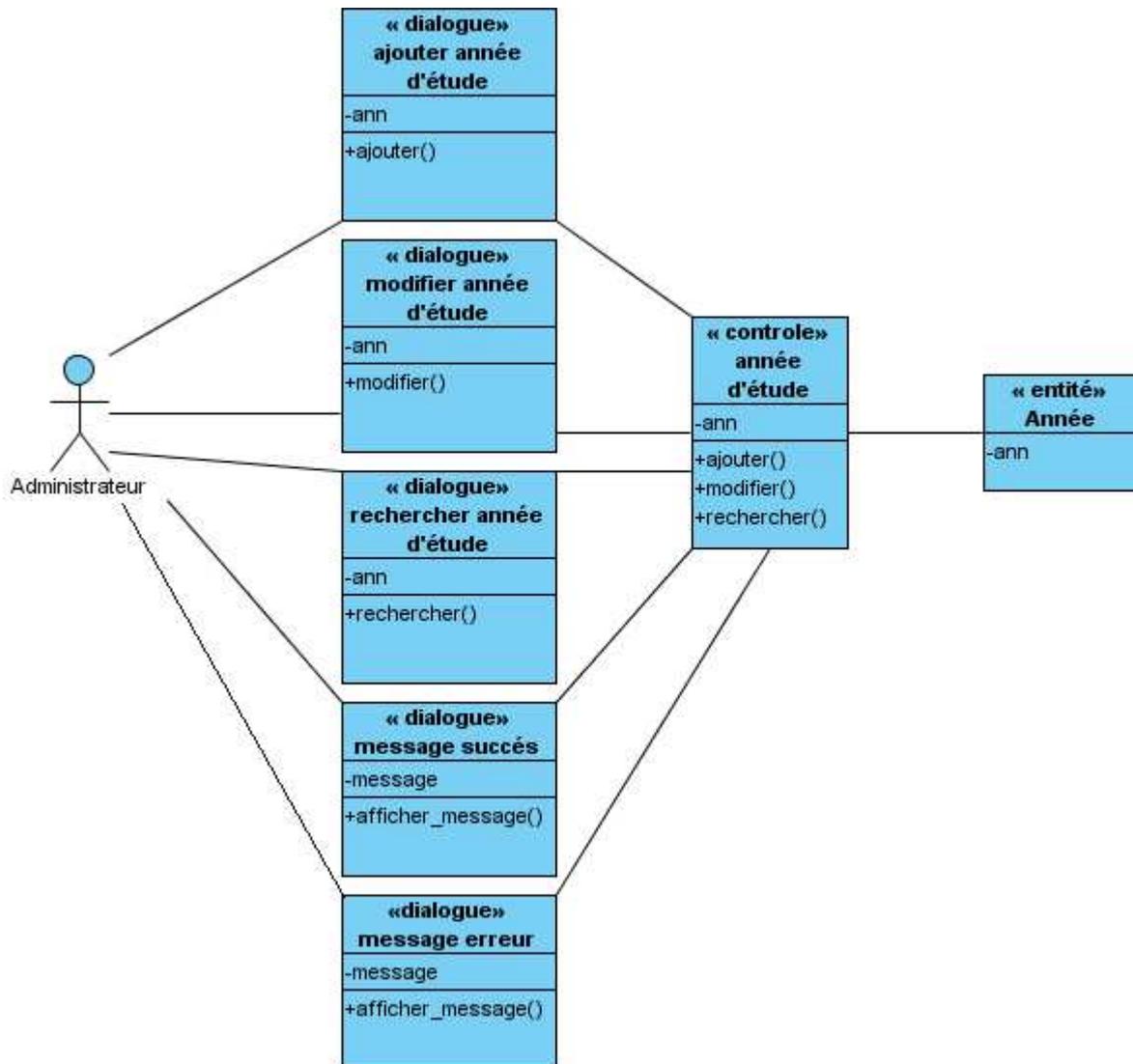


Figure 4.17 : Diagramme de classes de conception : gérer l'année d'étude

## 3.3. Initialiser les instituts :

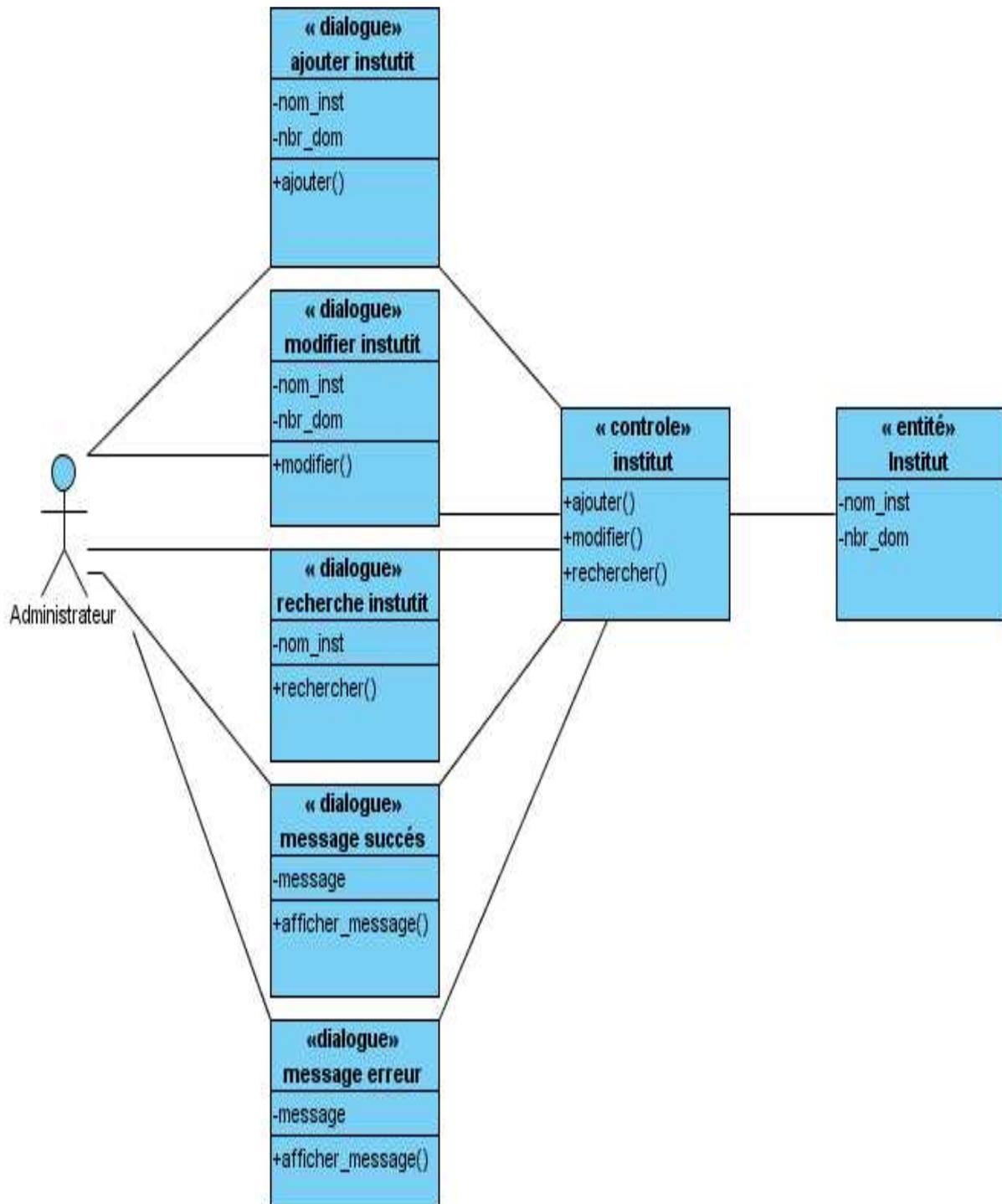


Figure 4.18 : Diagramme de classes de conception : Initialiser les instituts

3.4. Calculer les moyennes:

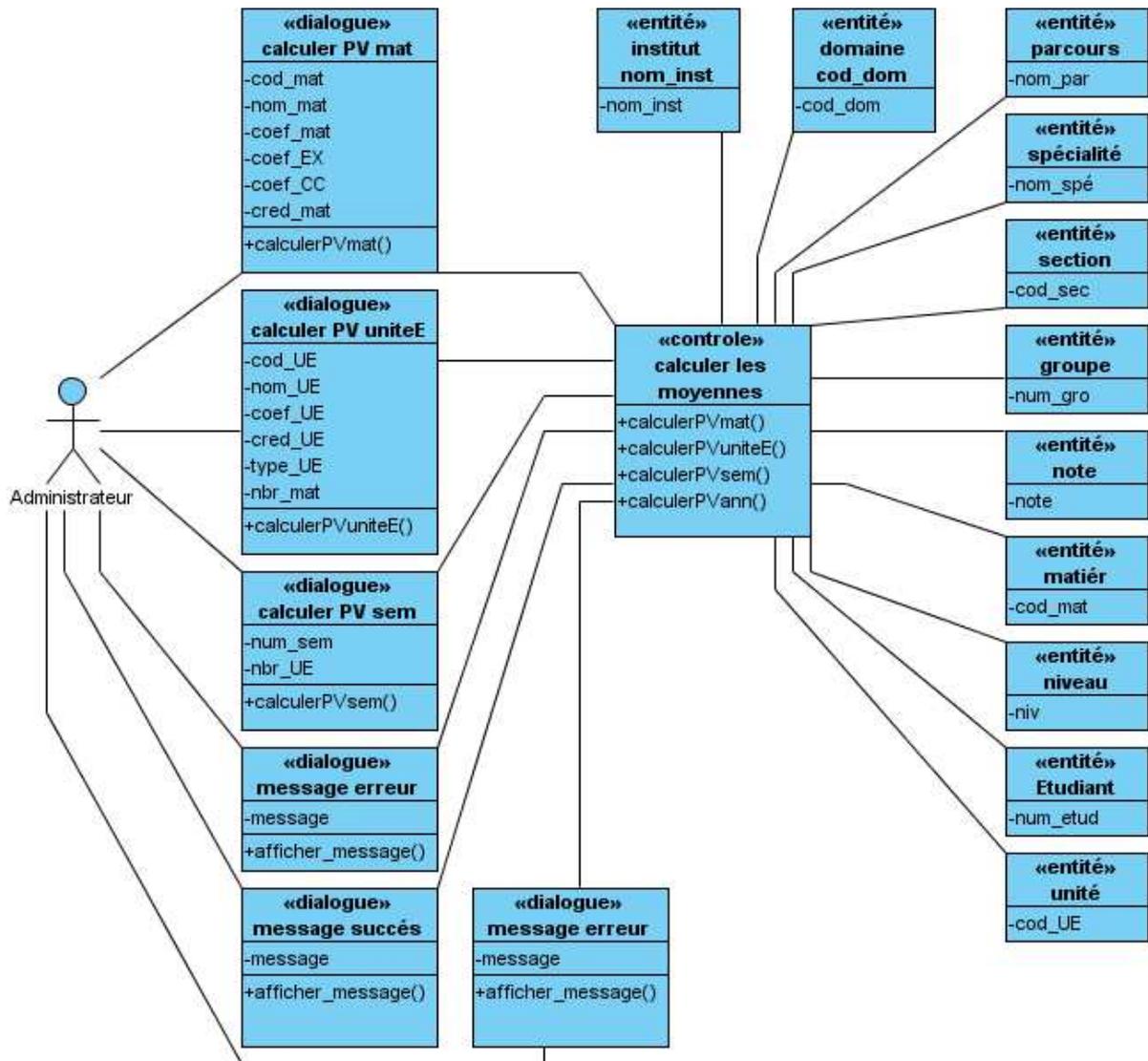


Figure 4.19 : Diagramme de classes de conception : calculer les moyennes.

3.5. Initialiser les étudiants :

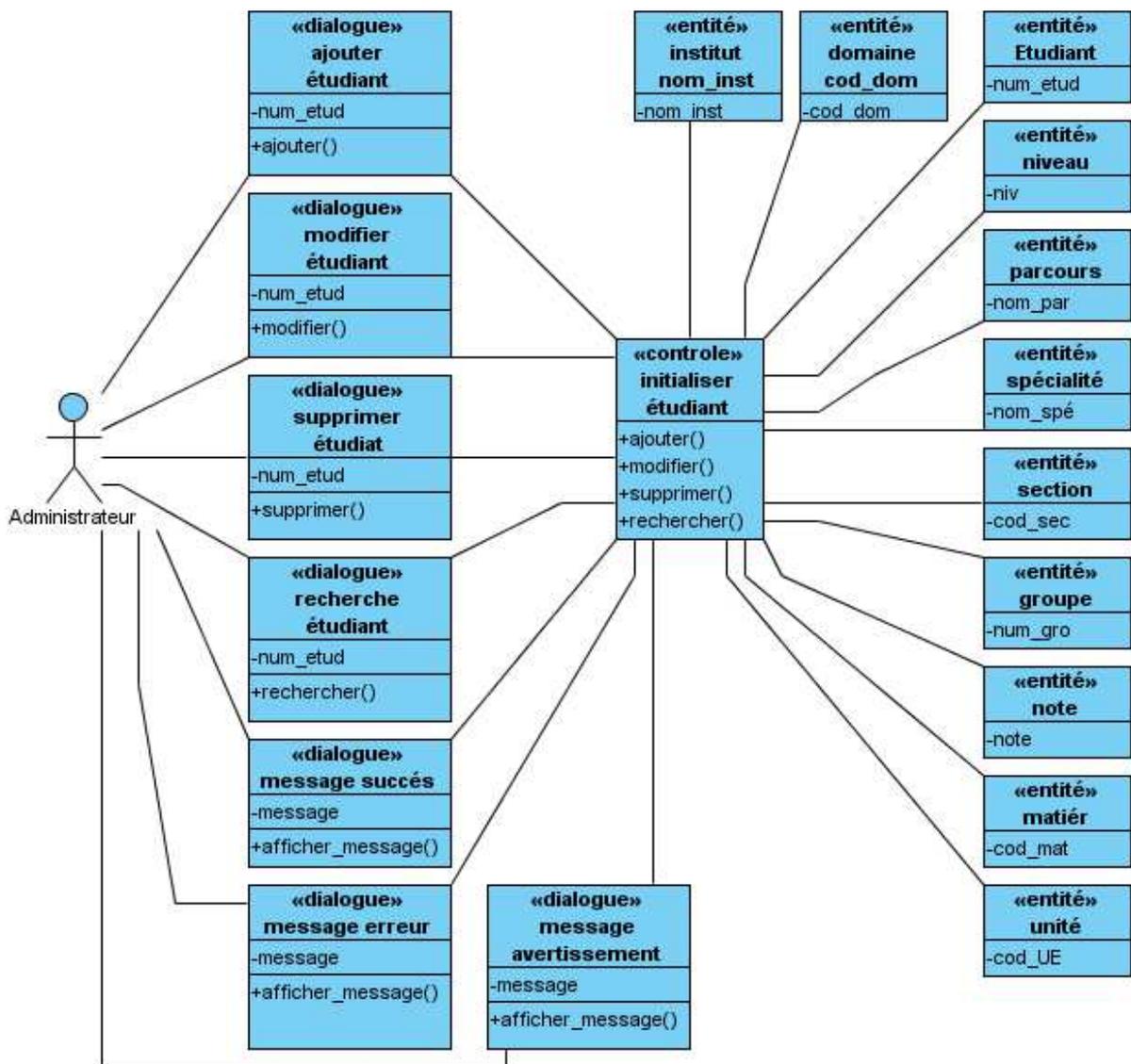


Figure 4.20: Diagramme de classes conception : initialiser les étudiants

### 3.6. Initialiser les enseignants :

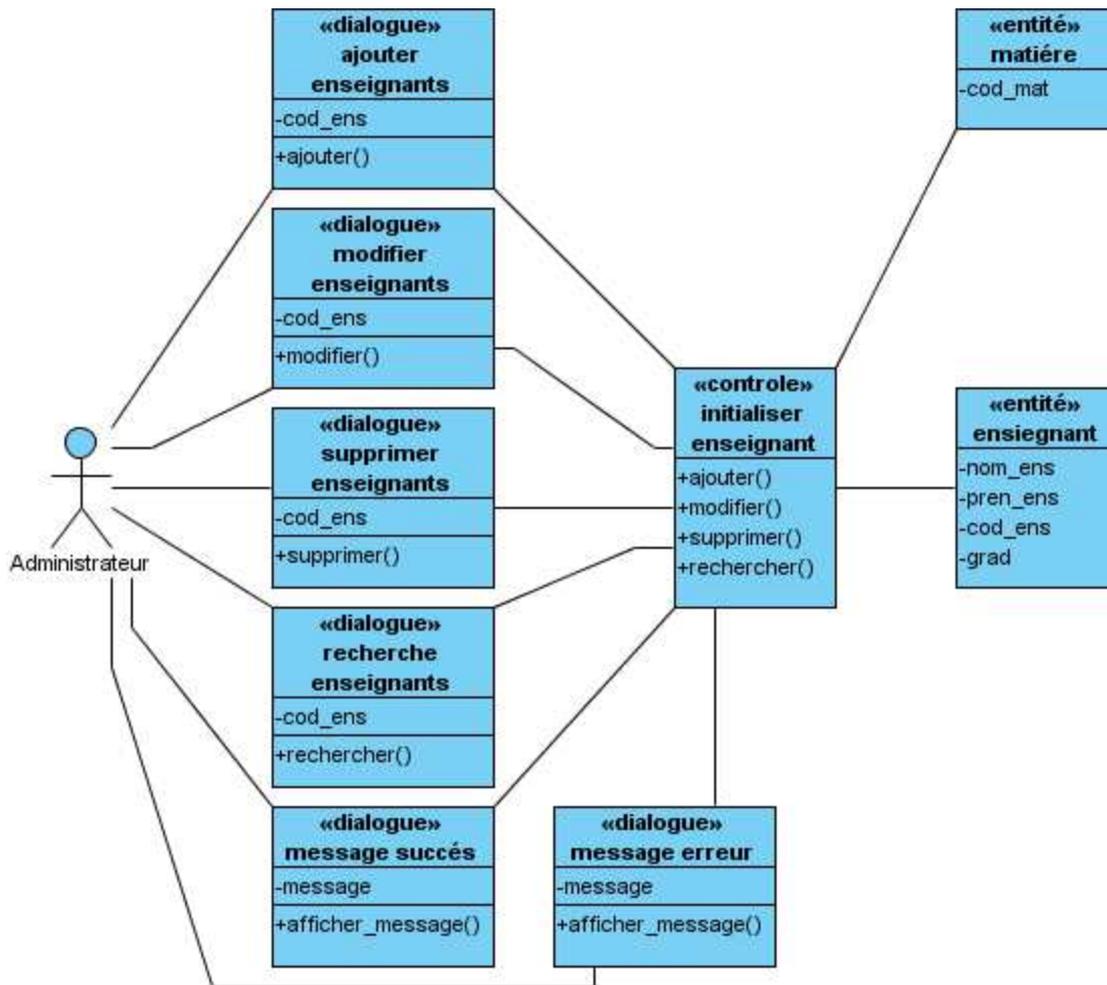


Figure 4.21: Diagramme de classes conception : initialiser les enseignants

### Conclusion :

Dans ce chapitre nous avons élaboré les diagrammes d'interactions qui nous permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse du diagramme de classes participantes. Parallèlement, une première ébauche de la vue statique de conception, c'est-à-dire du diagramme de classes de conception, est construite et complétée.

# Chapitre

## Implémentation

- *Introduction.*

*1- Visual Paradigme for UML 6.4  
Enterprise Edition.*

*2-Modèle relationnelle*

*3- Embarcadero RAD Studio  
2010relationnele.*

*4-La base de donner SQL Server  
R2 2008.*

*5-L'interface graphique.*

-*Conclusion.*

5

## ***Introduction***

Chaque application doit être accompagnée ou illustrée par un dossier technique, c'est-à-dire passer des spécifications détaillées et des contraintes techniques à la définition d'une solution en termes de traitement de données (production d'un logiciel), et qui constitue un guide manuel pour n'importe quel utilisateur. Nous essayerons dans ce dossier de présenter une brève description de l'environnement de programmation et d'autres outils utilisés, mettre l'accent sur quelques caractéristiques de notre application et des ressources utilisées pour le développement. L'environnement de développement opté est Embarcadero RAD Studio 2010 et SQL Server R2 2008.

### **1. Visual Paradigm for UML 6.4 Enterprise Edition:**

Visual Paradigme a introduit la conception soutien Patterns dans cette nouvelle version afin d'améliorer la réutilisation de vos œuvres qui a été fait dans les précédents projets. Dans de nombreux scénarios de développement, nous avons souvent besoin de réutiliser, d'affiner, d'élaborer et de transcrire une partie du schéma, connu comme un modèle de conception, pendant le processus de développement, ou entre les deux phases de développement. Avec cette conception Pattern Repository, vous pouvez désormais le plus besoin re-modèle, les modèles de conception fréquemment utilisés encore et encore.[17]

Développer progressivement conceptuel, logique et physique ERD

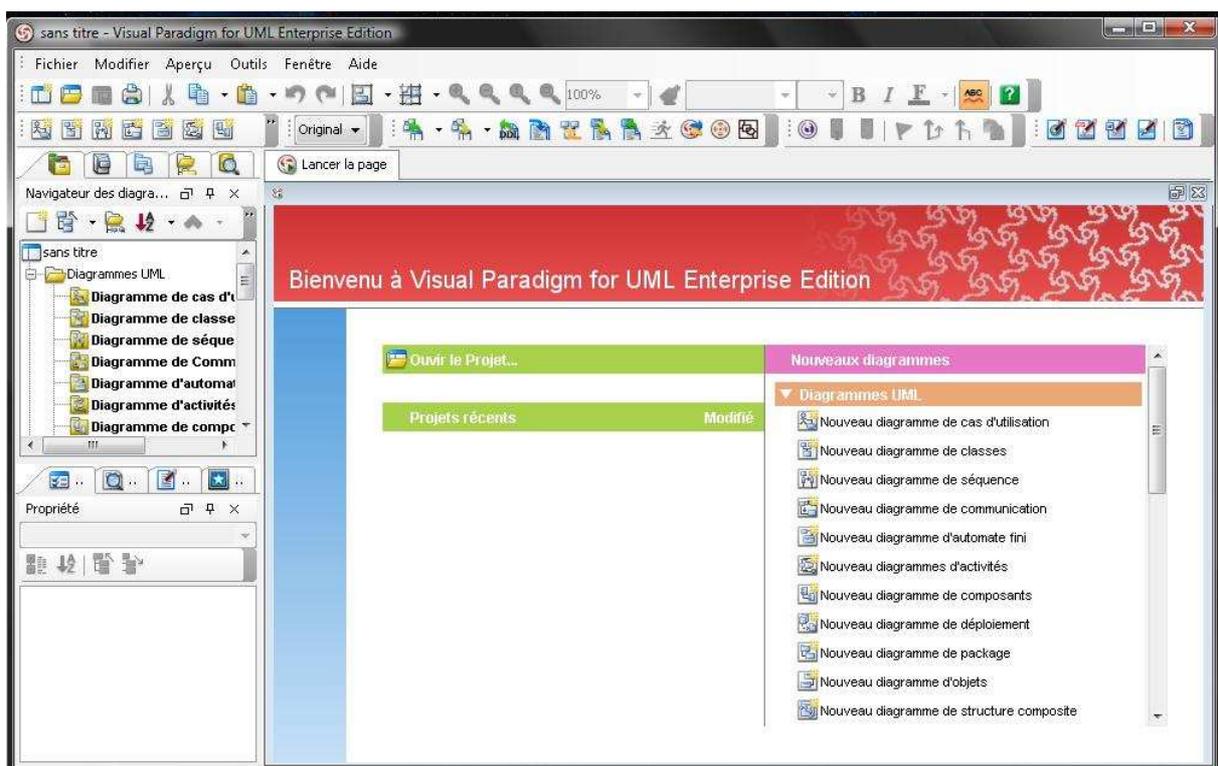
Il existe trois types de ERDs: le cadre conceptuel, la logique et la physique, qui supportent la modélisation des données en trois niveaux d'abstraction du modèle conceptuel de données pour la mise en œuvre d'une base de données spécifique. A partir de cette nouvelle version, Visual Paradigme permet à ces trois niveaux de modélisation des données pour faciliter la création d'un modèle de base de données indépendante sur le système de base de données particulière. Vous pouvez créer un modèle conceptuel et le modèle logique qui sont indépendants sur tous les SGBD. Ensuite, vous pouvez générer le modèle physique pour chaque SGBD que vous allez utiliser pour vos applications.

Démarrage du projet échelle de l'entreprise en quelques secondes et non de minutes

Visual Paradigm a été le chef de file dans le marché des outils de modélisation, et est connu pour être le meilleur de la performance du produit. Nous sommes toujours en faisons tous les efforts pour luttant pour une excellente performance des produits, en particulier sa vitesse de démarrage et de chargement de remplir le métier exigeant de nos jours. Un projet de VP énorme avec plusieurs milliers de formes, il faudrait être lancé dans environ deux minutes de temps dans le passé, mais pour cette nouvelle version, avec la même configuration matérielle, on parle d'une affaire de quelques secondes, ce qui entraîne de notre architecture système unique et la technologie de chargement rapide.

Afficher stéréotypés des éléments de modèle comme une icône de l'image

Les stéréotypes sont des mécanismes d'extensibilité dans UML qui permet aux concepteurs d'étendre le vocabulaire d'UML afin de créer de nouveaux éléments de modélisation, issus de ceux qui existent déjà, mais qui ont des propriétés spécifiques qui sont appropriés pour un domaine de problème particulier ou usage contraire spécialisé.



Par exemple, lors de la modélisation d'un réseau, vous pouvez utiliser une icône de l'image (un symbole) pour représenter les routeurs et concentrateurs. En utilisant des nœuds

stéréotypés vous pouvez faire ces choses apparaissent comme des blocs de construction primitifs et font la spécification beaucoup plus compréhensible.

Importer diagrammes Visio dans Visual Paradigme

Visio est un outil de création de diagrammes largement utilisé dans le marché, qui était aussi un favori pour beaucoup de nos utilisateurs. Mais après, ils ont adopté Visual Paradigme et l'expérience de sa capacité de montage imbattable, la plupart d'entre eux ont demandé pour convertir les schémas existants créés par Visio pour projet de Visual Paradigme.

Placement ciblé des classes dans la synchronisation de classe et ER Diagramme

Dans un système très simple, il peut y avoir qu'un seul diagramme de classes qui inclut normalement toutes les classes créées dans le référentiel de classe. Cependant, dans les projets d'entreprise à grande échelle, nous avons l'habitude d'avoir des centaines de classes, voire plus d'un millier. Chacun des développeurs ne serait d'utiliser un ensemble de classes sélectionnées pour le sous-système (s) il / elle est responsable, soit par un ou plusieurs diagrammes de classes. Ainsi, un diagramme de classes particulier ne comprend pas toutes les classes dans le référentiel de classe. En d'autres termes, un diagramme de classes particulier est une vision statique du sous-système qui comprend seulement un ensemble de classes, il est nécessaire sélectionné. De même, un système complexe peut avoir plus d'un ERD, et chacun de ces ERDs est lié avec seulement une sélection de classes dans le référentiel de classe.

## 2. Modèle relationnel

Le modèle relationnel a été inventé en 1970 et a fait l'objet de très nombreuses recherches qui ont débouché sur la réalisation et la commercialisation des SGBD relationnels. C'est le modèle le plus utilisé par les SGBD actuellement disponible sur le marché.

### 2.1. Définition

Le modèle relationnel est un modèle de données simple fondé sur la théorie mathématique bien connue des relations. Cette théorie se construit à partir de la théorie des ensembles. il y a trois notions importantes pour introduire les bases de données relationnelles [14].

- Domaine : le domaine est un ensemble de valeur caractérisé par un nom.

- Relation : est un sous-ensemble du produit cartésien d'une liste de domaines caractérisé par un nom.

- Attribut : Est une colonne d'une relation caractérisée par un nom.

## 2.2. Représentation de notre base de données :

<b>Relation</b>	<b>Propriétés</b>	<b>Signification</b>	<b>Type</b>
<b>Institut</b>	<u>nom_inst</u> nbr_dom	-nom d'institut -nombre de domaine	Varchar(50) int
<b>Niveau</b>	<u>cod_niv</u> nom_niv	-code de niveau -nom de niveau	Varchar(50) Varchar(50)
<b>Domaine</b>	<u>cod_dom</u> <u>nom_dom</u> nom_inst nbr_par	-code de domaine -nom de domaine -nom d'institut -nombre de parcours	Varchar(50) Varchar(50) Varchar(50) Int
<b>Parcours</b>	<u>nom_par</u> nom_dom nbr_spe	-nom de parcours -nom de domaine -nombre de spécialité	Varchar(50) Varchar(50) int
<b>Spécialité</b>	<u>nom_spe</u> nom_par nbr_sem nom_dom	-nom de spécialité -nom de parcours -nombre de semestre -nom de domaine	Varchar(50) Int Varchar(50) Varchar(50)
<b>Semestre</b>	<u>num_sem</u> nbr_UE nom_spé	-numéro de semestre -nombre d'unité nom de spécialité	Int Int Varchar(50)
<b>Unité d'enseignement</b>	<u>cod_UE</u> nom_UE coeff_UE cred_UE typ_UE nbr_mat	-code d'unité -nom d'unité -coefficient d'unité - crédit d'unité -type d'unité -nombre d'unité	Varchar(50) Varchar(50) Real Real Varchar(50) Int
<b>Matière</b>	<u>cod_mat</u> nom_mat coeff_mat coeff_EX coeff_CC cred_mat cod_UE coeff_TD coeff_TP	-code de matière -nom de matière -coefficient de matière -coefficient d'examine -coefficient de contrôle continue - crédit de matière -code d'unité -coefficient de travaux dirigés - coefficient de travaux pratique	Varchar(50) Varchar(50) Real Real Int Real Varchar(50) Int Real
<b>Section</b>	<u>cod_sec</u> nbr_gro nom_niv	-code de section -nombre de groupe -nom de niveau	Varchar(50) Int Varchar(50)

	<i>nom_dom</i> <i>nom_par</i> <i>nom_spe</i>	-nom de domaine -nom de parcours -nom de spécialité	<i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i>
<b>Groupe</b>	<i>num_gro</i> <i>nbr_etud</i> <i>cod_sec</i>	-numéro de groupe -nombre d'étudiants -code de section	<i>Int</i> <i>Int</i> <i>Varchar(50)</i>
<b>Etudiant</b>	<i>num_etud</i> <i>nom_etud</i> <i>pren_etud</i> <i>dat_naiss</i> <i>nationalit</i> <i>adresse</i> <i>sexe</i> <i>p_pere</i> <i>nom_p_mere</i> <i>num_gro</i>	-numéro d'étudiant -nom d'étudiant -prénom d'étudiant -date de naissance -nationalité -adresse -sexe - prénom de père -nom et prénom de mère -numéro de groupe	<i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i> <i>Int</i>
<b>Notes</b>	<i>num_etud</i> <i>cod_mat</i> <i>cod_ens</i> <i>ann</i> <i>not_TD</i> <i>not_TP</i> <i>not_EX</i>	-numéro d'étudiant -code de matière -code d'enseignant -année d'étude -la note de travaux dirigés -la note de travaux pratique -la note d'examen	<i>Int</i> <i>Varchar(50)</i> <i>Varchar(50)</i> <i>Int</i> <i>Int</i> <i>Int</i> <i>Int</i>
<b>Enseignant</b>	<i>cod_ens</i> <i>nom_ens</i> <i>pren_ens</i> <i>grad</i> <i>cod_mat</i>	-code d'enseignant -nom d'enseignant -prénom d'enseignant -grade d'enseignant	<i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i> <i>Varchar(50)</i>
<b>Année d'étude</b>	<i>Ann</i>	-année d'étude	<i>Int</i>

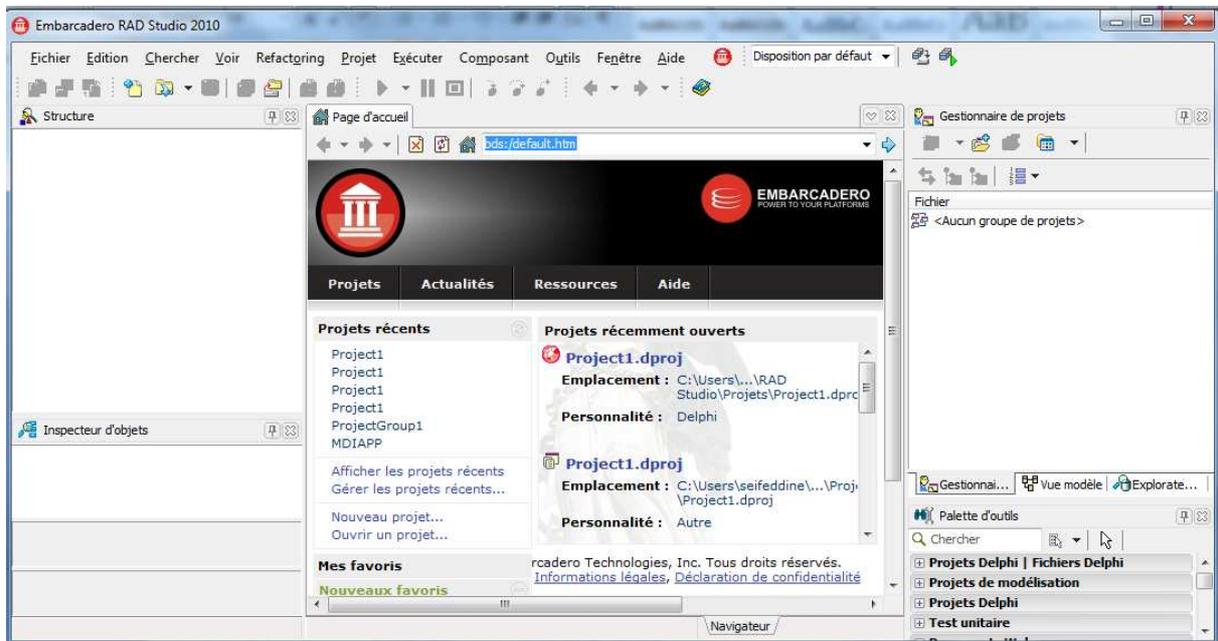
### 3. Le langage de programmation Embarcadero RAD Studio 2010

RAD Studio est un environnement de développement intégré (EDI) pour la construction d'applications Win32. L'EDI de RAD Studio offre un ensemble complet d'outils qui rationalisent et simplifient le cycle de vie des développements. Les outils disponibles dans l'EDI dépendent de la version de RAD Studio que vous utilisez. Les sections suivantes décrivent brièvement ces outils.[19]

#### 3.1. Conception des interfaces Embarcadero RAD Studio 2010

Sur l'espace du concepteur visuel de RAD Studio, vous pouvez créer des interfaces utilisateur graphiques par glisser-déposer des composants de la **palette**

**d'outils** sur une fiche. A l'aide des concepteurs, vous pouvez créer des applications Windows Formes qui utilisent la bibliothèque des composants visuels (VCL) étendue. Vous pouvez également personnaliser vos applications pour différentes versions de Windows, y compris Windows Vista et Windows 7.



### 3.2. Compilation, débogage et déploiement d'applications :

Depuis l'EDI vous pouvez définir les options du compilateur, compiler et exécuter votre application et voir les messages du compilateur. RAD Studio intègre MSBuild comme un moteur de construction, et les commandes de compilation et de construction invoquent MSBuild. Vous pouvez exécuter explicitement MSBuild à l'aide de la ligne de commande ou en utilisant **Invite de commande RAD Studio** sur le menu Démarrer. La commande Invite de commande RAD Studio ouvre une fenêtre console de commande, définit automatiquement le chemin sur l'exécutable MSBuild et initialise la variable d'environnement BDS de telle sorte qu'elle pointe sur votre répertoire d'installation.

Les options du compilateur et plusieurs autres options **Projet > Options** peuvent être enregistrées sous forme de configurations de construction nommées que vous pouvez appliquer à des projets spécifiques. Par défaut, l'EDI fournit les configurations de construction **Débogage** et **Release**.

Le débogueur Win32 intégré vous permet de trouver et de corriger les erreurs d'exécution ou logiques, de contrôler l'exécution du programme et d'exécuter le code pas à pas pour suivre les variables et modifier la valeur des données. RAD Studio inclut Install Aware pour la création d'installations Windows Installer.

### 3.3. Composants visuels :

Les composants visuels apparaissent dans la fiche lors de la conception et sont visibles pour l'utilisateur final à l'exécution. Ce sont les boutons, les libellés, les barres d'outils, les zones de liste, etc.

### 3.4. Aperçu d'une fiche :

Une icône d'aperçu en bas à droite du concepteur (pour les fiches VCL) présente votre fiche telle qu'elle est positionnée à l'écran au moment de l'exécution. Vous pouvez ainsi positionner les fiches de votre application les unes en fonction des autres, au fur et à mesure que vous les concevez.

### 3.5. Palette d'outils :

La **palette d'outils**, située sur la colonne de droite, contient des éléments qui vous permettent de développer votre application. Les éléments affichés dépendent de la vue en cours. Par exemple, si vous affichez une fiche sur le concepteur, la **palette d'outils** affiche les composants appropriés à cette fiche. Double-cliquez sur un contrôle pour l'ajouter à votre fiche. Vous pouvez aussi le faire glisser vers une position désirée de la fiche. Si vous visualisez du code dans l'**éditeur de code**, la **palette d'outils** affiche des segments de code que vous pouvez ajouter à votre application.

### 3.6. Composants personnalisés :

Outre les composants qui sont installés avec RAD Studio, vous pouvez ajouter des composants personnalisés ou tiers à la **palette d'outils** et les enregistrer dans leur propre catégorie.

### 3.7. Templates de composants :

Vous pouvez créer des templates composés d'un ou de plusieurs composants. Après avoir organisé les composants sur une fiche, défini leurs propriétés et écrit du code pour eux, vous pouvez les enregistrer sous la forme d'un template de composant. Par la suite, en sélectionnant le template dans la **palette d'outils**, vous pouvez placer les composants préconfigurés sur une fiche en une seule étape ;

toutes les propriétés et tout le code de gestion d'événement associés sont simultanément ajoutés à votre projet. Vous pouvez repositionner les composants indépendamment les uns des autres, réinitialiser leurs propriétés, et créer ou modifier des gestionnaires d'événements comme si vous aviez positionné chaque composant séparément.

#### **4. SQL Server R2 2008 :**

Microsoft SQL Server Management Studio est un outil puissant et souple. Cependant, les utilisateurs novices de Microsoft Visual Studio rencontrent parfois des difficultés à accéder rapidement à la bonne fonctionnalité. Cette leçon montre comment les fenêtres s'articulent pour offrir une zone de conception facilitant le travail. À la fin de cette leçon, vous aurez appris comment supprimer les fenêtres inutiles, comment afficher rapidement des fenêtres avec les informations de votre choix et comment configurer la zone de conception pour vous aider dans vos tâches récurrentes.[18]

Le didacticiel SQL Server Management Studio. Intégré à Microsoft Visual Studio, Management Studio fournit des outils graphiques pour l'administration des bases de données ainsi qu'un environnement de développement riche. Management Studio permet d'accéder et de gérer le Moteur de base de données, Analysis Manager et l'Analyseur de requêtes SQL dans un seul outil. Il permet en outre d'écrire des instructions Transact-SQL, MDX, XMLA et XML.

Ce didacticiel vous aidera à comprendre la présentation des informations dans Management Studio et à tirer parti des fonctionnalités.

#### **4.1Contenu du didacticiel :**

La meilleure façon de se familiariser avec Management Studio consiste à effectuer des exercices pratiques. Ce didacticiel vous apprendra à gérer les composants de Management Studio et à rechercher les fonctionnalités utilisées régulièrement.

#### **4.2.Composants Management Studio :**

Management Studio fournit des informations dans des fenêtres dédiées à des types d'informations précis. Les informations sur les bases de données s'affichent dans l'Explorateur d'objets et dans des fenêtres de documents.

- L'Explorateur d'objets est une arborescence qui présente tous les objets de base de données sur un serveur. Elle peut inclure les bases de données de Moteur de base de données SQL Server, Analysis Services, Reporting Services, Integration Services et de SQL Server Compact 3.5 SP2. L'Explorateur d'objets contient des informations sur tous les serveurs auxquels il est connecté. Lorsque vous ouvrez Management Studio, le système vous demande de vous connecter à l'Explorateur d'objets avec les derniers paramètres utilisés. Vous pouvez double-cliquer sur n'importe quel serveur dans la fenêtre Serveurs inscrits pour y établir une connexion, mais il n'est pas nécessaire d'inscrire un serveur pour pouvoir vous y connecter.
- La fenêtre de document occupe la plus grande partie de Management Studio. Elle contient les fenêtres des éditeurs de requêtes et du navigateur. Par défaut, la page Résumé s'affiche, connectée à l'instance de Moteur de base de données sur l'ordinateur actif.

### 4.3. Affichage de fenêtres supplémentaires :

Les utilisateurs familiarisés avec l'outil SQL Server Enterprise Manager de SQL Server 2000 peuvent à leur guise afficher la fenêtre Serveurs inscrits.

Pour afficher la fenêtre Serveurs inscrits

1. Dans le menu Affichage, cliquez sur Serveurs inscrits.

La fenêtre Serveurs inscrits apparaît au-dessus de l'Explorateur d'objets. La fenêtre Serveurs inscrits présente la liste des serveurs que vous administrez fréquemment. Il est possible d'ajouter et de supprimer des serveurs de cette liste. Si SQL Server 2000 Enterprise Manager était déjà installé sur cet ordinateur, le système vous demande d'importer la liste des serveurs inscrits. Sinon, les seuls serveurs répertoriés sont les instances SQL Server installées sur l'ordinateur sur lequel vous exécutez Management Studio.

2. Si votre serveur n'apparaît pas dans la liste des serveurs inscrits, cliquez avec le bouton droit sur Moteur de base de données et choisissez Mettre à jour l'inscription du serveur local.

## 5. Les interfaces graphiques :

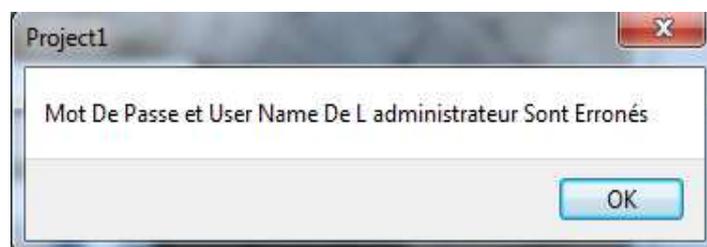
Dans cette partie nous allons présenter quelque fenêtre « notre application » :

### 5.1. Fenêtre authentification :

Permet d'accéder au système avec le login (nom d'utilisateur) et le mot passe de la base de donnée.



Si le mot de passe est erroné alors on affiche la boîte de dialogue suivant :



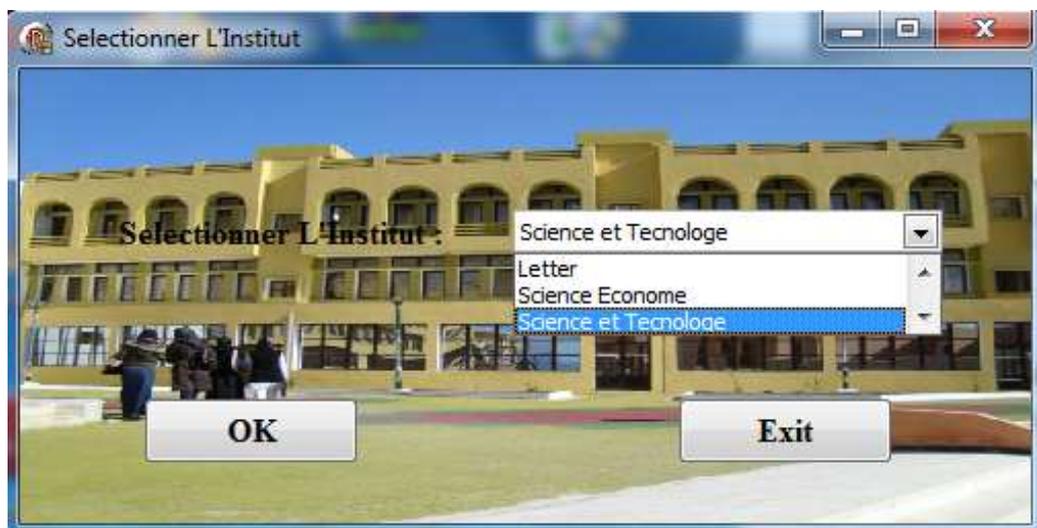
### 5.2. Fenêtre de la délibération:

Est affichée directement après l'authentification.



### 5.3. Fenêtre sélection institut :

Est affichée si l'utilisateur clique sur le bouton «Initialisation », elle permet de sélectionner le nom d'institut.



#### 5.4. Fenêtre Initialisation :

Est affichée si l'utilisateur clique sur le bouton «OK », elle permet d'afficher le menu de composent.



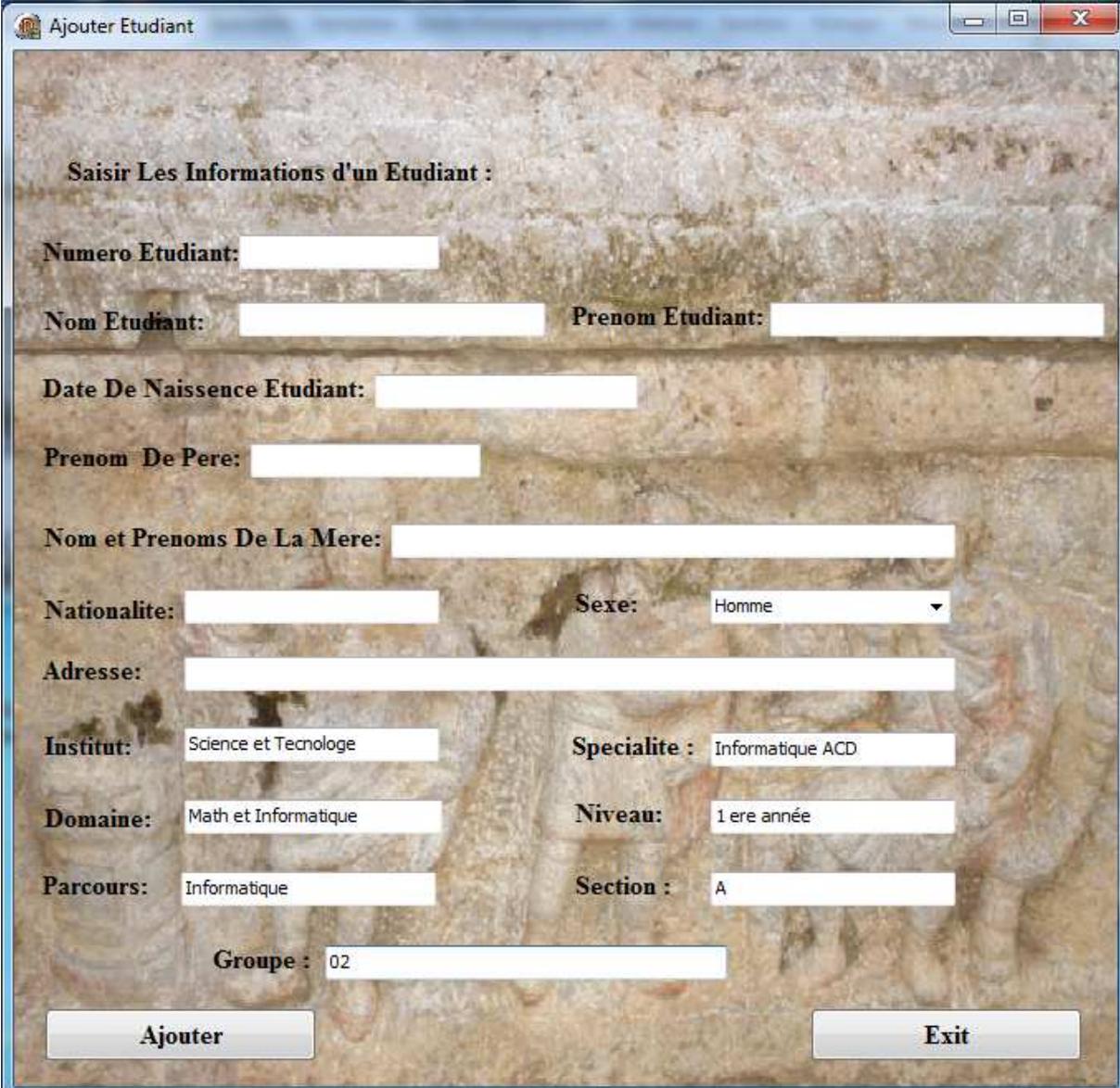
#### 5.5. Fenêtre sélection étudiant :

Est affichée si l'utilisateur clique sur le menu " étudiant>Ajouter étudiant", elle permet de sélectionnée le niveau, le domaine, le parcours, spécialité, section, groupe



### 5.6. Fenêtre ajouter étudiant

Est affichée si l'utilisateur clique sur bouton «OK »de la fenêtre « sélection », elle permet de saisir les informations d'un étudiant.

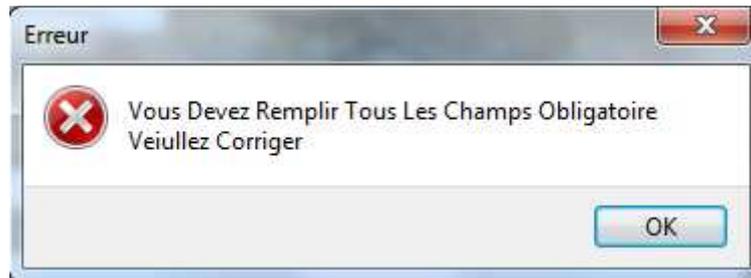


The screenshot shows a window titled "Ajouter Etudiant" with a form for entering student information. The form is set against a textured, stone-like background. The fields are as follows:

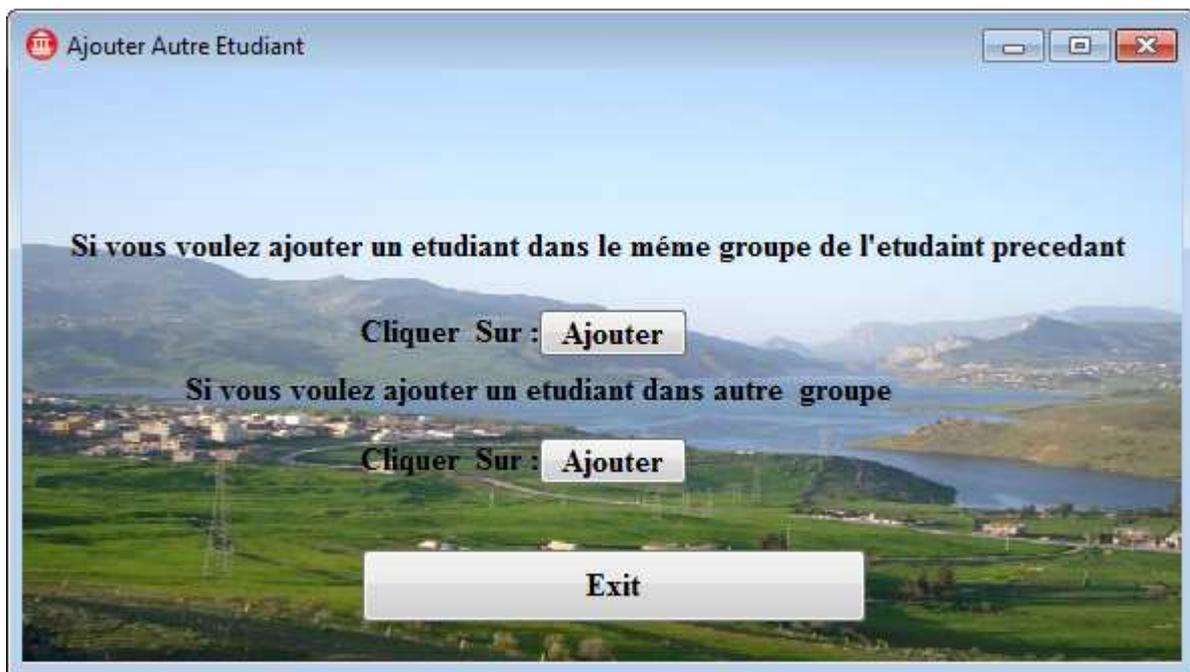
- Saisir Les Informations d'un Etudiant :**
- Numero Etudiant:** [Empty text box]
- Nom Etudiant:** [Empty text box]
- Prenom Etudiant:** [Empty text box]
- Date De Naissance Etudiant:** [Empty text box]
- Prenom De Pere:** [Empty text box]
- Nom et Prenoms De La Mere:** [Empty text box]
- Nationalite:** [Empty text box]
- Sexe:** [Dropdown menu with "Homme" selected]
- Adresse:** [Empty text box]
- Institut:** [Text box containing "Science et Technologie"]
- Specialite :** [Text box containing "Informatique ACD"]
- Domaine:** [Text box containing "Math et Informatique"]
- Niveau:** [Text box containing "1 ere année"]
- Parcours:** [Text box containing "Informatique"]
- Section :** [Text box containing "A"]
- Groupe :** [Text box containing "02"]

At the bottom of the form, there are two buttons: "Ajouter" on the left and "Exit" on the right.

*Boite de dialogue « message d'avertissement » .tous les champs devons être remplis.*

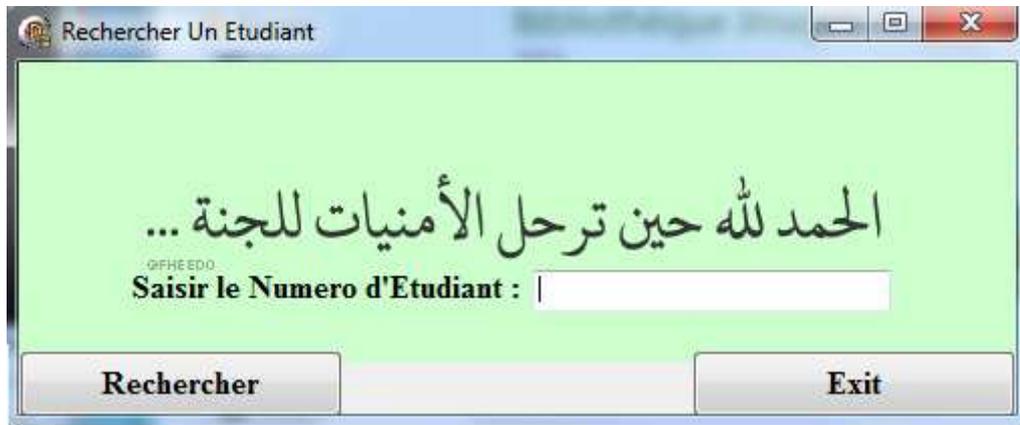


*Boite de dialogue « message d'avertissement »*



**5.7. Fenêtre recherche étudiant :**

Est affichée si l'utilisateur clique sur le menu " étudiant>modifier étudiant", elle permet de rechercher par numéro d'étudiant.

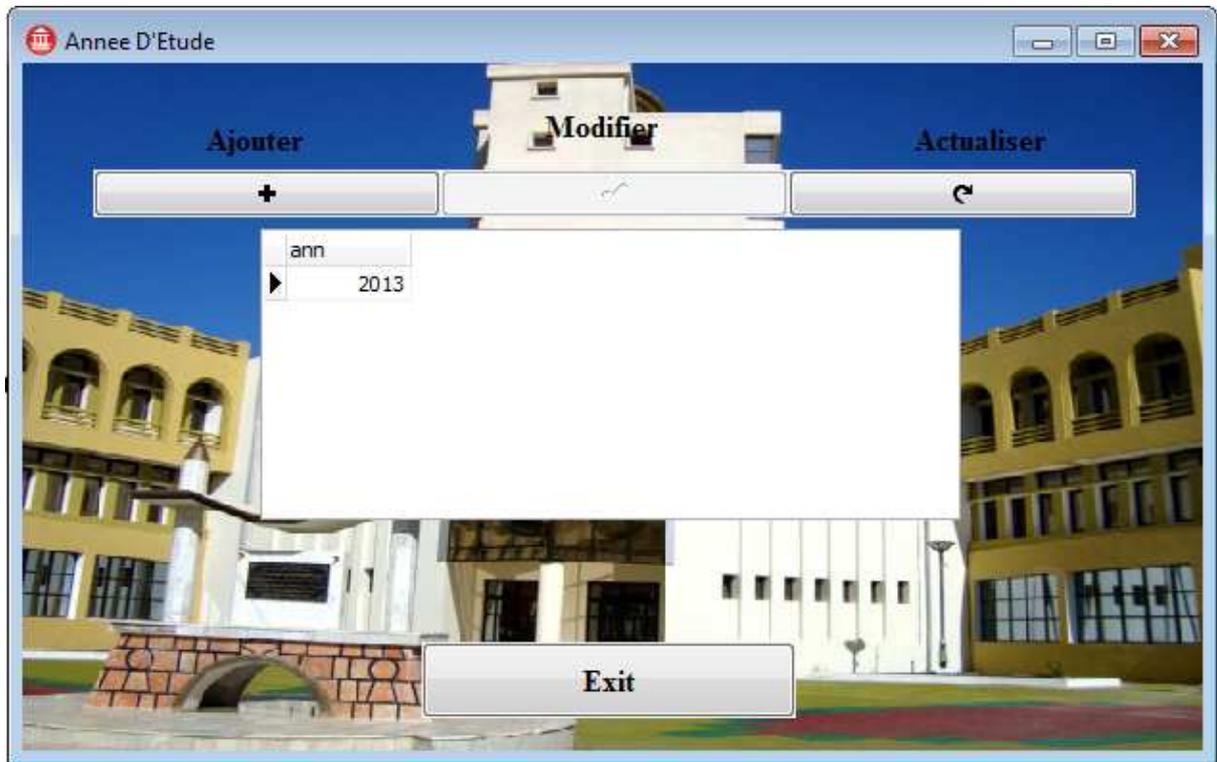
**5.8. Fenêtre initialiser institut :**

Est affichée si l'utilisateur clique sur le bouton « ajouter Institut ou modifier » de la fenêtre « initialiser institut », elle permet de saisir les informations d'institut.



### 5.9. Fenêtre année d'étude:

Est affichée si l'utilisateur clique sur le bouton « ajouter ou modifier année d'étude » de la fenêtre « année d'étude », elle permet de saisir les informations d'institut.



### 5.10. Fenêtre confirmer :

Est affichée si l'utilisateur voulez quitter l'application.



**Conclusion :**

Dans ce chapitre, nous avons abordé tous les détails techniques en rapport avec le modèle d'implémentation de notre application. Dans sa première partie, ce chapitre a été consacré à la présentation des différents outils utilisés pour la réalisation de notre site. Nous avons introduit les concepts relatifs à l'environnement de développement tels qu'Embarcadero RAD Studio 2010, SQL Server R2 2008.

Dans la deuxième partie, nous avons présenté le dossier technique de notre application. Toutes les fonctionnalités de notre outil sont accessibles à travers une interface graphique permettant de faciliter l'interaction entre l'utilisateur et le système.

## *Conclusion générale*

*Le travail présenté dans ce mémoire, ayant pour thème gestion scolaire au niveau des instituts du centre universitaire de Mila.*

*Le langage UML a été utilisé pour la modélisation de notre application et nous avons utilisé « Visual Paradigme » pour établir les diagrammes du modèle de domaine de notre projet.*

*Nous avons pur-survoler les phases d'identification des besoins, analyse, conception et implémentation.*

*L'environnement de développement « Embarcadero RAD Studio 2010 » ont été utilisé pour l'implémentation de notre application avec une base de données « SQL Server R2 2008 ».*

*Malgré que nous n'ayons pas eu assez du temps, nous avons réalisé une grande partie de notre objectif.*

# *Les références bibliographiques*

[1]	<i>UML 2 en action de l'analyse des besoins a la conception, Pascal Roques .Franch Valle, db de Sébastopol, 2009, 4 éditions.</i>
[2]	<i>UML 2 pratique de la modélisation, Benoit Charroux. Aomar Osmaini. Yann Thierry Mieg, la source d'or, 2008, 2<sup>eme</sup> édition</i>
[3]	<i>UML 2 modéliser une application web, Pascal Roques, la source d'or, 2008,4<sup>eme</sup> éditions.</i>
[4]	<i><a href="http://présentation.synthétique.du.UML.org/">http // présentation synthétique du UML.org/</a></i>
[5]	<i><a href="http://UML.free.fr/">http//UML.free.fr/</a></i>
[6]	<i>UML 2, Laurent AUDIBERT, Édition 2007-2008</i>
[7]	<i>UML2, initiation exemples et exercice corrigés, Laurent Debrauwer. Fien Van Der Heyde, la source d'or, 2008, Seconde Edition.</i>
[8]	<i>UML 2 pour l'analyse d'un système d'information, Chantal Morley, Jean Hugues, Bernard Leblanc, les presses de snel Z.I, 2008, 4 éditions.</i>
[9]	<i>« Conception et réalisation d'un site Dynamique Pour Une Agence Commerciale - Etude de cas Gestion des demandes et réclamations des abonnées », Chahdane Karima, Benhammada Lamya, Université Mentouri de Constantine 2010.</i>
[10]	<i>Mémoire gestion électronique des données université Mentouri de Constantine 2010.</i>
[11]	<i><a href="http://www.comment ça marche.com/">http://www.comment ça marche.com/</a>.</i>
[12]	<i><a href="http://www.web.libre.org/">http://www.web.libre.org/</a>.</i>
[13]	<i>Mémoire de fin d'étude «UML pour la réalisation d'un système de gestion des ressources humaines au sein de la direction SONELGAZ de la Wilaya de MILA ».</i>
[14]	<i><a href="http://fr.wikipedia.org">http://fr.wikipedia.org</a></i>
[15]	<i><a href="http://office.microsoft.com/fr-be/access-help/microsoft-office-access-2007-les-dix-principaux-avantages-HA010165021.aspx">http://office.microsoft.com/fr-be/access-help/microsoft-office-access-2007-les-dix-principaux-avantages-HA010165021.aspx</a></i>
[16]	<i>Mémoire de fin d'étude «gestion de l'emploi de temps», Bouridah Soad Saidani Aicha, centre université Mila</i>

[17]	<a href="http://msdn.microsoft.com/fr-fr/library/ms166547(v=sql.105).aspx">http://msdn.microsoft.com/fr-fr/library/ms166547(v=sql.105).aspx</a>
[18]	<a href="http://www.visualparadigm.com/support/vpuml/releasenotes/640.jsp">http://www.visualparadigm.com/support/vpuml/releasenotes/640.jsp</a>
[19]	<a href="http://www.embarcadero.com/products/rad-studio">http://www.embarcadero.com/products/rad-studio</a>