

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



**Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.**
Option : IA

Sujet :

**Extension d'une plate forme de
E-learning par les aspects de suivi
des étudiants.**

Présenté par : MANCER Yasmine
HASSAINE Yamina

Promoteur : Mr D. Bennouar

- 2006/2007-

MIG-004-163-1

Remerciements

Nous tenons à remercier avant tout Dieu tout puissant de nous avoir donné la force, le courage et la volonté pour réaliser ce travail.

Nous remercions chaleureusement notre promoteur Monsieur D.Bennouar, d'avoir bien voulu diriger ce projet. Nous lui sommes reconnaissantes pour son appui et ses conseils, ainsi que pour sa disponibilité et la confiance qu'il nous a accordée.

Sans oublier tous les enseignants de notre département qui ont participé à notre formation d'ingénieur.



Dédicace

Je dédie ce travail :

A toute ma famille,

A toutes mes amies,

Et à tous ceux qui me sont chers.

M.Yasmine



Dédicace

Je dédie ce travail :

A mes chers parents,

A mes chers frères et sœurs,

A toute ma famille,

A mon fiancé et sa famille,

A ma binôme Yasmine et toutes mes amies,

*A tous les enseignants et les étudiants du
département informatique de l'université de blida*

H. Yamina

SOMMAIRE

Introduction générale

1. Présentation du sujet1
2. Approche méthodologique.....3

Chapitre I : E-learning

1. Définition du E-learning..... 8
2. Typologie de l'apprentissage en ligne8
 - 2.1 100% apprentissage en ligne8
 - 2.2 Mixte..... 9
3. Les avantages du E-learning.....9
4. Les plates formes de E-learning.....10
5. Comparaison entre 6 plates formes de E-learning.....10
 - 5.1 Les principaux critères que doivent vérifier les plates formes10
 - 5.2 Tableau de comparaison.....11
6. La plate forme Claroline.....13

Chapitre II : Analyse des besoins

1. Introduction 14
2. Modélisation des besoins..... 14
 - 2.1 Comportement du système.....14
 - 2.2 les acteurs.....14
 - 2.3 les cas d'utilisation.....15

3. Diagramme des cas d'utilisation.....	16
---	----

Chapitre III : La conception

1. Introduction	21
2. Description des cas d'utilisation	22
3. Diagramme de classe global	107
4. Description du diagramme de classe	108

Chapitre IV : Implémentation et réalisation

1. Introduction	111
Détail de l'architecture trois tiers.....	112
2. Les trois couches	113
2.1 Couche Présentation (premier niveau)	113
2.2 Couche Métier / Business (second niveau)	113
2.3 Couche Accès aux données (troisième niveau)	113
3. Stratégie de développement.....	114
4. Configuration du système	115
5. Installation du système	116
6. La présentation du logiciel	119
 conclusion générale.....	 151

Résumé

La E-formation est la possibilité de suivre un programme de formation à distance, en autoformation ou accompagné, de manière individuelle ou collective.

La E-formation s'appuie sur Internet et les outils multimédia pour offrir des modules de formations courts, progressifs et adaptés aux niveaux et besoins des apprenants.

De nombreuses plates-formes de E-formation proposent des outils de gestion des formations et des apprenants mais les utiliser suppose l'adhésion à un certain type d'évaluation, qui n'est pas forcément adapté à tous les enseignements.

Notre objectif est de faire l'extension d'une plate-forme de E-formation pour lui permettre de gérer les aspects de suivi des étudiants et de s'adapter à l'enseignement universitaire et plus particulièrement au système LMD.

Abstract

The E-learning is the possibility of following a training scheme remote, in self-training or accompanied, in an individual or collective way.

The E-learning is pressed on Internet and the tools multi-media to offer modules of formations short, progressive and adapted to the levels and needs for learning.

Many platforms of E-learning propose management tools of the formations and learning but to use them adhesion with a certain type of evaluation supposes, which is not inevitably adapted to the entire lesson.

Our objective is to make the extension of a platform of E-learning to enable him to manage the aspects of follow-up of the students and to adapt to the university education and more particularly to the system LMD.

Introduction

1. Présentation du sujet :

L'apprentissage en ligne ou e-learning est une modalité pédagogique et technologique qui concerne la formation continue, mais aussi l'enseignement supérieur.

Actuellement les systèmes de e-learning ne prennent pas en charge les aspects liés à la gestion du suivi des étudiants notamment les aspects passage d'une étape à une autre et tout ce qui en découle dans un contexte représentant la réalité, d'autre part la gestion de la scolarité et de la pédagogie est l'une des principales préoccupations des établissements d'enseignement elle concerne un large public, depuis les futurs étudiants de l'établissement jusqu'aux responsables des services administratifs .

Nous pouvons distinguer trois groupes d'individus en fonction de leurs activités:

- Le groupe administratif : en charge de la direction de l'établissement et qui assure le fonctionnement des services de la scolarité.
- Les enseignants qui assure l'enseignement des étudiants.
- Les étudiants qui suivent des formations dans l'établissement.

Les besoins de chacun de ses groupes sont différents.

Avec la nouvelle génération d'applications centrées sur les serveurs HTTP, la généralisation d'Internet et les plates-formes de E-learning, de nouveaux besoins et problèmes ont émergés dans la gestion de la scolarité et de la pédagogie notamment avec le nouveau système LMD qui est plus complexe dans sa gestion qu'un système d'enseignement "classique".

C'est dans ce nouveau contexte que s'inscrit notre projet de fin d'étude qui consiste à «L'extension d'un système de e-learning par les aspects de suivi des étudiants » son objectif est de chercher des solutions techniques qui permettent d'augmenter la communication entre (administration, enseignant, étudiant) et d'améliorer l'adaptabilité des outils existants (réseau, ordinateur) à l'environnement technique et humain.

L'objectif principal de ce sujet consiste à étendre un système existant par les concepts suivants:

- Une administration et un contrôle plus poussé de l'exploitation.
- Authentification et contrôle d'accès plus élaboré.
- Sécurité accrue de l'information.
- L'ajout de fonctionnalités de suivi des étudiants et les aspects saisi d'informations d'évaluation (assiduité, notes des examens, TP, TD, projet, ..).

En finalité, notre projet permet de gérer les aspects de la scolarité dans une université et pour illustrer notre travail on a choisit en particulier le système LMD, plus l'assurance d'un apprentissage à distance et qui peut s'inscrire dans le cadre de l'enseignant virtuel (université virtuelle) ou bien un support d'aide pour les enseignants et les étudiants (complément du travail fait à l'université) et tous ça depuis n'importe quel poste relié au réseau.

2. Approche méthodologique :

Description de l'environnement

L'environnement de développement du logiciel visé, à savoir L'extension d'un système de e-learning par les aspects de suivi des étudiants, est caractérisé par les éléments suivants:

- Le logiciel à réaliser est un logiciel pour une éventuelle utilisation locale, réalisé dans le contexte de la préparation d'un diplôme de fin d'étude.
- Il doit être conçu et réalisé par deux étudiantes sans aucune expérience, ou haute compétence concernant les outils de programmation de programme centré sur les serveurs HTTP.
- Le logiciel n'est pas destiné à être étendu ou maintenu par un autre groupe.
- Il n'y a aucune contrainte financière.
- Une période d'apprentissage des outils est nécessaire.
- Une période est nécessaire pour la compréhension du problème.
- La nécessité de mise en place dans les plus brefs délais d'un prototype fonctionnel.
- Le temps restreint pour terminer le travail (période de stage limitée).

Les méthodes traditionnelles :

Les démarches traditionnelles, basées sur la fameuse séquence « spécification > conception > réalisation > validation », concentrent la plupart des décisions en début de projet.

L'objectif de cette approche est louable : le client veut des garanties sur ce qu'il obtiendra en fin de projet, et le chef de projet souhaite disposer des informations nécessaires à l'organisation de son équipe.

Malheureusement, les équipes qui évoluent dans un environnement changeant ou complexe savent à quel point il est difficile de s'en tenir aux décisions initiales. Le client réalise que ses besoins ont changé, ou bien l'équipe découvre en phase d'implémentation des erreurs de spécification ou de conception qui compromettent les plans de développement.

Le changement s'impose donc tôt ou tard, mais voilà : cette organisation suppose l'absence de changement, et celui-ci se révèle bien vite très coûteux, suffisamment parfois pour compromettre la rentabilité du projet.

Mais puisque le changement est une composante incontournable de tout projet de développement logiciel, pourquoi ne pas l'accepter ?

La méthode proposée :

Vu l'environnement décrit précédemment, et la lourdeur des méthodes traditionnelles et la complexité de leur maîtrise, la méthodologie que nous avons suivie est une méthodologie dite agile et elle est basée sur la programmation intensive, itérative et progressive (incrémentale) centré sur les tests.

La méthodologie incite à commencer par les aspects les plus simples, les réaliser et les tester et passer ensuite à la réalisation d'un autre aspect. Ce dernier aspect pourra mettre en cause les aspects précédents. Dans ce contexte un réajustement des étapes précédentes est nécessaire. Il faut revenir en arrière pour refaire la conception /réalisation.

Ce processus pratique permet d'une part de maîtriser l'outil de programmation et d'avoir à chaque étape une version fonctionnelle d'une partie du logiciel.

Cette méthodologie fait partie d'une famille émergente de processus dit processus "agiles", qui se démarquent des démarches traditionnelles en mettant l'accent sur le travail d'équipe et la réactivité. La méthode XP fait partie de cette famille et la méthodologie que nous avons suivie s'apparente sur beaucoup d'aspect à l'XtremeProgramming. Notre méthodologie se focalise sur la construction proprement dite du logiciel, en aval des phases préparatoires d'études d'opportunité ou de faisabilité.

Dans notre méthode :

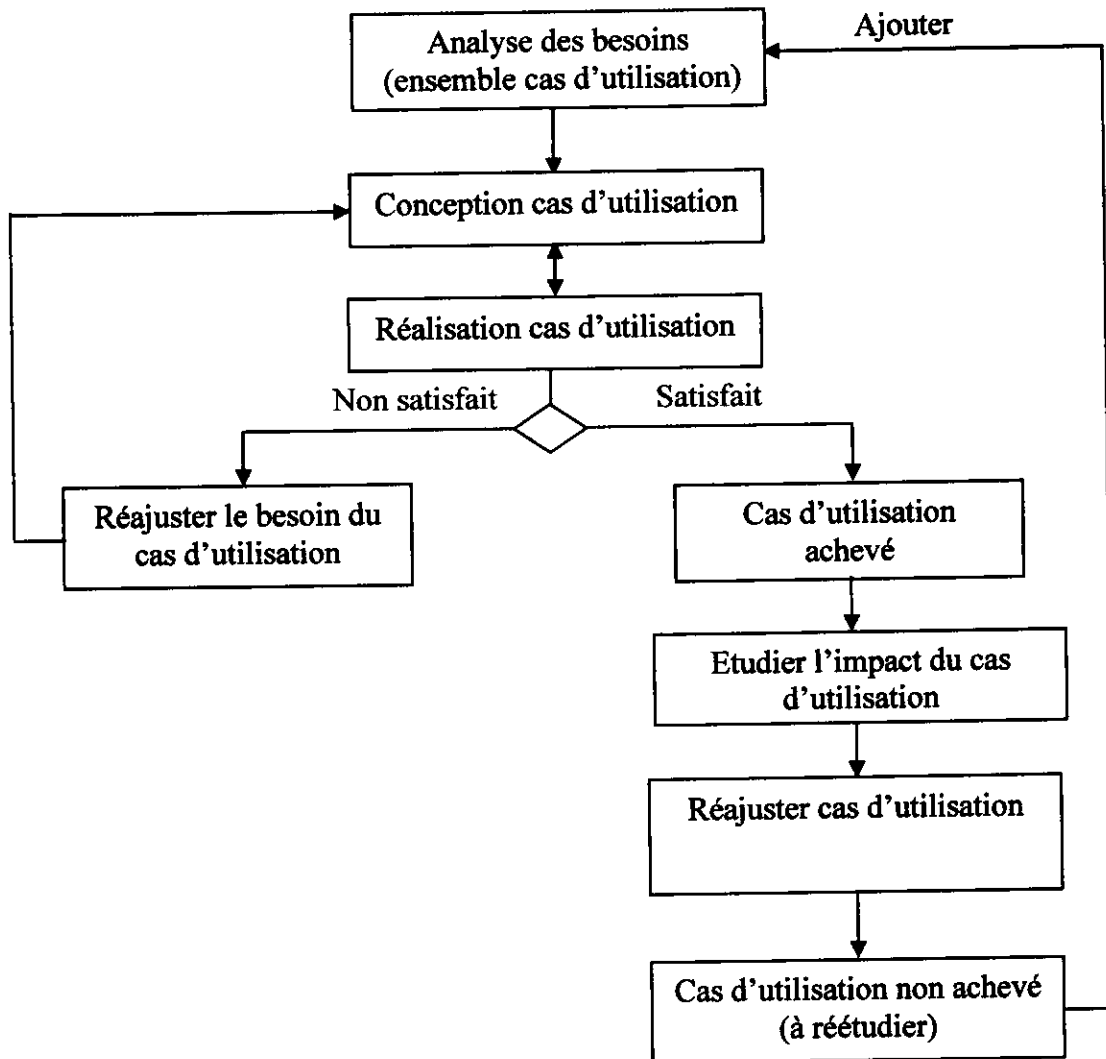
- Le client pilote lui même le projet, et ce de très près grâce à des cycles itératifs extrêmement courts (1 ou 2 semaines). Le client dans notre cas est le promoteur principal du sujet, à savoir Mr Djamal BENNOUAR, Chargé de Cours au Département Informatique de l'USDB.
- L'équipe formé de deux étudiante en phase de préparation de leur mémoire de fin d'étude, livre, très tôt une première version du logiciel, et les livraisons de nouvelles versions s'enchaînent ensuite à un rythme soutenu pour obtenir un feedback maximal sur l'avancement des développements.
- L'équipe constituée des deux étudiantes, s'organise elle même pour atteindre ses objectifs, en favorisant une collaboration maximale entre ses membres.
- L'équipe doit mettre en place un ensemble de jeux d'essai ou doit mettre en place des tests automatiques pour toutes les fonctionnalités qu'elle développe, ce qui devrait garantir au produit un niveau de robustesse très élevé.
- Les développeurs améliorent sans cesse la structure interne du logiciel pour que les évolutions y restent faciles et rapides.

Le grand gagnant de cette démarche est d'abord le client du projet. Plutôt que de voir son intervention cantonnée à la phase initiale à savoir recueil des besoins, il intègre véritablement le projet pour en devenir le pilote. A chaque itération, il choisit lui-même les fonctionnalités à implémenter, collabore avec l'équipe pour définir ses besoins dans le détail, et reçoit une nouvelle version du logiciel qui intègre les évolutions en question.

Cette démarche présente de nombreux avantages en termes de conduite de projet :

- Le client jouit d'une très grande visibilité sur l'avancement des développements.
- Le client utilise le logiciel lui même comme support de réflexion pour le choix des fonctionnalités à implémenter, il peut en particulier intégrer très tôt les retours des utilisateurs pour orienter les développements en conséquence.
- La première mise en production du logiciel intervient très tôt dans le projet, ce qui avance d'autant le moment à partir duquel le client peut en tirer des bénéfices.

- L'ordre d'implémentation des fonctionnalités n'est pas guidé par des contraintes techniques, mais par les demandes du client. Celui-ci peut donc focaliser les efforts de l'équipe sur les fonctionnalités les plus prioritaires (les plus simples dans notre cas) dès le début du projet, et ainsi optimiser l'utilisation d'un éventuel budget.



Organisation du mémoire :

Notre projet consiste à l'extension d'une plate forme de E-learning, pour cela on va parler dans le premier chapitre du E-learning et de la plate forme de l'extension Claroline.

Le second chapitre présente l'analyse des besoins à travers les diagrammes des cas d'utilisation.

Le troisième chapitre présente la conception du projet avec la modélisation UML à travers les scénarios, les diagrammes de séquence, les diagrammes d'activité et les diagrammes de classe.

L'architecture de notre logiciel est l'architecture trois tiers, on va parler de cette architecture dans le quatrième chapitre qui est l'implémentation et la réalisation et on va parler aussi de l'installation et de la configuration ainsi que de la présentation du logiciel.

Chapitre 1 :

E-learning

1. Définition du E-learning : [w-01]

Le e-learning, ou apprentissage en ligne, est une méthode d'apprentissage qui repose sur la mise à disposition de contenus pédagogiques via un support électronique (Cédérom, Internet, intranet, télévision interactive, etc.).

Le e-learning comprend aussi bien des outils et des applications pédagogiques que des contenus pédagogiques.

Le e-learning concerne autant les jeunes qui utilisent ces supports électroniques à des fins pédagogiques que les adultes qui perfectionnent leur formation ou mettent à jour leurs connaissances.

L'apprentissage en ligne (online learning), constituant une des facettes du e-learning, se différencie par l'utilisation de la technologie Web. Ces applications et ces contenus sont dispensés via un ordinateur relié à un intranet ou à Internet et constituent une étape plus avancée que l'utilisation du Cédérom. L'apprentissage en ligne offre des possibilités de collaboration et d'interactivité.

2. Typologie de l'apprentissage en ligne :

On peut définir une formation en ligne selon deux axes:

- ✓ **L'axe de l'espace formateur – apprenant:** Il peut exister une grande proximité, lorsque le professeur donne un cours devant ses étudiants, on parle de formation en présentiel. Lorsque le formateur n'est pas en contact avec l'apprenant, on parlera de formation à distance.
- ✓ **L'axe du temps de la communication:** Si la communication est directe et immédiate, on parle d'outil synchrone. C'est le cas du professeur face à ses étudiants mais aussi lors d'une séance de "chat" ou une visioconférence. Lorsqu'un délai existe entre une question et sa réponse on parlera d'outil asynchrone. Le forum, le courrier électronique sont des outils de communication asynchrone

2.1 100%apprentissage en ligne :

Dans cette typologie de formation, l'apprenant suit sa session d'apprentissage en ligne entièrement à distance. Il n'interagit pas directement en vis-à-vis avec le ou les formateurs. Il peut néanmoins naturellement avoir des contacts avec eux via des outils type forum, chat, téléphone, email.

2.2 Mixte

Le mode d'apprentissage mixte (appelé aussi Blended Learning en anglais) désigne l'utilisation conjointe du e-Learning et du mode classique d'apprentissage appelé souvent « présentiel ». En général, l'apprenant va ainsi alterner entre des sessions à distance en ligne et des sessions en face-à-face avec le ou les formateurs.

3. Les avantages du E-learning :

1. Se former sans se déplacer

La E-formation offre plus de souplesse dans la planification des sessions. Elle permet d'éviter les pertes de temps liées aux déplacements ainsi qu'à leurs organisations. La E-formation est donc particulièrement adaptée aux contraintes d'organisation croissante des services et des entreprises.

La formation est donc moins contraignante car elle peut être suivie depuis le poste de travail de l'apprenant, en centre de ressources ou sur un poste dédié.

2. Se former de manière individuelle ou collective

Toutes les E-formations se déclinent en cours individuels ou collectifs. Ces différents contextes d'apprentissage apportent à l'apprenant une progression assurée. Dans le cadre de cours particulier, il a la possibilité d'utiliser ses propres documents de travail.

Dans tous les cas une véritable relation "continue et individualisée" est établie entre apprenants et formateurs (relation interactive, collaborative et dynamique).

3. Se former de manière progressive et souple

La E-formation permet une approche plus modulaire des programmes de formation. Les découpages pédagogiques sont propices à une meilleure progression et mise en pratique immédiate par les stagiaires.

Le stagiaire est véritablement au cœur du dispositif ; il peut choisir son rythme de progression, la fréquence de ses sessions de formation et pilote véritablement sa formation. Il en devient l'acteur et le metteur en scène.

La E-formation permet la réduction des coûts globaux de formation.

Elle permet entre autres :

- La réduction des temps de formation et la réduction consécutive des absences des collaborateurs à leur poste de travail.
- La réduction des frais indirects (coût d'organisation, transport, hébergement, restauration,...).
- Une augmentation du rapport "efficacité de la formation/temps et investissements consacrés".

4. Les plates formes de E-learning : [w-02]

Une plate forme de E-learning est un système informatique de gestion de formation et de gestion de contenu de formation.

Il existe aujourd'hui environ 250 plates-formes e-Learning ! Une trentaine sont en Open Source et gratuites.

5. Comparaison entre 6 plates formes E-learning : [w-03]

La comparaison entre 3 plates formes gratuites et 3plates formes payantes:

Les plates formes gratuites :

- Claroline
- ACOLAD
- MOODLE

Les plates formes payantes :

- LEARNING SPACE
- VIGIPORTAL
- WEBCT

5.1 Les principaux critères que doivent vérifier les plates-formes :

- **Le coût total** de la mis en place de la plate-forme (gratuite ou non).
- **Les besoins de maintenance** de la plate-forme d'un point de vue strictement informatique, tout ce qui devra être géré par les administrateurs (voir également le coût en personnel de maintenance).
- **La sécurité de la plate-forme** d'un point de vue purement informatique (login // mot de passe).
- **Le nombre d'utilisateurs maximum** que pourra supporter la plate-forme.
- **La possibilité de mettre en place des QCM** (Questions à Choix Multiples) sur la plate-forme, s'il est possible d'insérer une image, si le système de questions est interactif (avec animations), s'il y a possibilité de faire un tirage aléatoire des questions.
- **Un forum de discussion** asynchrone (dont l'accès sera étroitement lié à la sécurité interne de la plateforme) et un système de chat synchrone.

- **La technologie utilisée**, du point de vue du langage pour pouvoir éventuellement faire évoluer le système et sa capacité d'accueil.
- **Un système de gestion de mails**, pour qu'un professeur puisse éventuellement envoyer des mails à l'une de ses classes, également lui permettre de gérer des groupes.
- **La façon de poster un cours** (à partir de documents Powerpoint, enregistrés au format HTML et transférés sur un serveur).
- **Les moyens requis pour consulter la plate-forme.**
- **Est-il possible d'arriver à terme à une formation purement virtuelle**, sans aucune attache physique entre étudiants et professeurs ?
- **La possibilité de diffuser les cours par vidéo** soit en temps réel, soit en différé.

Le point le plus important est de garder en mémoire que la plate-forme doit pouvoir accueillir une grande base (environ 13 000 étudiants et 800 professeurs).

Pour le bon fonctionnement d'une plate-forme, il y a 90% de chances qu'il faille mettre en place un studio de création de cours car le nombre de types ou de formats de cours qui peuvent être mis à disposition restent immense.

5.2 Tableau de comparaison :

	Comptes utilisateurs	QCM	chat	mails	forum	Cours vidéo
CLAROLINE	20 000	oui	non	oui	oui	en différé
ACOLAD	15 000	non	oui	oui	oui	en différé
MOODLE	20 000	oui	non	non	oui	non
VIGIPORTAL	20 000	oui	oui	oui	oui	non
WEBCT	25 000	oui	oui	oui	oui	en différé
LEARNINGSPEACE	20 000	oui	oui	oui	oui	non

Création / Gestion des documents pédagogiques	
CLAROLINE	Tous types de fichiers envoyés à partir d'un navigateur
ACOLAD	Tous types de fichiers envoyés à partir d'un navigateur
MOODLE	Système propriétaire intégré
VIGIPORTAL	Système propriétaire intégré
WEBCT	Système propriétaire intégré
LEARNINGSPACE	Tous types de fichiers envoyés à partir d'un navigateur

Technologie Employée	
CLAROLINE	PHP / MySQL
ACOLAD	SQL, Access 2000, ASP, JRE, PHP4
MOODLE	PHP / MySQL
VIGIPORTAL	PHP / MySQL
WEBCT	Oracle / Java / PHP
LEARNINGSPACE	PHP / MySQL

6. La plate forme Claroline : [w-04]

Claroline est une plate-forme Open Source, distribuée sous licence GPL, qui permet à des centaines d'institutions issues de 84 pays de créer gratuitement des espaces cours en ligne. Pour chaque cours, le formateur dispose d'une série d'outils lui permettant de :

- Rédiger une description du cours
- Publier des documents dans tous les formats (texte, PDF, HTML, vidéo...)
- Administrer des forums de discussion publics ou privés
- Elaborer des parcours pédagogiques
- Créer des groupes de participants
- Composer des exercices
- Structurer un agenda avec des tâches et des échéances
- Publier des annonces (aussi par e-mail)
- Proposer des travaux à rendre en ligne
- Consulter les statistiques de fréquentation et de réussite aux exercices

Adaptable à différents contextes de formation, Claroline est utilisée non seulement dans les écoles et les universités, mais également dans les centres de formation, les associations et les entreprises.

Une technologie appropriée :

Claroline peut facilement accueillir un très grand nombre d'utilisateurs. Elle est compatible avec les environnements Linux, Mac OS et Windows.

Claroline est réalisée grâce à des technologies libres telles que PHP et MySQL et utilise des standards tels que SCORM et IMS/QTI pour les échanges de contenu.

Qui développe Claroline ?

- Le logiciel a été initialisé par l'Université de Louvain (Belgique) et est distribué sous une licence Open Source (GPL). Depuis lors, une communauté mondiale de développeurs contribue à son développement. Télécharger et utiliser Claroline est complètement gratuit.
- Claroline a été développé par un réseau international de professeurs et de développeurs dispersés dans le monde. Ils recyclent des programmes entiers ou des morceaux de code trouvés dans la vaste bibliothèque de la communauté intermédiaire du GPL Open Source internet.
- Thomas De Praetere a créé Claroline à l'Université Catholique de Louvain (IPM - Institut de Pédagogie universitaire et des Multimédias). Il a été aidé financièrement par la Fondation Louvain. Hugues Peeters (qui a inventé le nom "Claroline").

Chapitre 2 :
Analyse
des
besoins

1. Introduction :

L'analyse des besoins est l'une des étapes les plus importantes, en effet elle est l'origine de toute activité de développement et donc c'est une étape déterminante pour la suite. L'analyse des besoins s'intéresse d'abord au « QUOI », à savoir les besoins du client ou de l'utilisateur.

Pour exprimer les besoins UML propose les cas d'utilisation ou « use cases ».

L'analyse se poursuit par la modélisation du domaine, à savoir par l'identification des objets et des interactions entre objets, par l'identification des classes et des relations entre classes.

Ces objets et classes appartiennent au domaine, ils n'expriment en aucun cas la solution informatique, celle-ci étant modélisée lors de la conception. Pour réaliser cette modélisation UML propose " les diagrammes de cas d'utilisation ".

2. Modélisation des besoins :

2.1. Comportement du système :

La description du comportement d'un système est illustrée par un modèle de cas d'utilisation qui met en évidence les fonctions attendues du système (cas d'utilisation), un environnement (les acteurs) et les relations entre les cas d'utilisation et acteurs (diagramme de cas d'utilisation).

2.2. Les acteurs :

Les acteurs représentent l'environnement du système. Dans notre cas les acteurs sont :

- ✓ L'administrateur.
- ✓ L'enseignant.
- ✓ L'étudiant.
- ✓ Le visiteur.

2.3. Les cas d'utilisation:

- ✓ Pour l'administrateur :
 - Installation du système.
 - Gestion du système.
 - Gestion des utilisateurs.
 - Consultation.
 - Discussion.

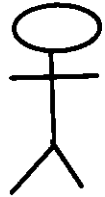
- ✓ Pour l'enseignant :
 - Gestion des cours.
 - Consultation.
 - Discussion.

- ✓ Pour l'étudiant :
 - Consultation.
 - Passer les examens.
 - Réaliser les travaux.
 - Discussion.

- ✓ Pour le visiteur :
 - Consultation.

3. Diagramme des cas d'utilisation :

Administrateur



Systeme

Installation du système

Gestion des universités

Gestion des facultés

Gestion des départements

Gestion des formations

Gestion des unités
d'enseignement

Gestion des cours

Gestion des enseignants

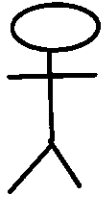
Gestion des étudiants

Le nombre de sections et de
groupes par niveau

Affectation des enseignants
aux cours

Affectation des étudiants
aux sections et aux groupes

Administrateur



Systeme

Consulter la liste des étudiants d'un niveau

Consulter les relevés de notes

Gestion des dettes

Consulter le PV des délibérations

Etablir le passage

Gestion des outils des cours

Agenda

Forums

Annonces

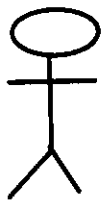
Messagerie

Analyser les statistiques

Mise à jour

Ouverture et fermeture de l'année scolaire

Enseignant



Systeme

Créer un cours

Gestion de ses cours

Préparer les examens

Saisir les notes des examens

Consulter la liste des étudiants d'un niveau

Consulter les relevés de notes

Consulter les dettes

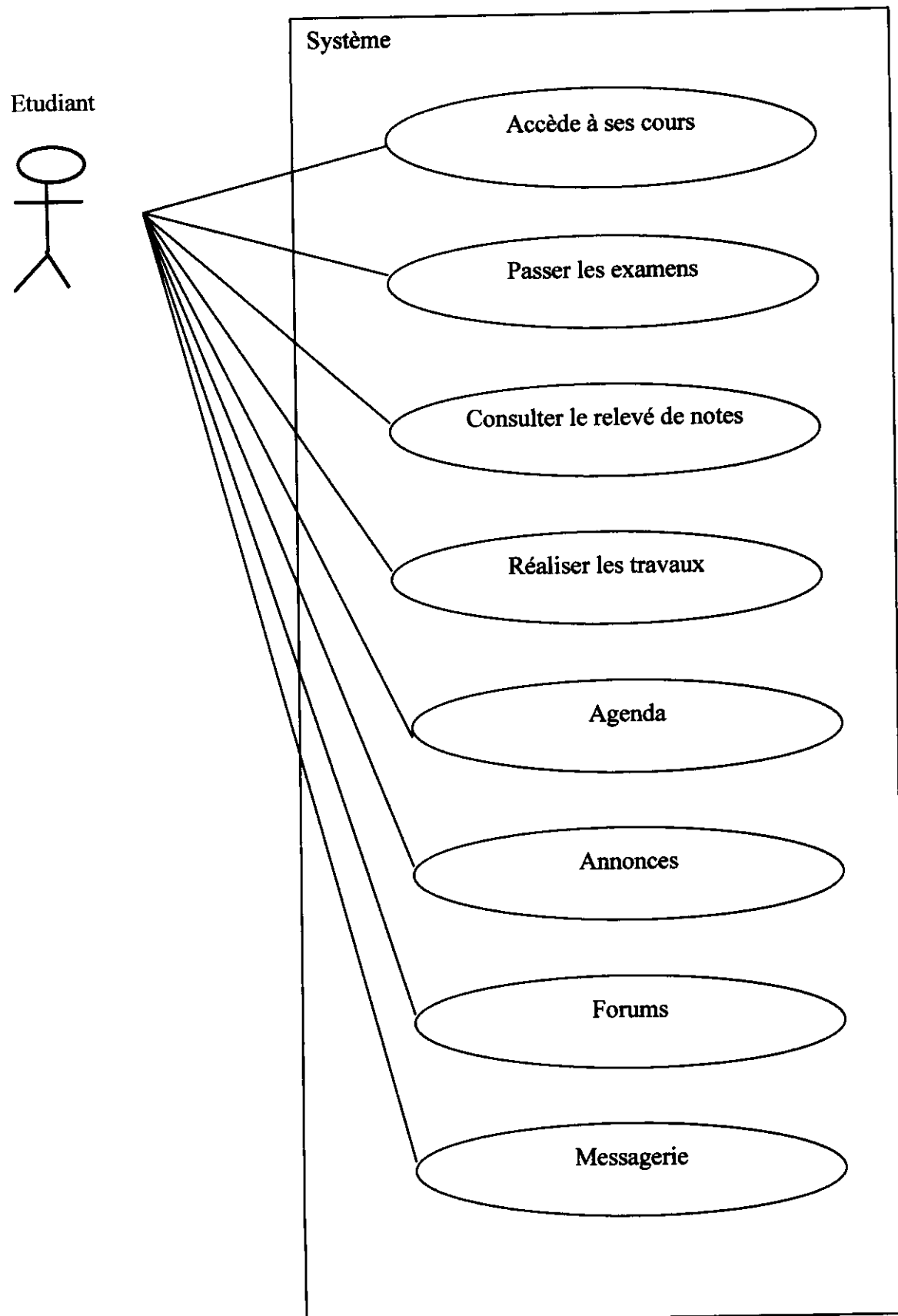
Consulter le PV des délibérations

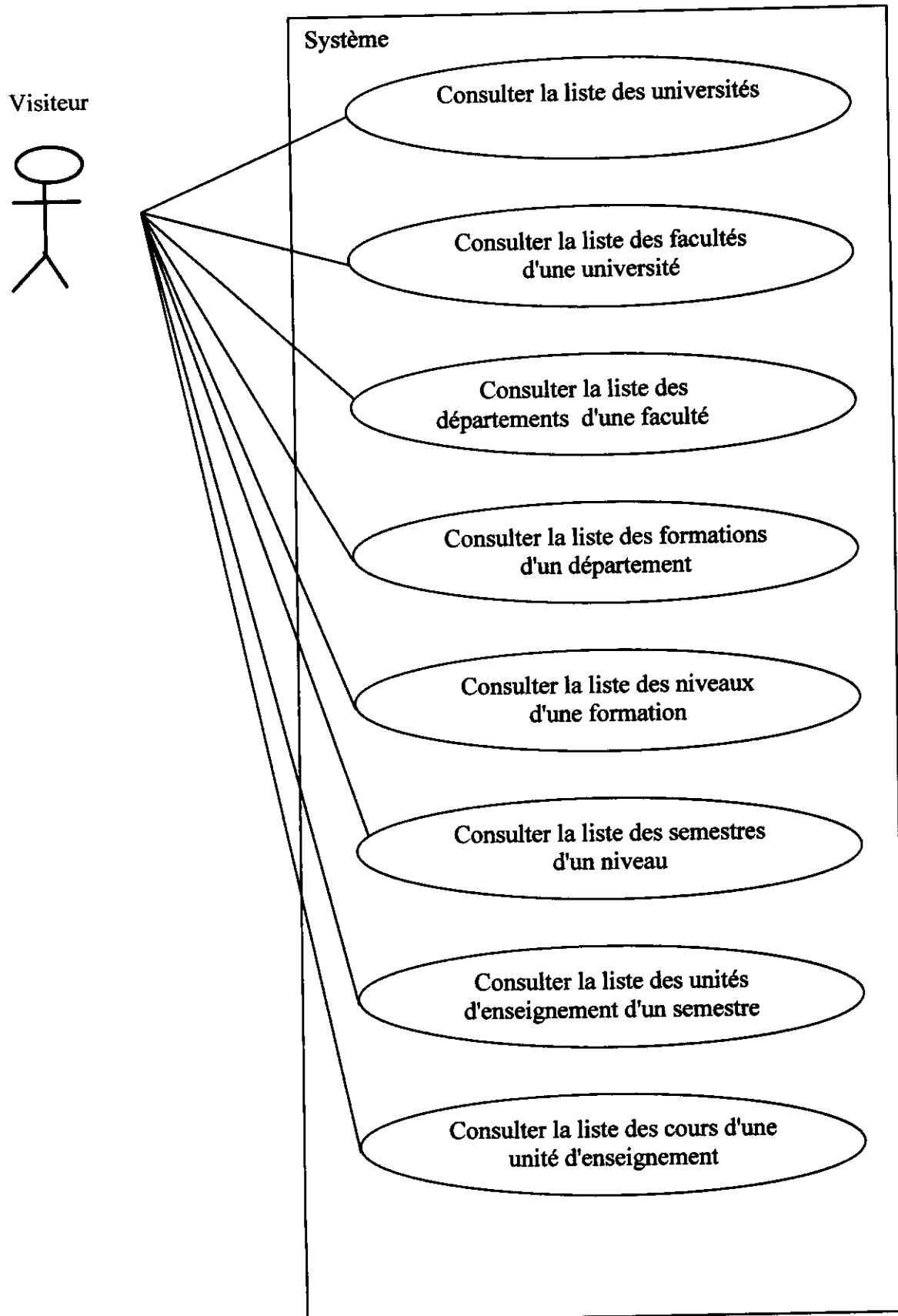
Agenda

Annonces

Forums

Messagerie





Chapitre 3 :

La conception

1. Introduction:

Pour tout système d'information, la conception tient une place prépondérante. Elle doit donc être traitée avec précision et en détail. Elle décrit comment modéliser et construire des systèmes logiciels de manière fiable et reproductible. Elle est le reflet du futur système avant même sa concrétisation, elle doit être précédée d'une étude profonde et réfléchie.

La conception s'intéresse d'abord au « COMMENT », à savoir la solution logique, pour réaliser la conception logique, UML propose la description des cas d'utilisation, les diagrammes de classes. A travers ces diagrammes nous pouvons modéliser le système.

Remarque :

La conception que nous allons présenter est notre conception et pas celle de la plate-forme «Claroline», en effet cette conception illustre notre travail à savoir l'extension de la plate-forme.

1. Description des cas d'utilisation:

On a quatre utilisateurs, chacun a plusieurs cas d'utilisation :

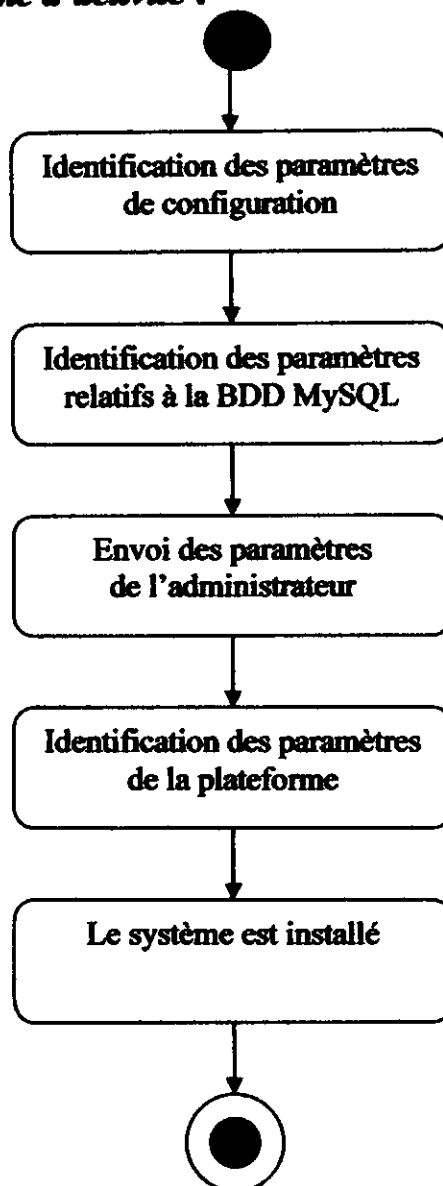
❖ L'administrateur :

1. Cas d'utilisation « Installation du système » :

1.1 Scénario:

1. L'administrateur identifie les paramètres de configuration.
2. L'administrateur identifie les paramètres relatifs à la BDD MySQL.
3. L'administrateur indique ses paramètres.
4. L'administrateur identifie les paramètres de la plateforme.

1.2 Diagramme d'activité :



2. Cas d'utilisation «Gestion des universités» :**2.1 Scénario:**

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.

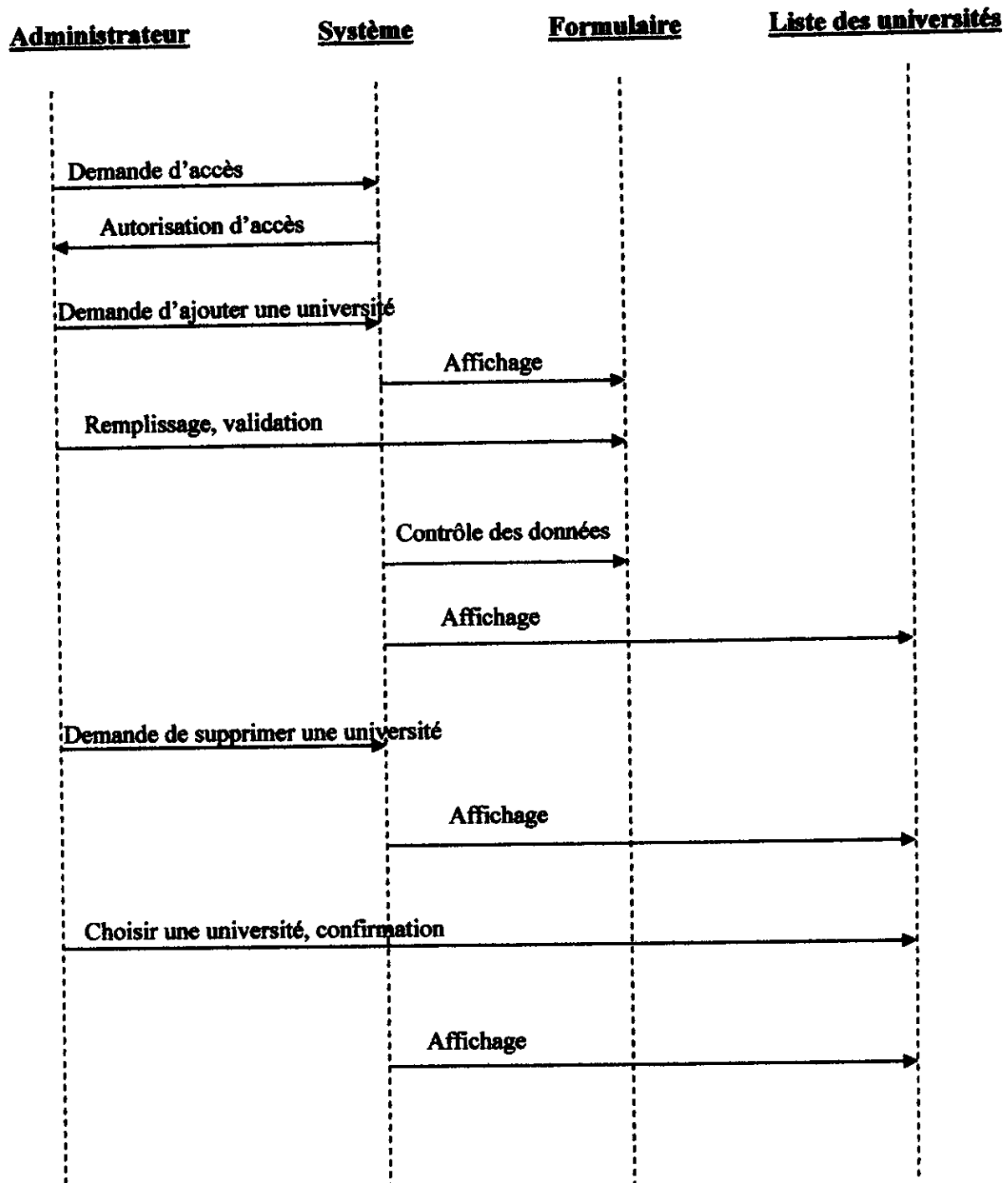
A. Ajouter une université :

4. L'administrateur demande d'ajouter une université.
5. Le système affiche un formulaire vide.
6. L'administrateur saisit les paramètres de l'université puis valide.
7. Le système contrôle les données saisies et affiche la liste des universités.

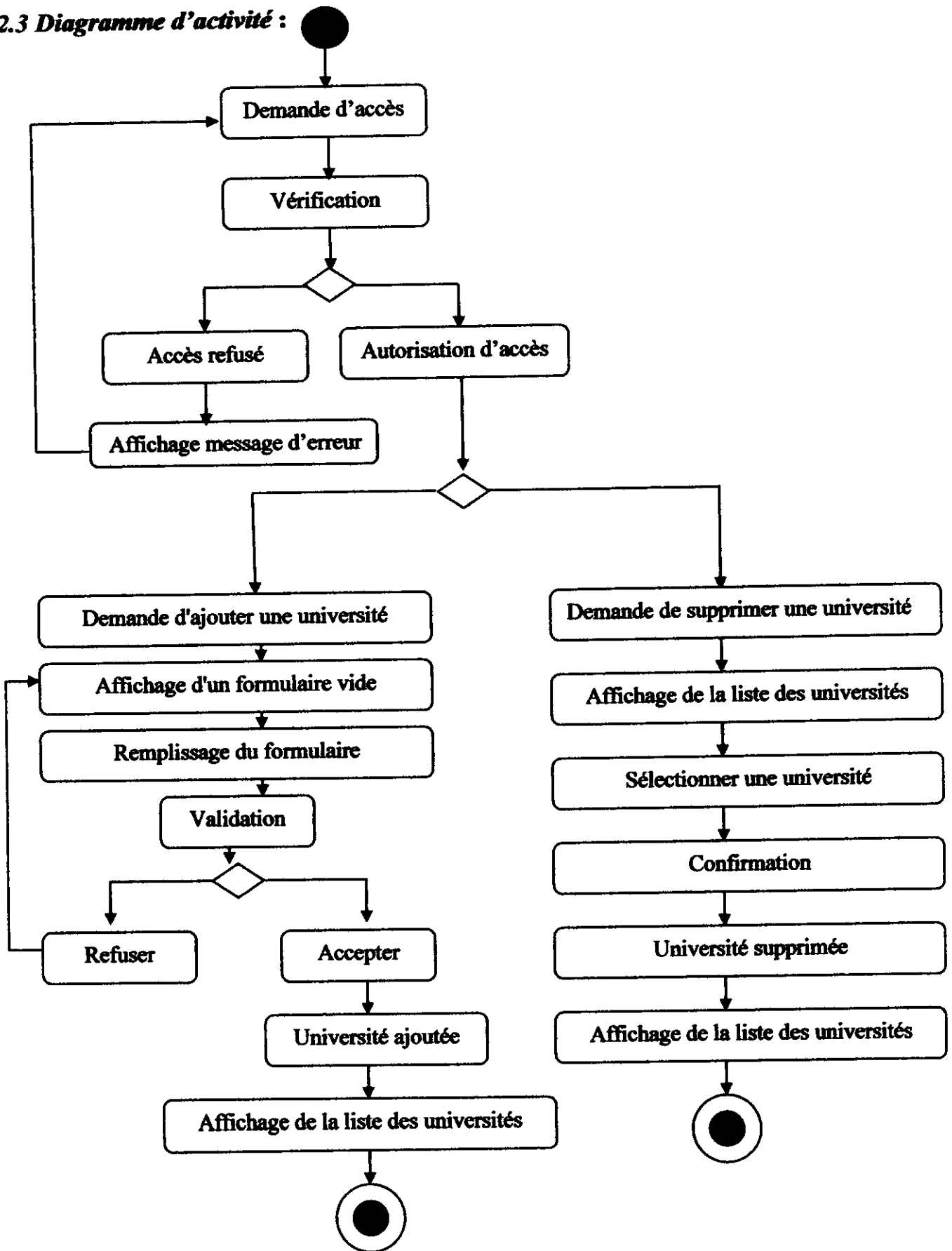
B. Supprimer une université :

4. L'administrateur demande de supprimer une université.
5. Le système affiche la liste des universités.
6. L'administrateur sélectionne une université puis confirme.
7. Le système supprime l'université et affiche la liste des universités.

2.2 Diagramme de séquence:



2.3 Diagramme d'activité :



3. Cas d'utilisation «Gestion des facultés» :

3.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.

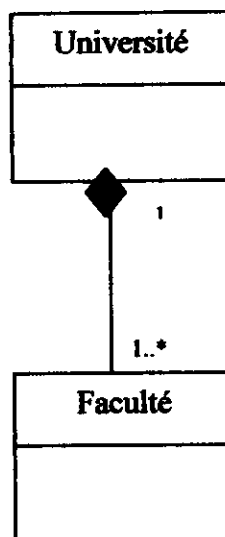
A. Ajouter une faculté :

4. L'administrateur demande d'ajouter une faculté.
5. Le système affiche un formulaire vide avec la liste des universités.
6. L'administrateur choisit une université.
7. L'administrateur saisit les paramètres de la faculté puis valide.
8. Le système contrôle les données saisies et affiche la liste des facultés.

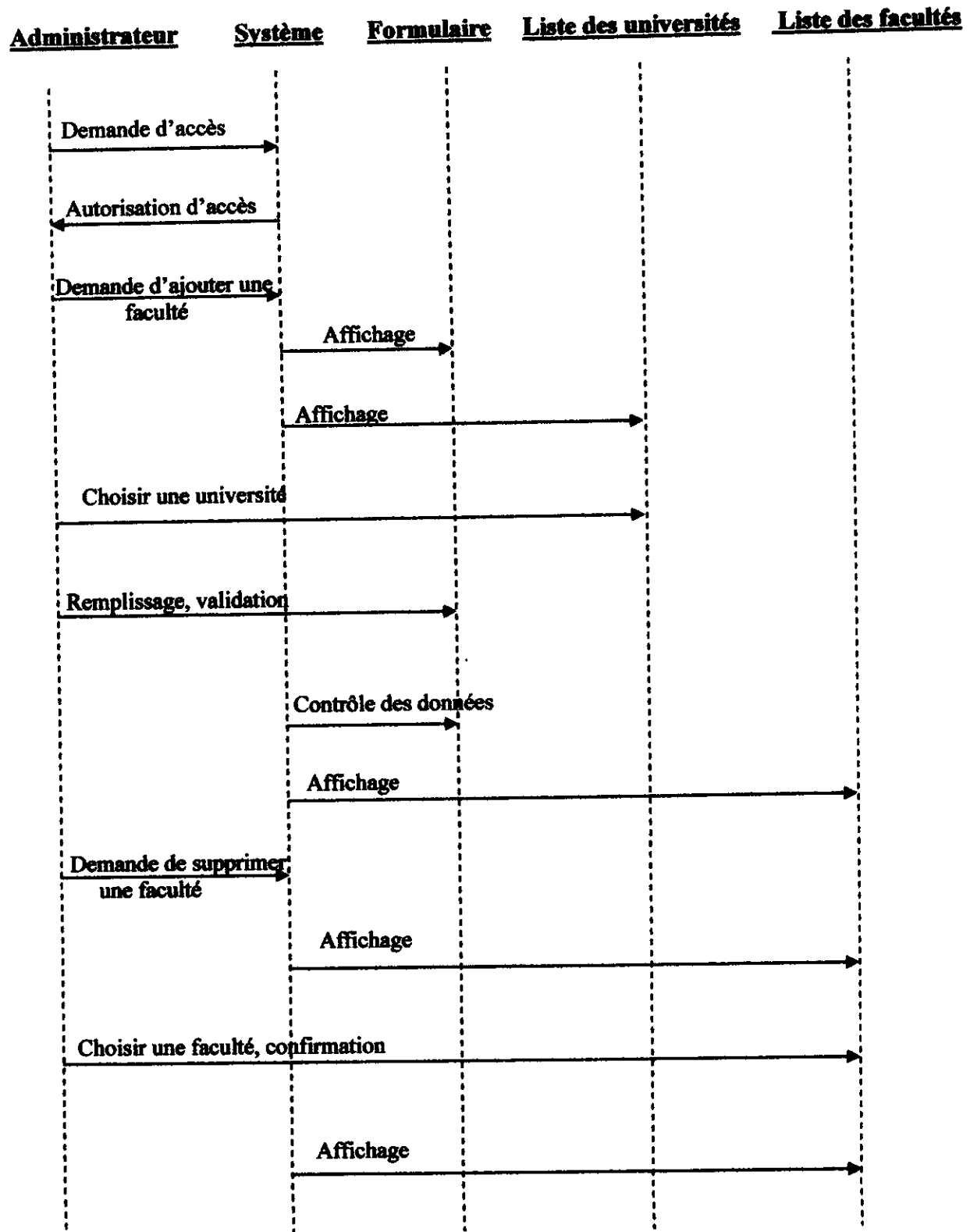
B. Supprimer une faculté :

4. L'administrateur demande de supprimer une faculté.
5. Le système affiche la liste des facultés.
6. L'administrateur sélectionne une faculté puis confirme.
7. Le système supprime la faculté et affiche la liste des facultés.

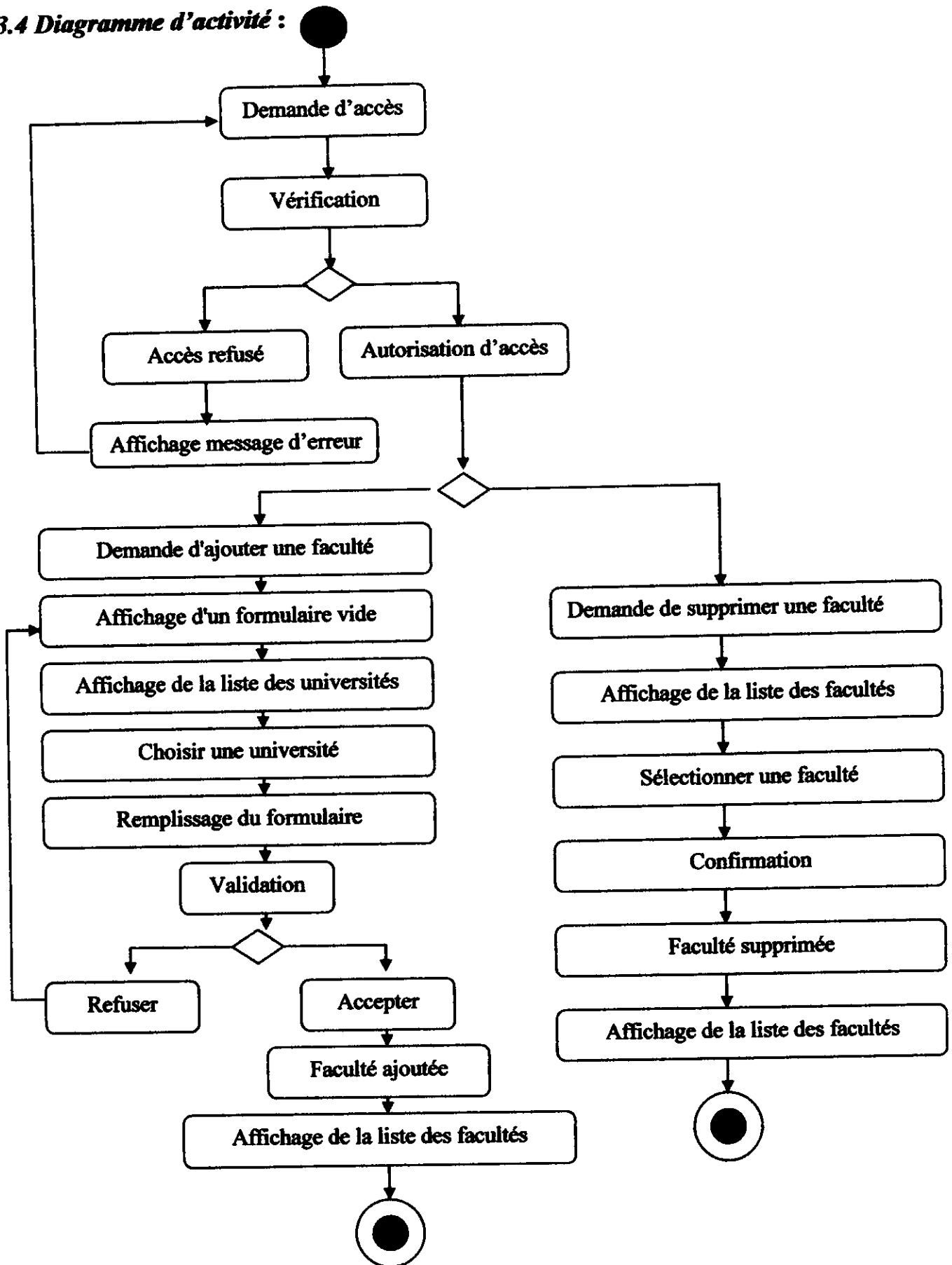
3.2 Diagramme de classe :



3.3 Diagramme de séquence:



3.4 Diagramme d'activité :



4. Cas d'utilisation «Gestion des départements» :

4.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.

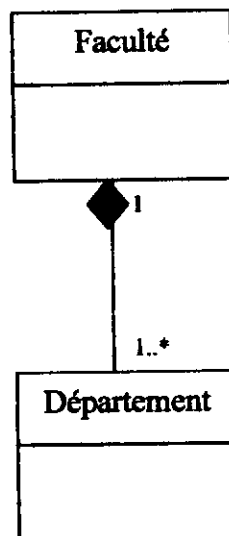
A. Ajouter un département :

4. L'administrateur demande d'ajouter un département.
5. Le système affiche un formulaire vide avec la liste des facultés.
6. L'administrateur choisit une faculté.
7. L'administrateur saisit les paramètres du département puis valide.
8. Le système contrôle les données saisies et affiche la liste des départements.

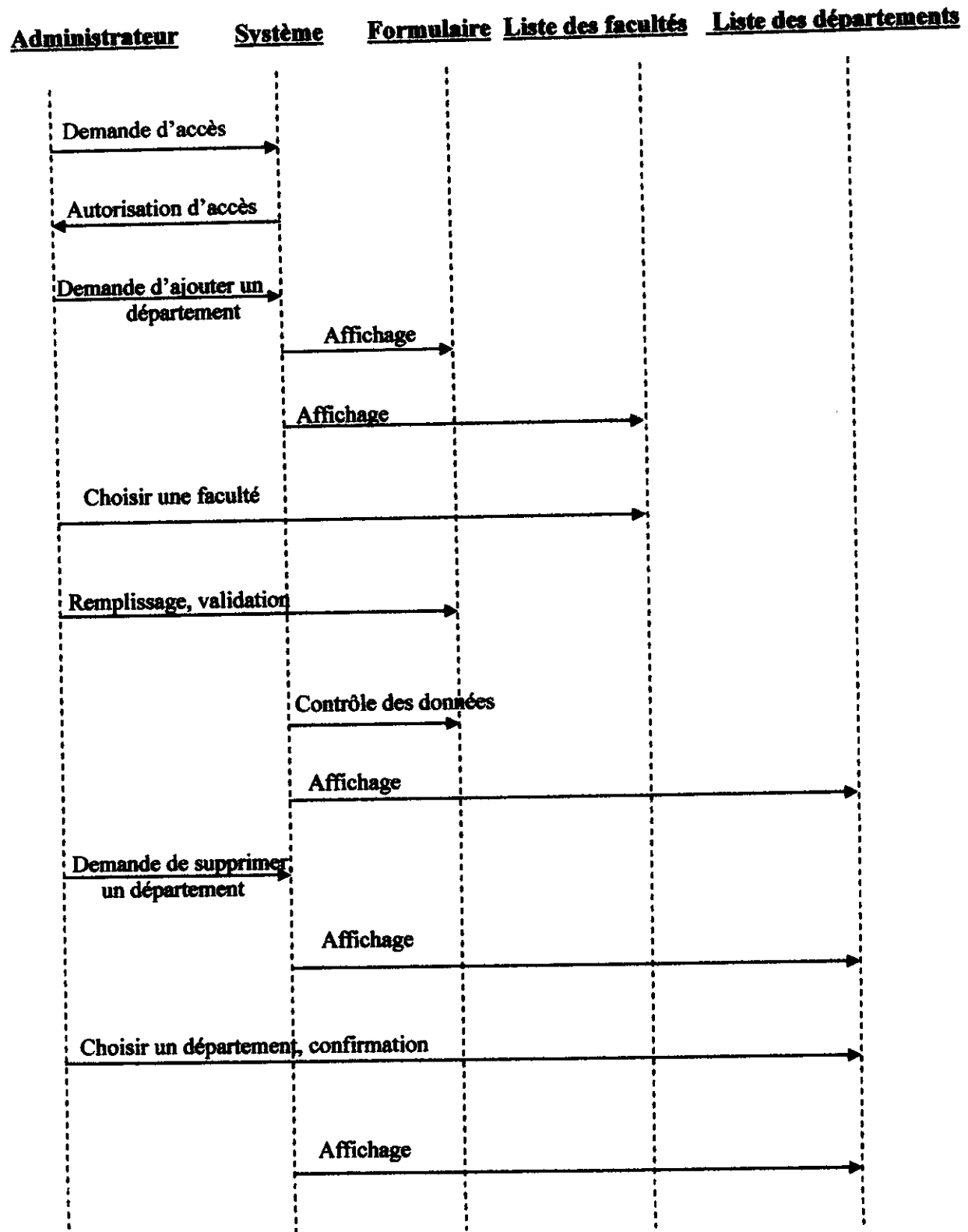
B. Supprimer un département :

4. L'administrateur demande de supprimer un département.
5. Le système affiche la liste des départements.
6. L'administrateur sélectionne un département puis confirme.
7. Le système supprime le département et affiche la liste des départements.

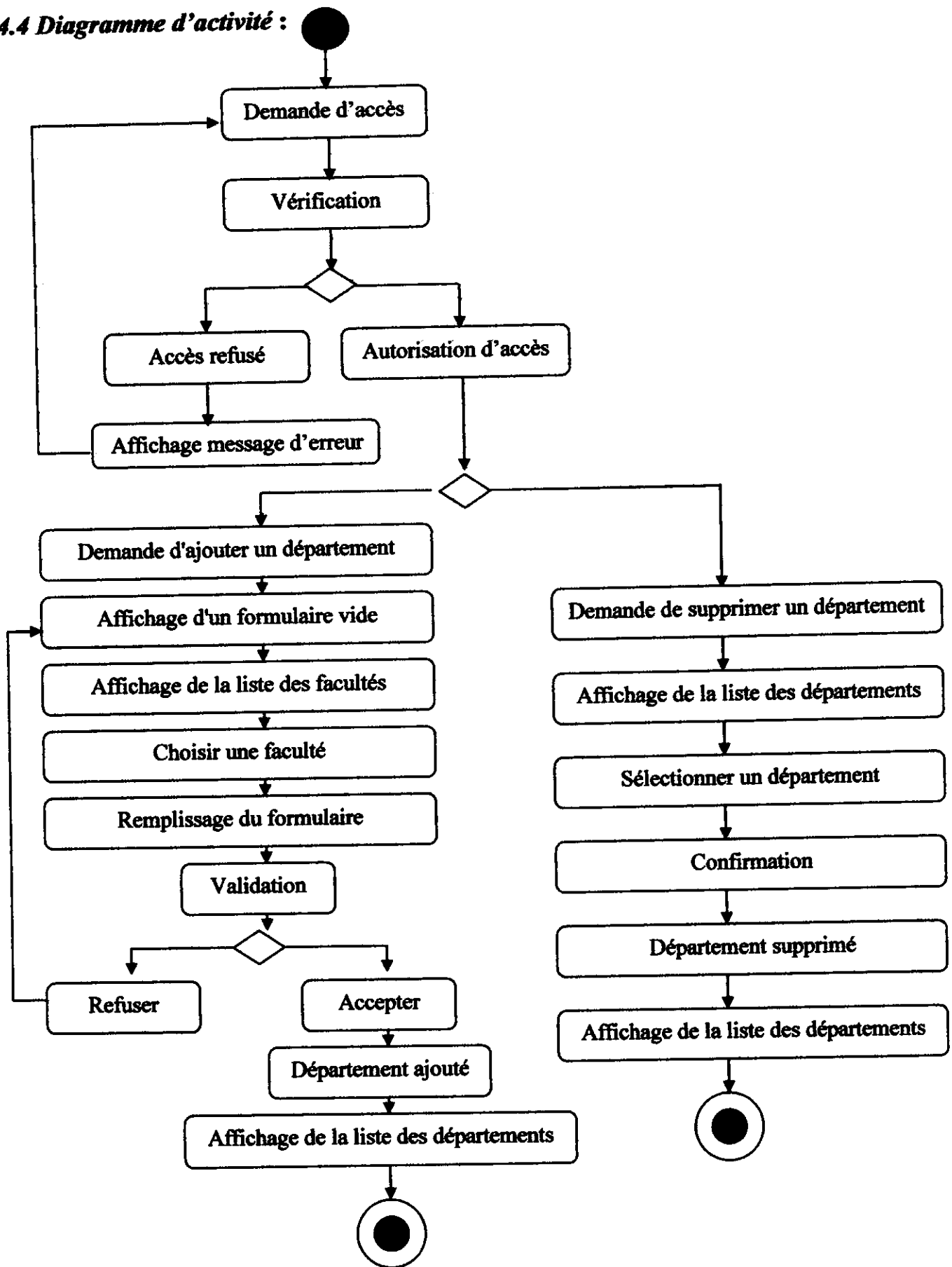
4.2 Diagramme de classe :



4.3 Diagramme de séquence:



4.4 Diagramme d'activité :



5. Cas d'utilisation «Gestion des formations» :**5.1 Scénario:**

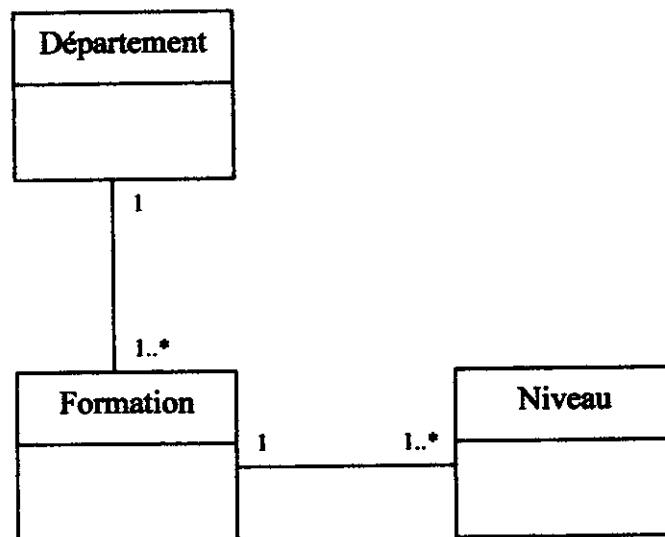
1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.

A. Ajouter une formation :

4. L'administrateur demande d'ajouter une formation.
5. Le système affiche un formulaire vide avec la liste des départements.
6. L'administrateur choisit un département.
7. L'administrateur saisit les paramètres de la formation puis valide.
8. Le système contrôle les données saisies et affiche la liste des formations.

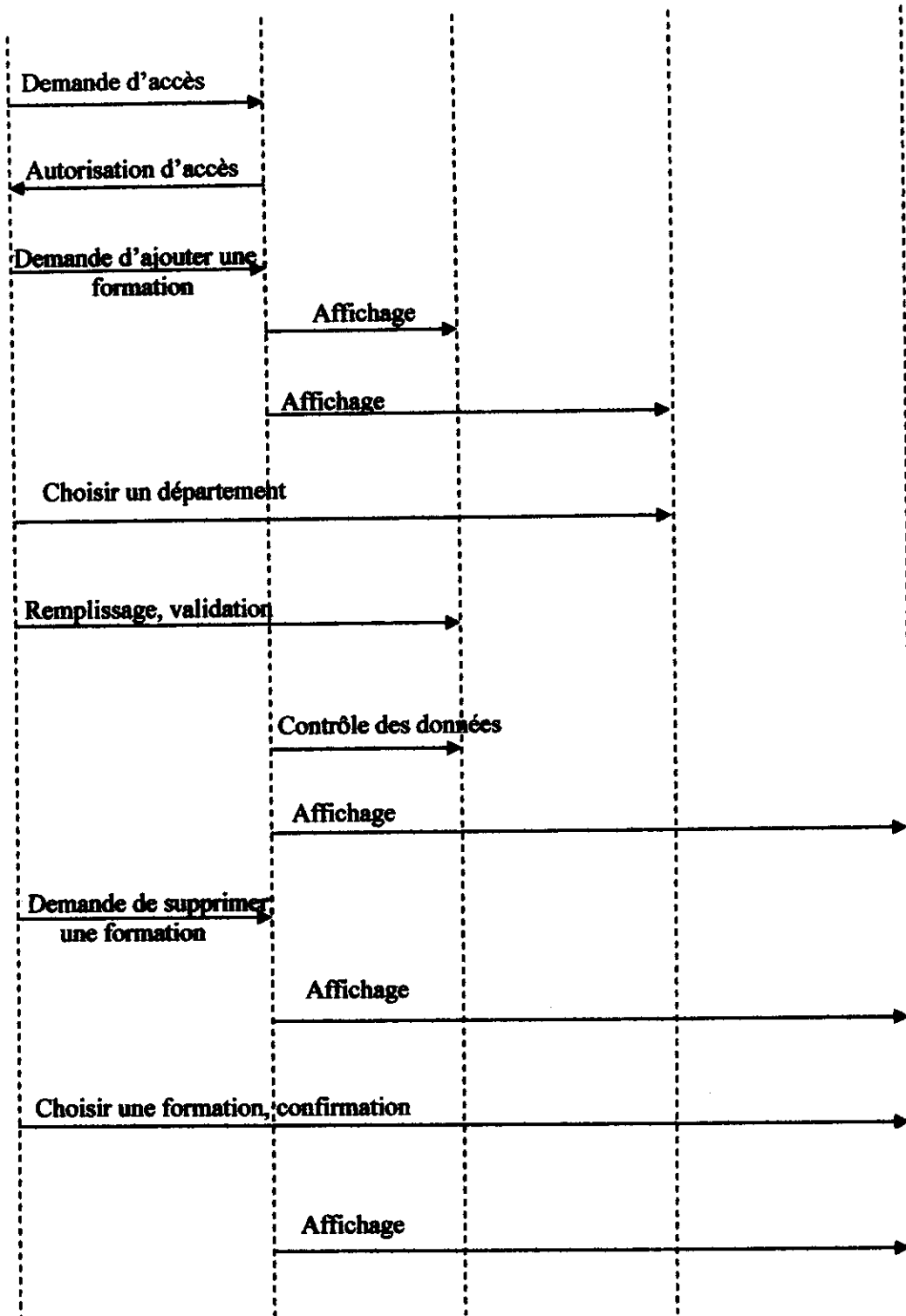
B. Supprimer une formation :

4. L'administrateur demande de supprimer une formation.
5. Le système affiche la liste des formations.
6. L'administrateur sélectionne une formation puis confirme.
7. Le système supprime la formation et affiche la liste des formations.

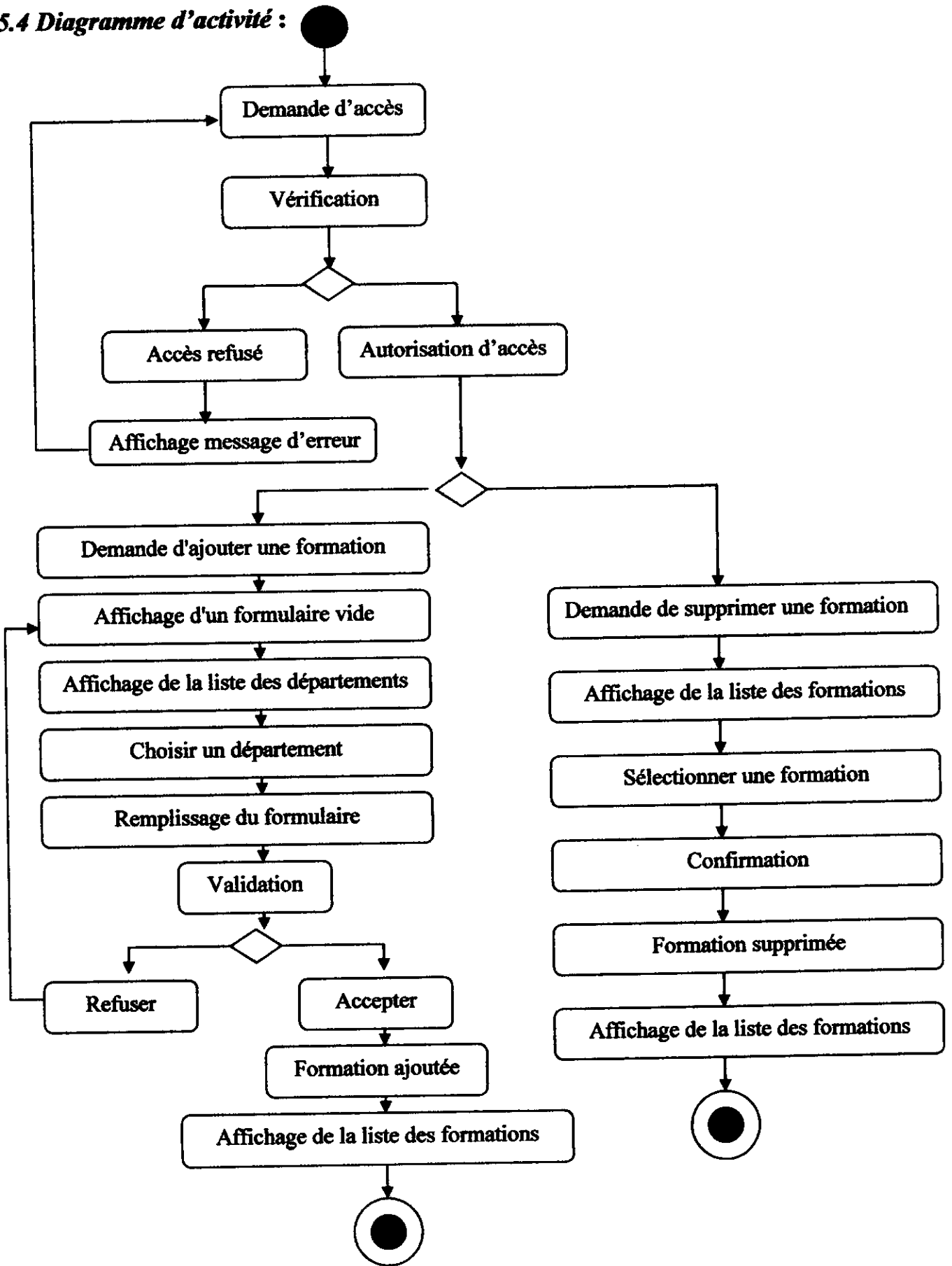
5.2 Diagramme de classe :

5.3 Diagramme de séquence:

Administrateur Système Formulaire Liste des départements Liste des formations



5.4 Diagramme d'activité :



6. Cas d'utilisation «Gestion des unités d'enseignement» :

6.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.

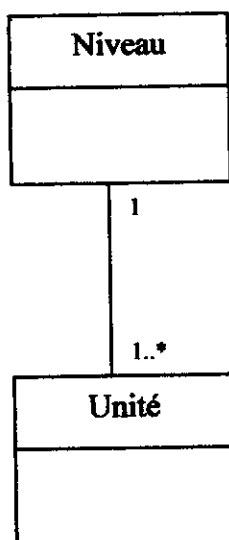
A. Ajouter une unité d'enseignement :

4. L'administrateur demande d'ajouter une unité d'enseignement.
5. Le système affiche un formulaire vide avec la liste des formations.
6. L'administrateur choisit une formation.
7. L'administrateur saisit les paramètres de l'unité d'enseignement puis valide.
8. Le système contrôle les données saisies et affiche la liste des unités d'enseignement.

B. Supprimer une unité d'enseignement :

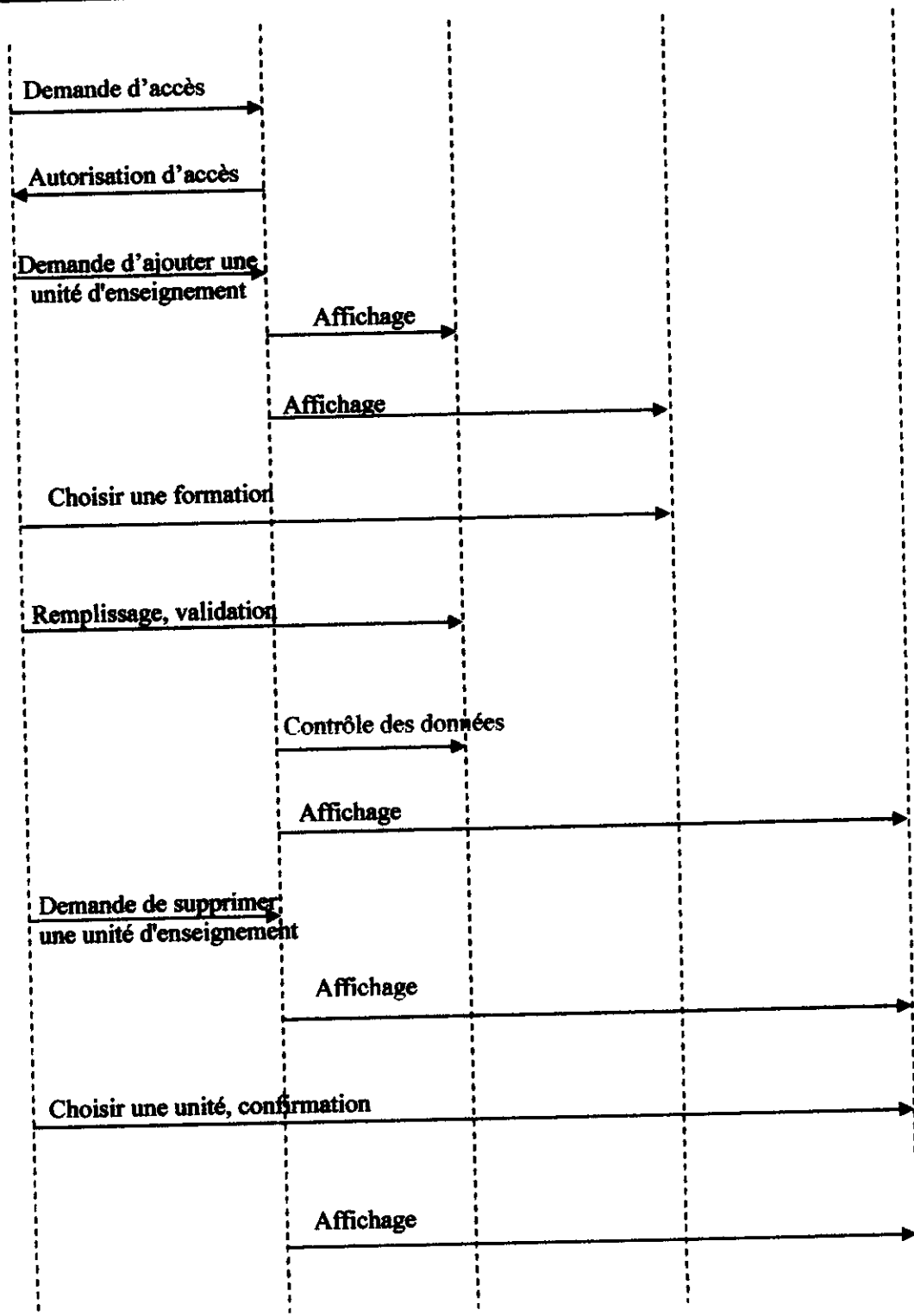
4. L'administrateur demande de supprimer une unité d'enseignement.
5. Le système affiche la liste des unités d'enseignement.
6. L'administrateur sélectionne une unité d'enseignement puis confirme.
7. Le système supprime l'unité d'enseignement et affiche la liste des unités d'enseignement.

6.2 Diagramme de classe :

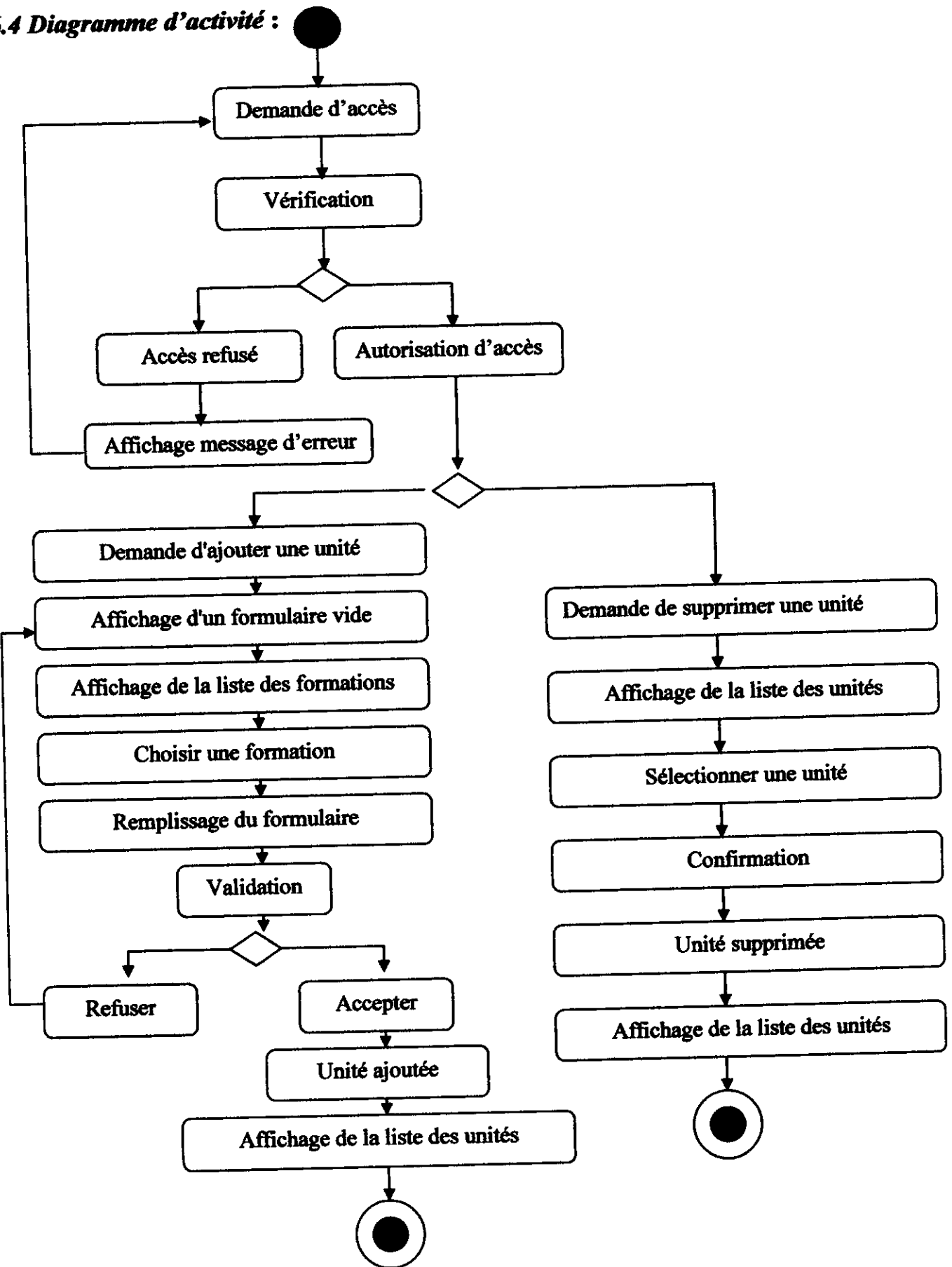


6.3 Diagramme de séquence:

Administrateur Système Formulaire Liste des formations Liste des unités



6.4 Diagramme d'activité :



7. Cas d'utilisation «Gestion des cours» :

7.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.

A. Ajouter un cours:

4. L'administrateur demande d'ajouter un cours.
5. Le système affiche un formulaire vide avec la liste des unités d'enseignement.
6. L'administrateur choisit une unité d'enseignement.
7. L'administrateur saisit les paramètres du cours puis valide.
8. Le système contrôle les données saisies et affiche la liste des cours.

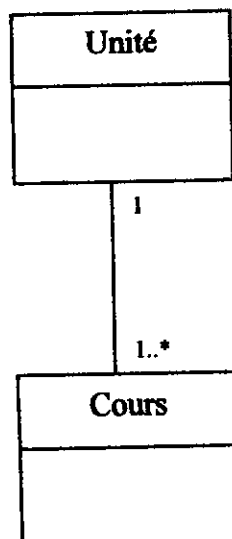
B. Supprimer un cours:

4. L'administrateur demande de supprimer un cours.
5. Le système affiche la liste des cours.
6. L'administrateur sélectionne un cours puis confirme.
7. Le système supprime le cours et affiche la liste des cours.

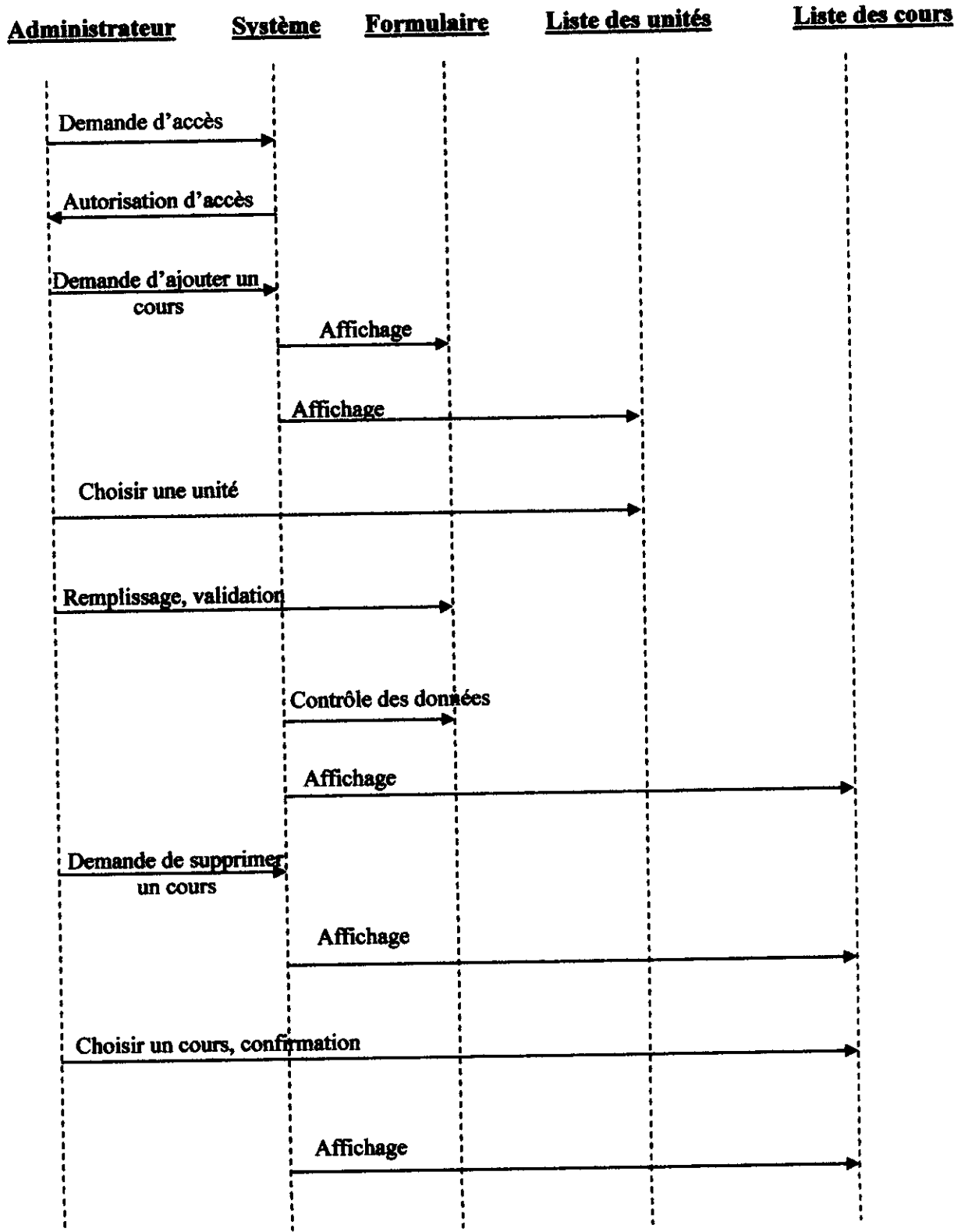
Remarque:

L'administrateur identifie les cours qui vont être enseignés dans un niveau, il décide aussi si un cours possède un TD ou un TP mais c'est à l'enseignant de créer le cours (contenu) et l'administrateur peut accéder à tous les cours.

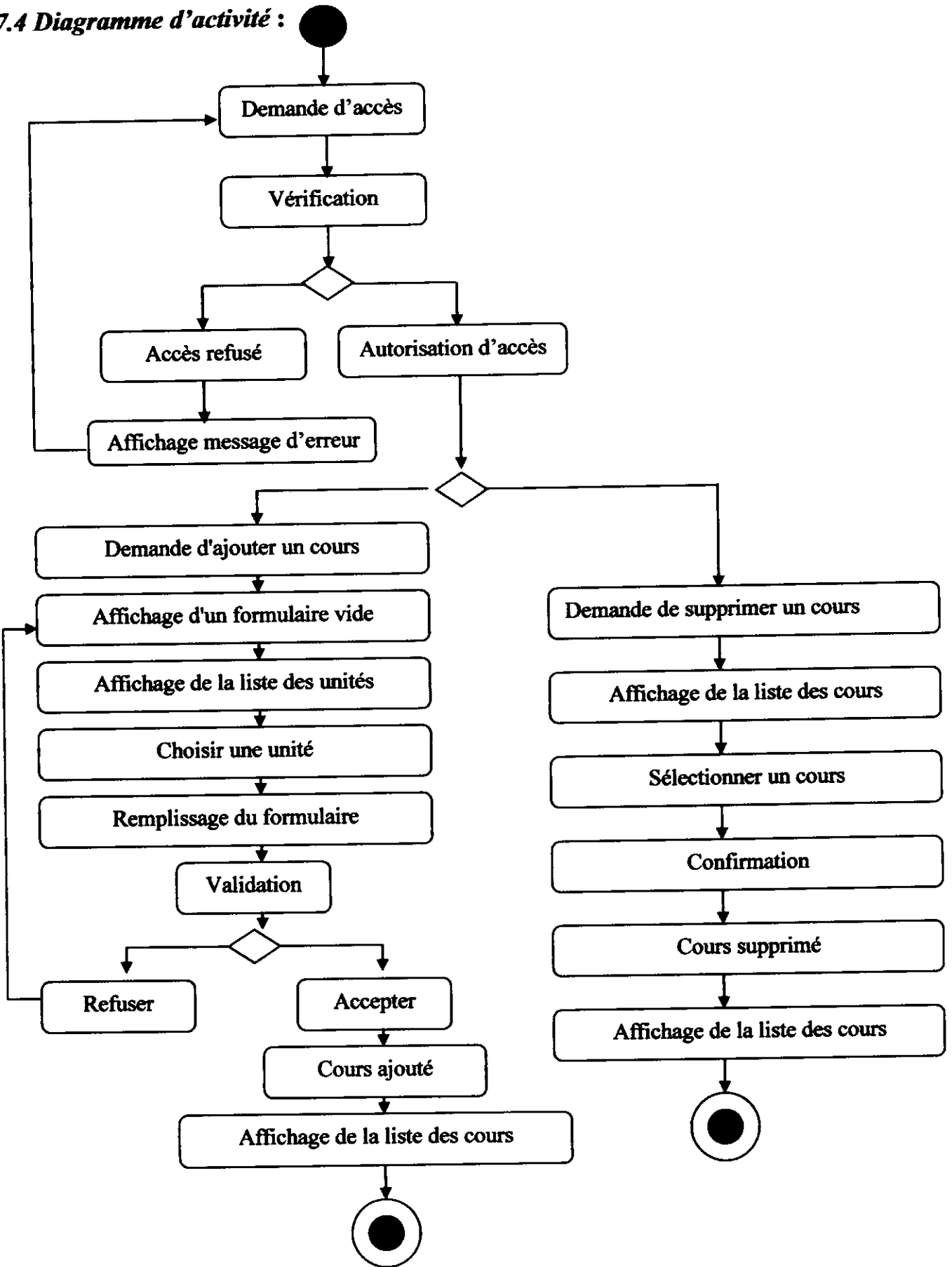
7.2 Diagramme de classe :



7.3 Diagramme de séquence:



7.4 Diagramme d'activité :



8. Cas d'utilisation «Gestion des enseignants» :**8.1 Scénario:**

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.

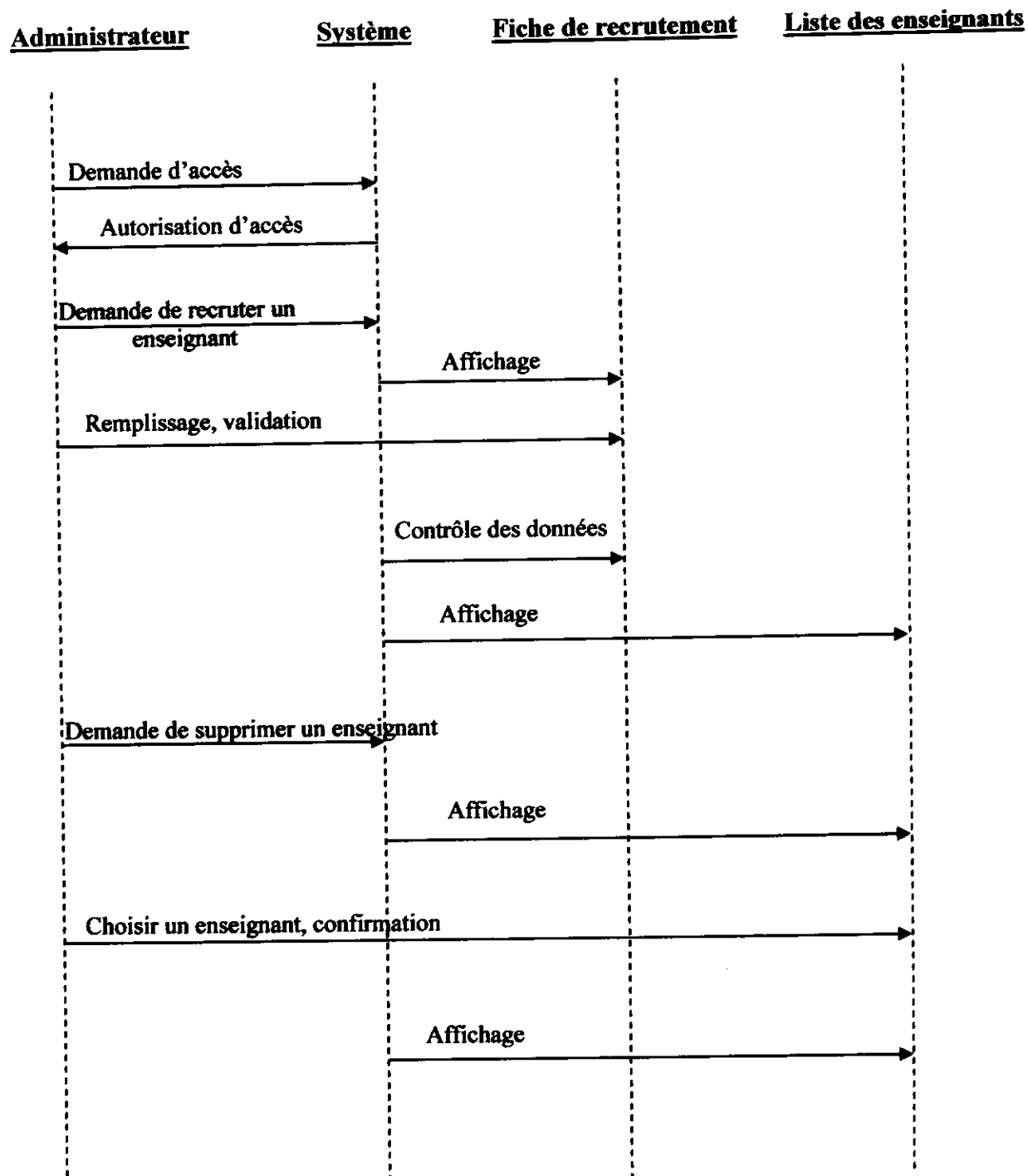
A. Ajouter un enseignant:

4. L'administrateur demande la fiche de recrutement des enseignants.
5. Le système affiche un formulaire vide.
6. L'administrateur remplit la fiche puis valide.
7. Le système contrôle les données saisies et affiche la liste des enseignants.

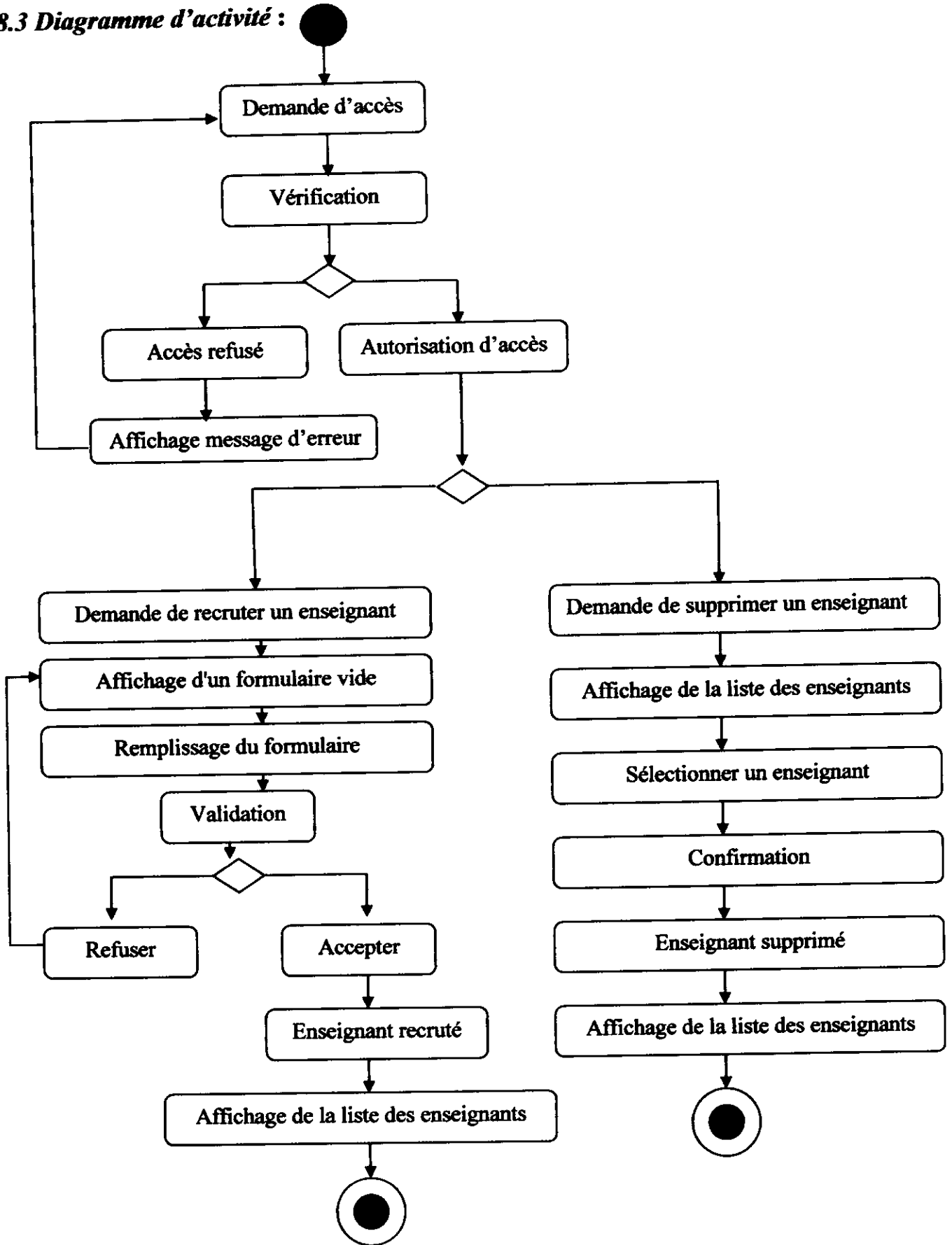
B. Supprimer un enseignant:

4. L'administrateur demande de supprimer un enseignant.
5. Le système affiche la liste des enseignants.
6. L'administrateur sélectionne un enseignant puis confirme.
7. Le système supprime l'enseignant et affiche la liste des enseignants.

8.2 Diagramme de séquence:



8.3 Diagramme d'activité :



9. Cas d'utilisation «Gestion des étudiants» :

9.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.

A. Ajouter un étudiant:

4. L'administrateur demande d'ajouter un étudiant.
5. Le système affiche un formulaire vide.
6. L'administrateur saisit les paramètres de l'étudiant puis valide.
7. Le système contrôle les données saisies et affiche la liste des étudiants.

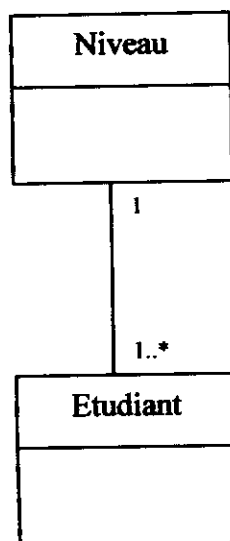
B. Ajouter un groupe d'étudiants à la fois à partir d'un fichier CSV:

4. L'administrateur demande d'ajouter un groupe d'étudiants à la fois.
5. Le système donne l'accès pour chercher le fichier.
6. L'administrateur sélectionne le fichier puis valide.
7. Le système contrôle les données du fichier et affiche la liste des étudiants.

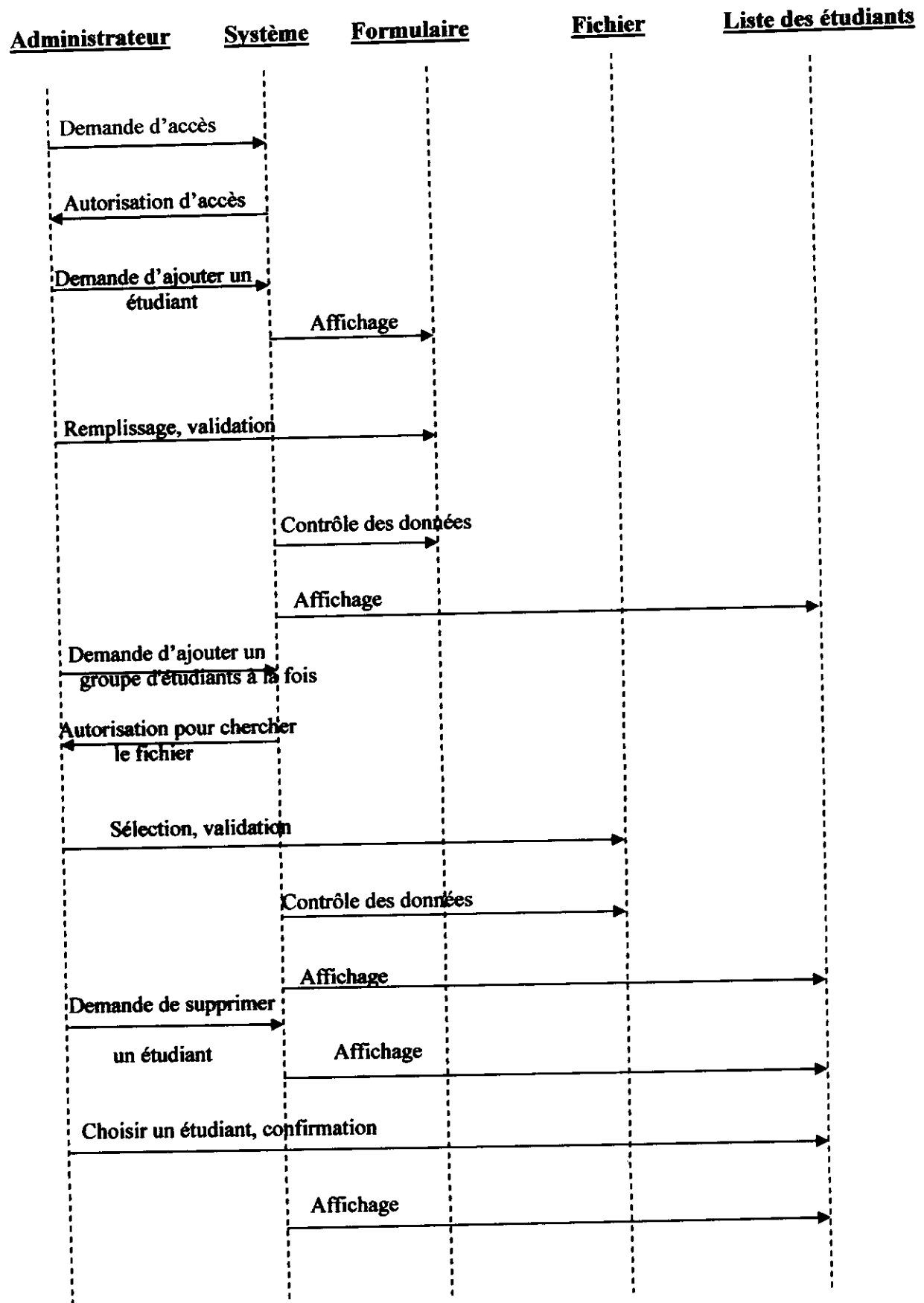
C. Supprimer un étudiant:

4. L'administrateur demande de supprimer un étudiant.
5. Le système affiche la liste des étudiants.
6. L'administrateur sélectionne un étudiant puis confirme.
7. Le système supprime l'étudiant et affiche la liste des étudiants.

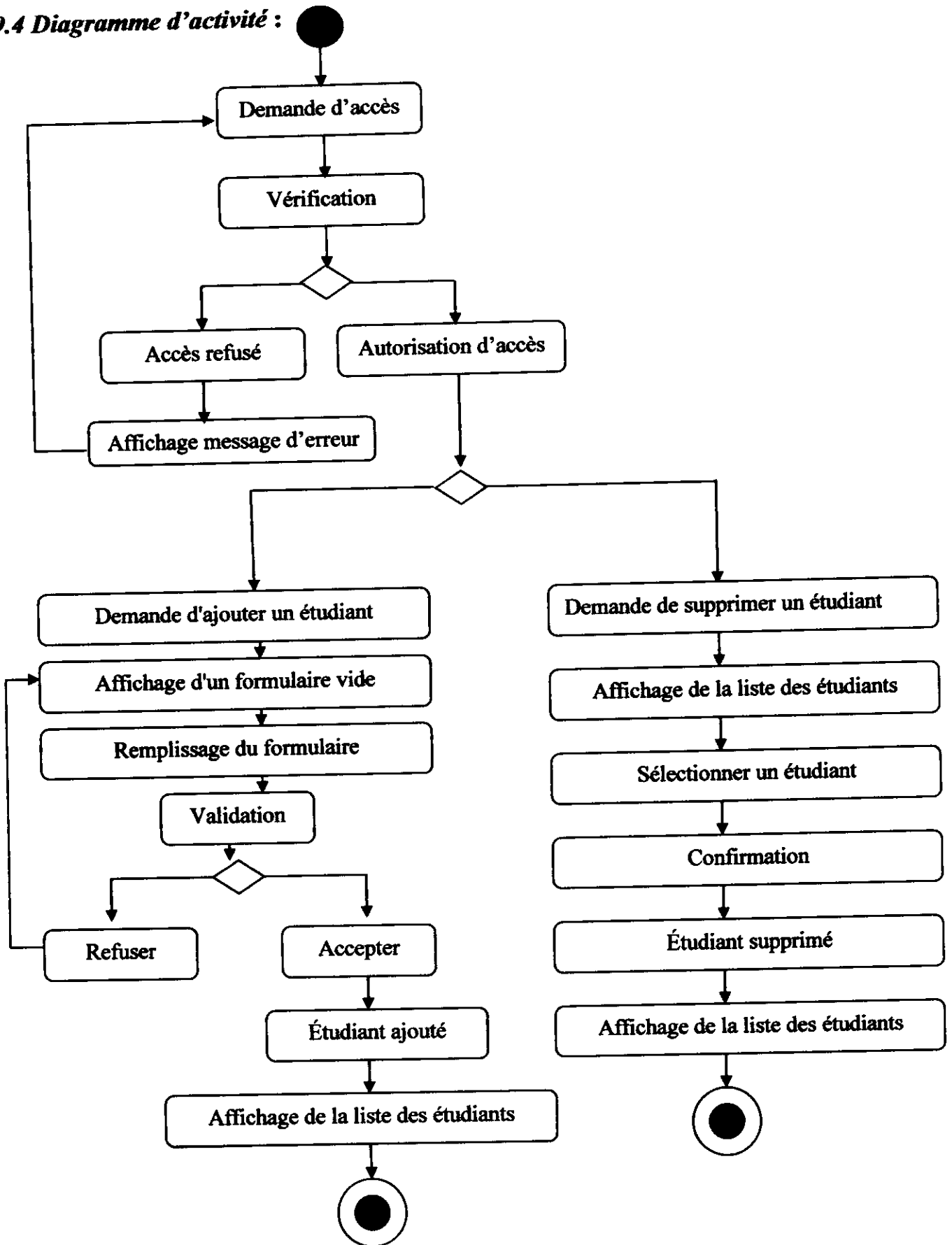
9.2 Diagramme de classe :

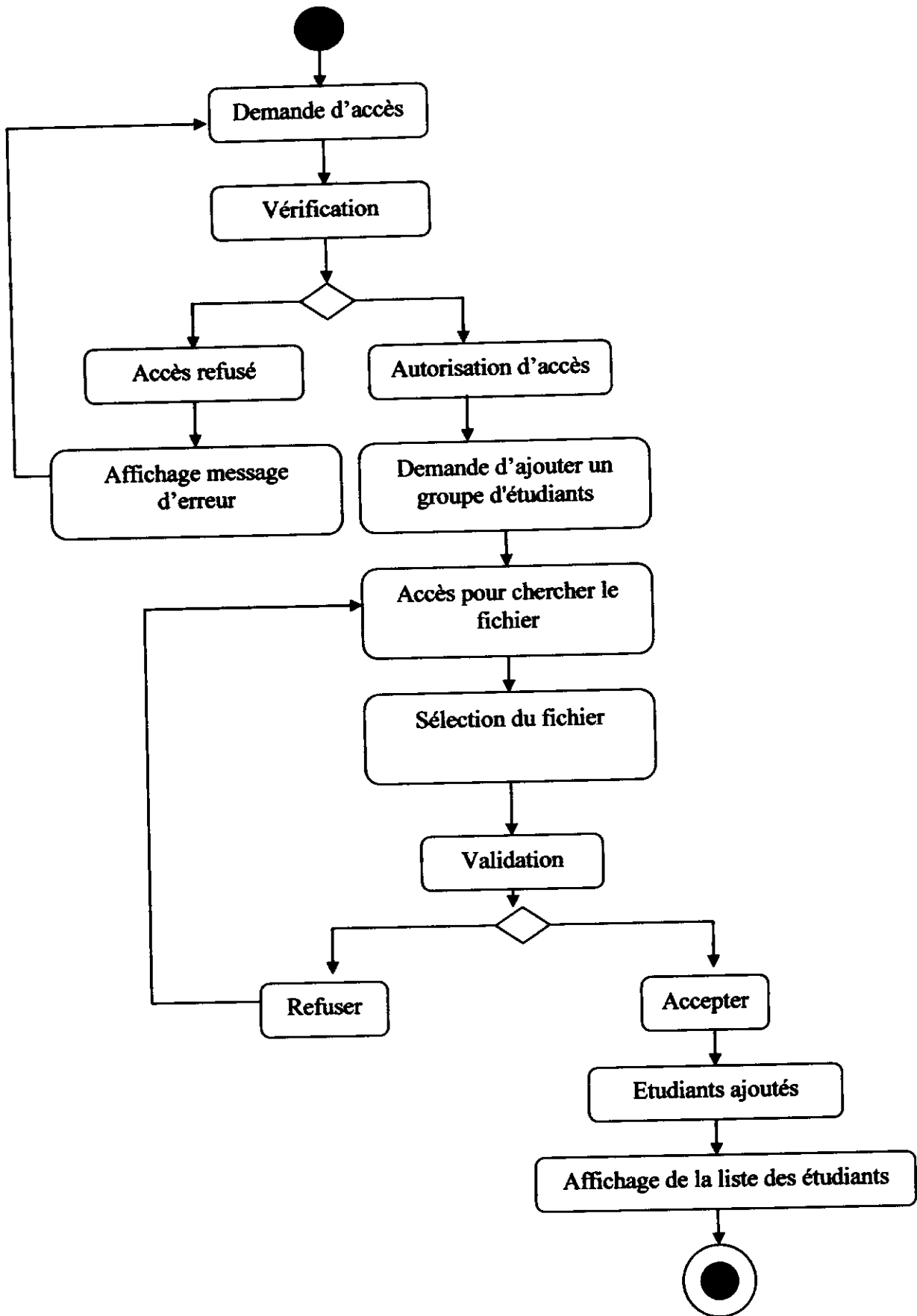


9.3 Diagramme de séquence:



9.4 Diagramme d'activité :



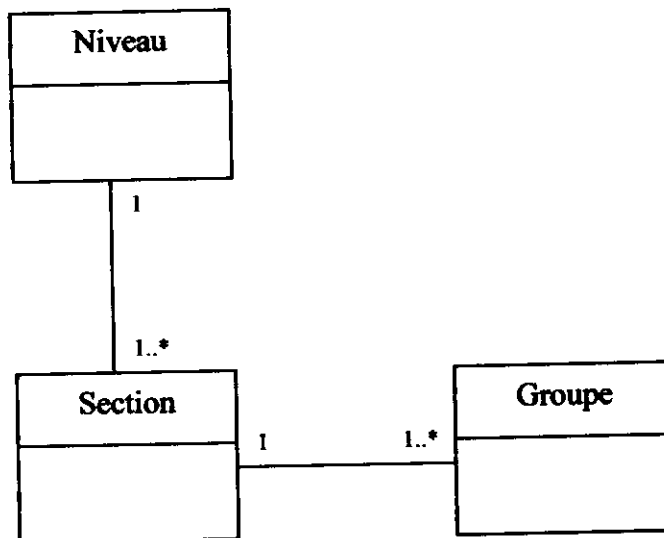


10. Cas d'utilisation «Déterminer le nombre de sections et de groupes par niveau » :

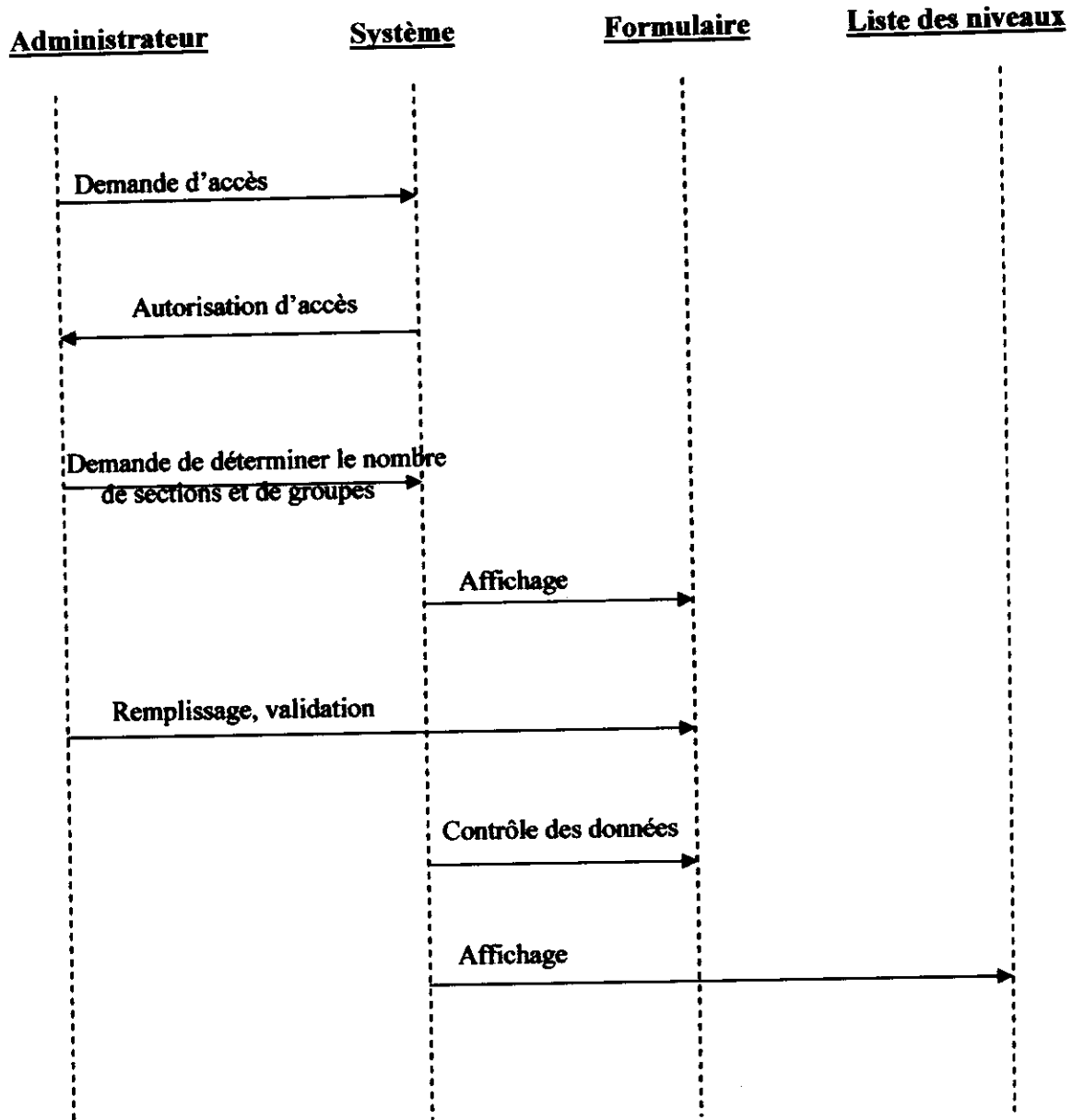
10.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande de déterminer le nombre de sections et de groupes par niveau.
5. Le système affiche la liste des niveaux.
6. L'administrateur saisit les chiffres, puis valide.
7. Le système contrôle les données saisies et affiche la liste des niveaux avec le nombre de sections et de groupes par niveau.

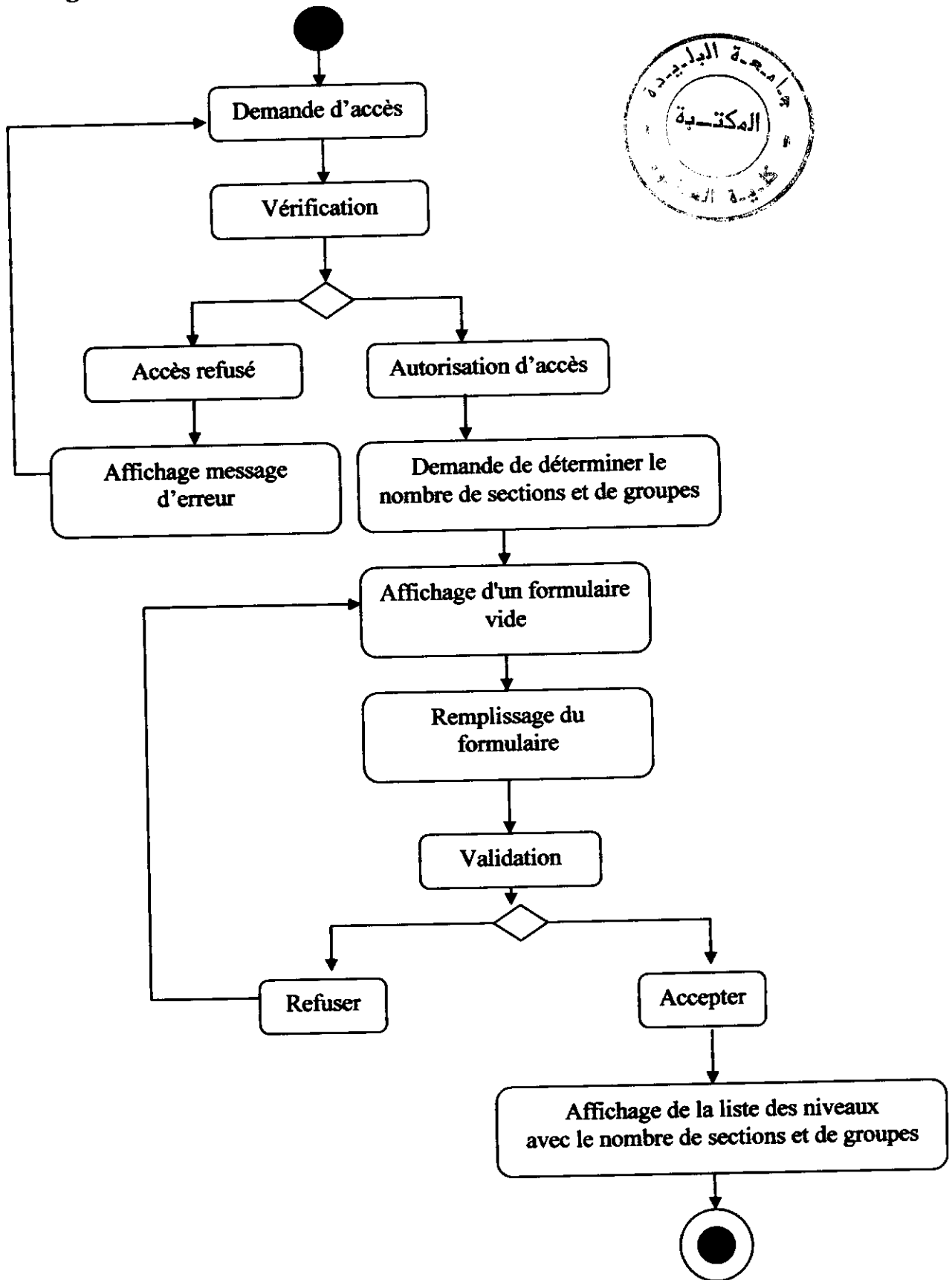
10.2 Diagramme de classe :



10.3 Diagramme de séquence:



10.4 Diagramme d'activité :

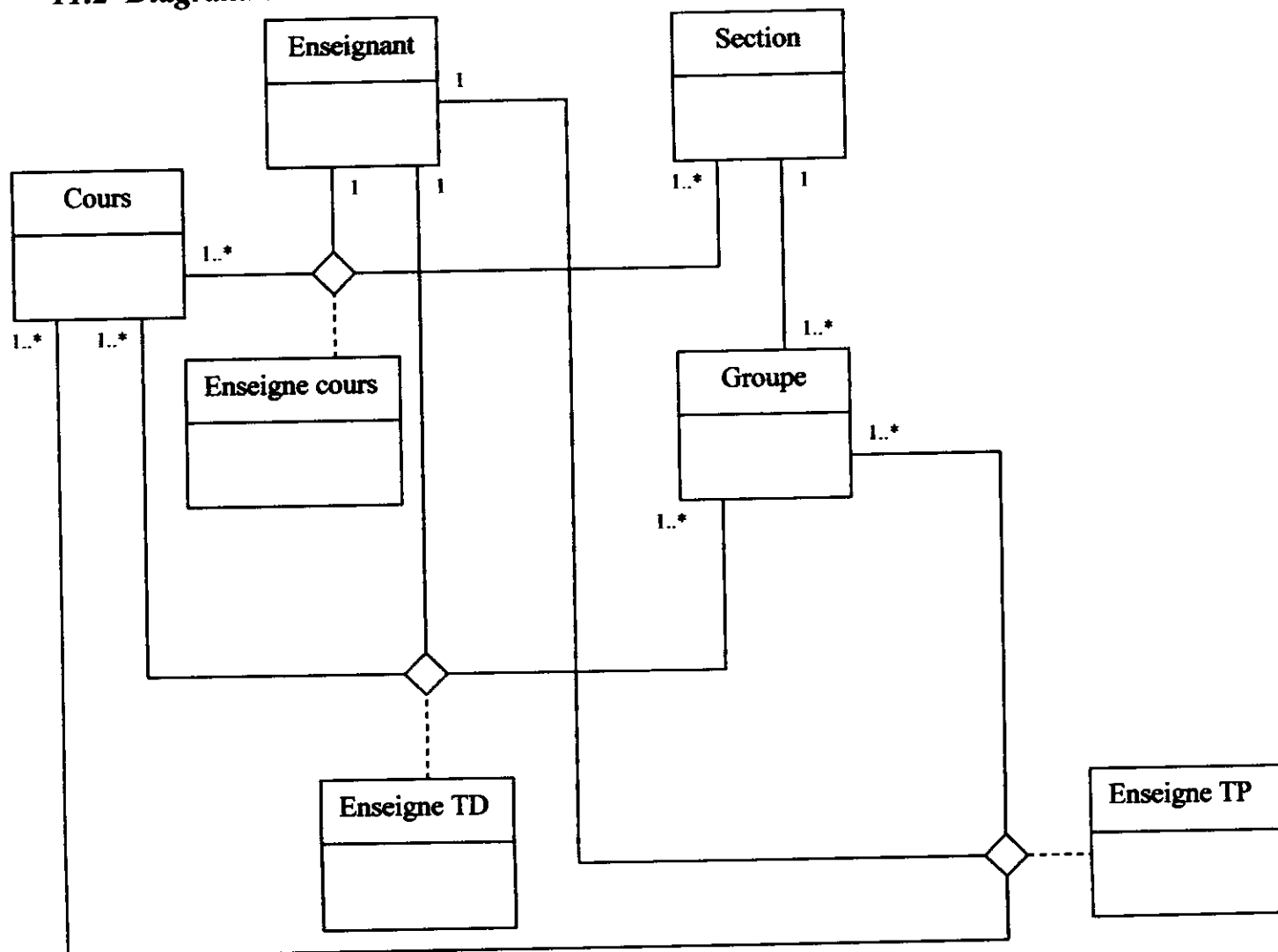


11. Cas d'utilisation «Affectation des enseignants aux cours» :

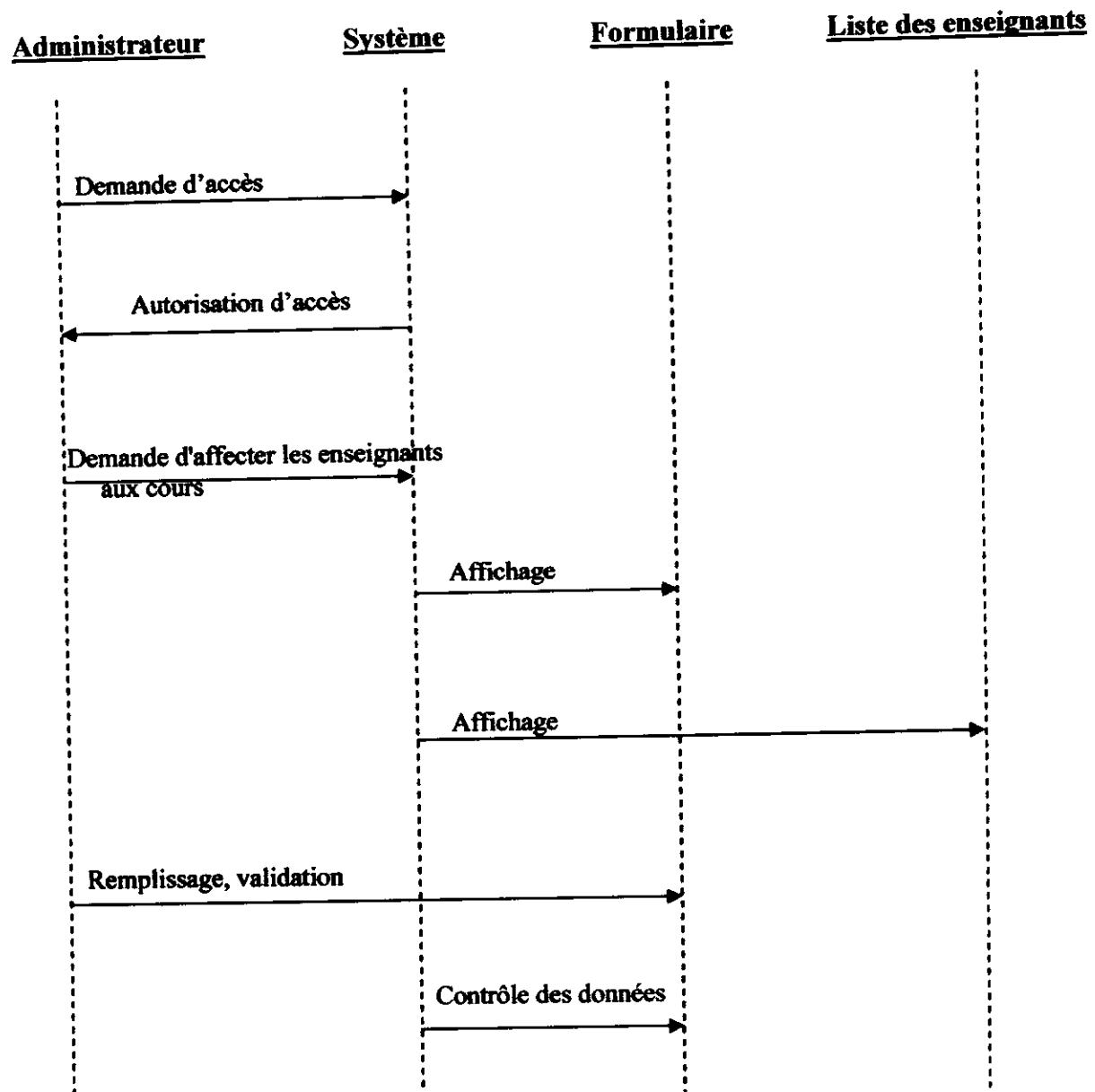
11.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande la fiche d'affectation des enseignants aux cours.
5. Le système affiche la fiche avec la liste des enseignants.
6. L'administrateur remplit la fiche en affectant un enseignant à un cours pour chaque section et un enseignant à un TD ou TP pour chaque groupe, puis valide.
7. Le système contrôle les données saisies et affiche la liste des enseignants affectés à leurs cours.

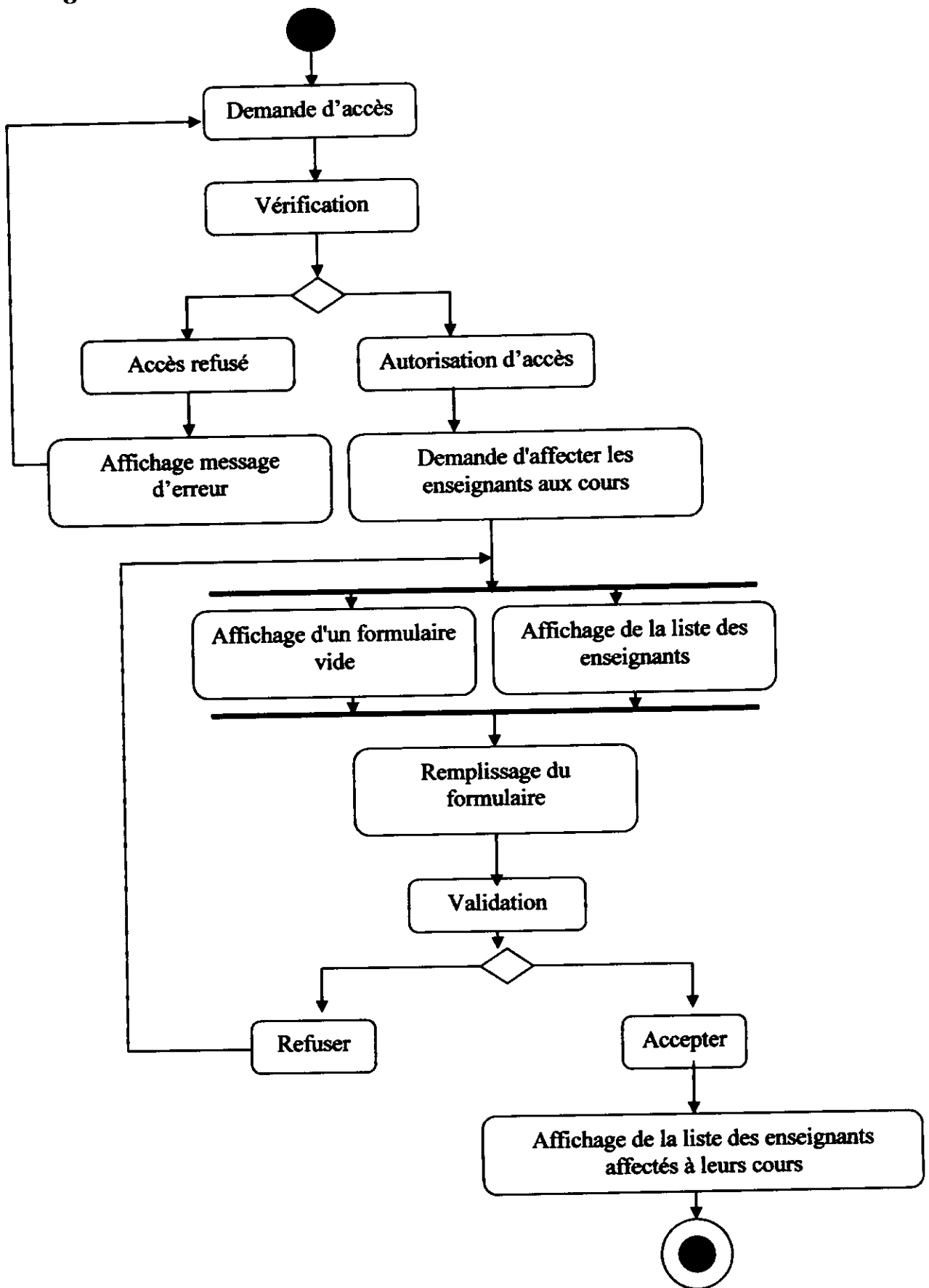
11.2 Diagramme de classe :



11.3 Diagramme de séquence:



11.4 Diagramme d'activité :

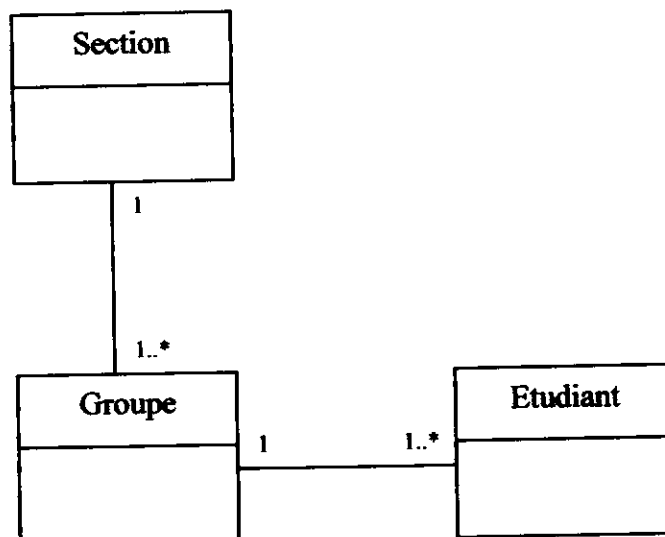


12. Cas d'utilisation «Affectation des étudiants aux sections et aux groupes» :

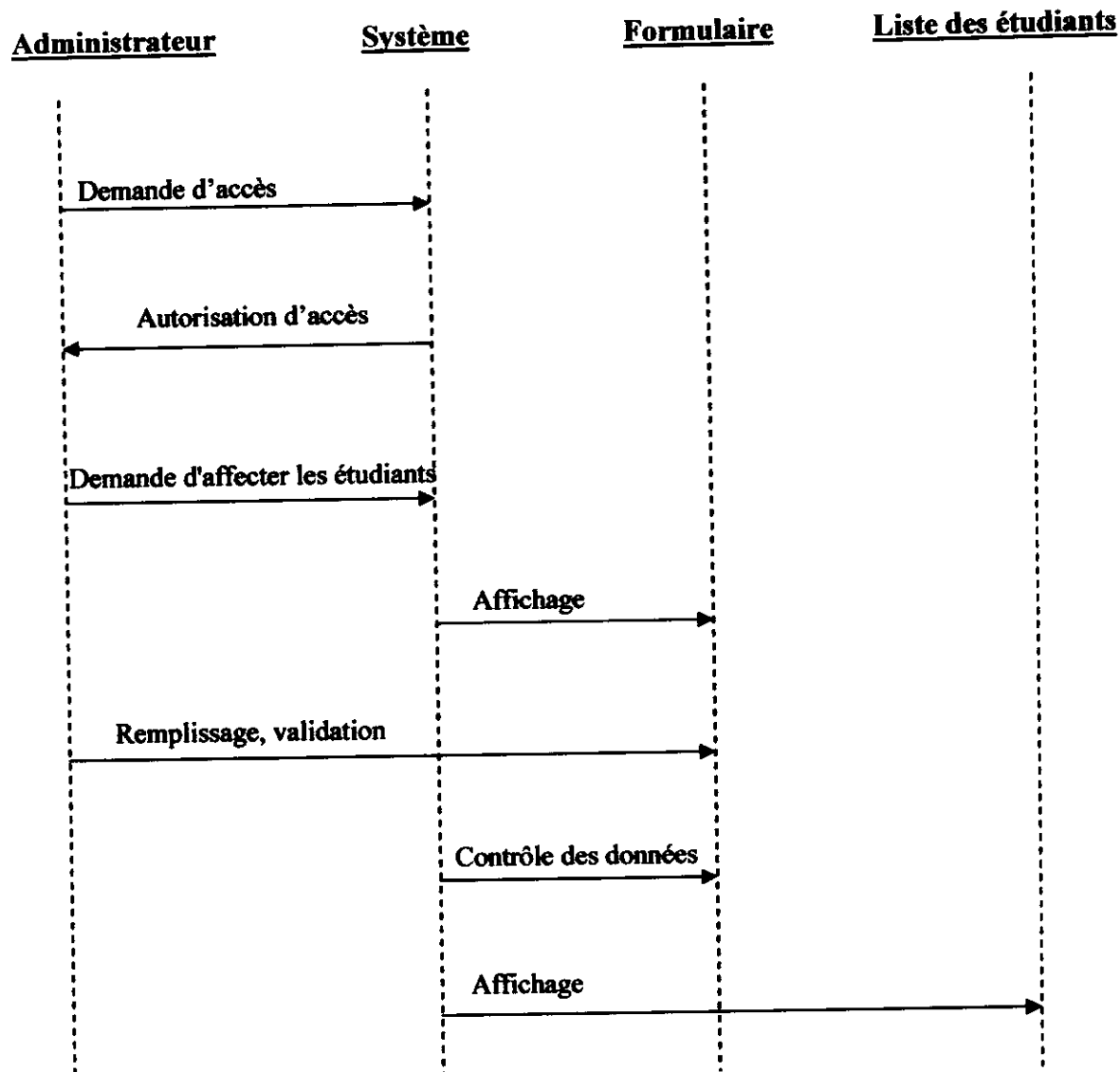
12.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande la fiche d'affectation des étudiants.
5. Le système affiche la page.
6. L'administrateur choisit un numéro de section et un numéro de groupe pour chaque étudiant, puis valide.
7. Le système contrôle les données saisies et affiche la liste des étudiants affectés à leur section et à leur groupe.

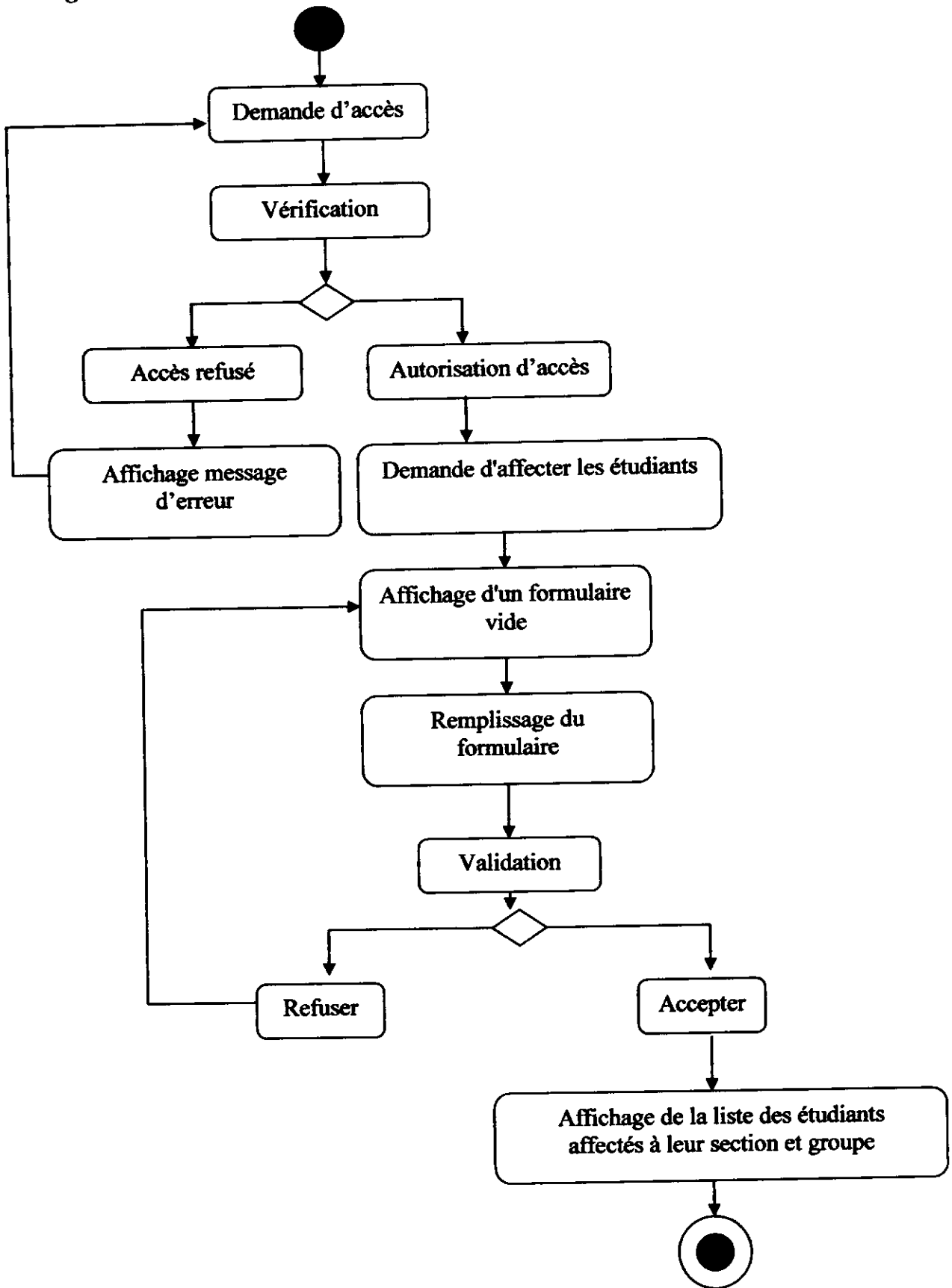
12.2 Diagramme de classe :



12.3 Diagramme de séquence:



12.4 Diagramme d'activité :

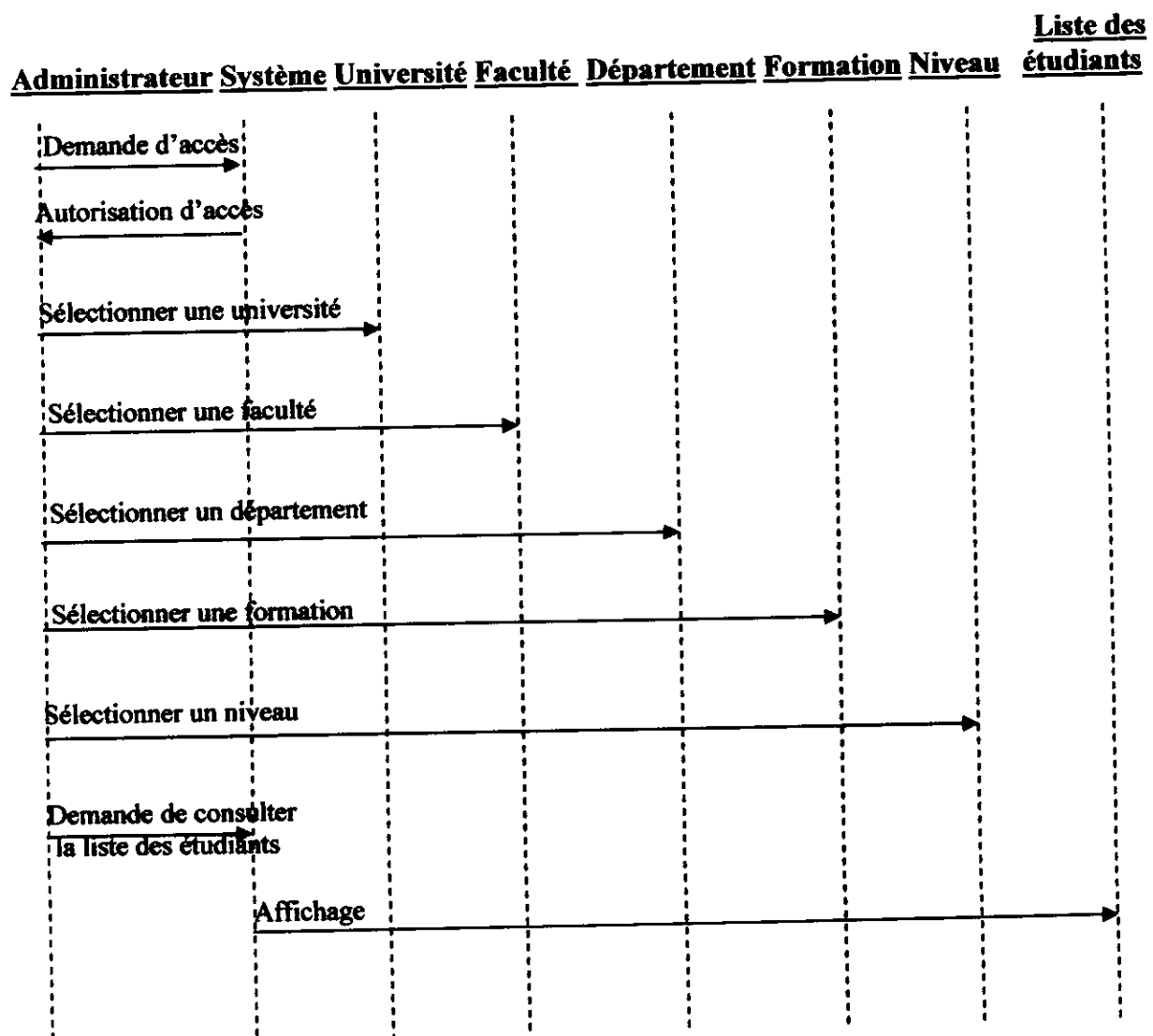


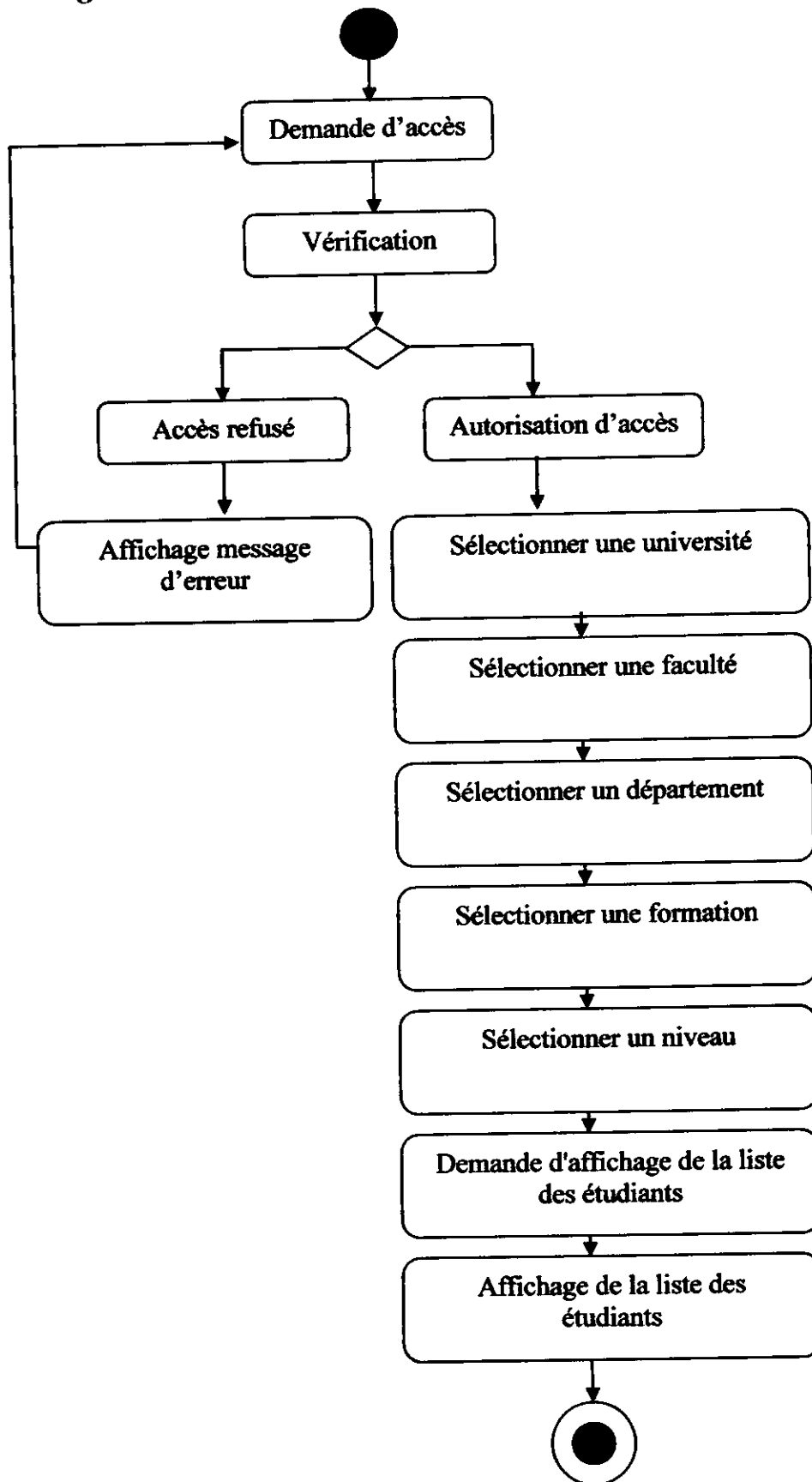
13. Cas d'utilisation «Consulter la liste des étudiants d'un niveau» :

13.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur sélectionne une université.
5. L'administrateur sélectionne une faculté.
6. L'administrateur sélectionne un département.
7. L'administrateur sélectionne une formation.
8. L'administrateur sélectionne un niveau.
9. L'administrateur demande la liste des étudiants de ce niveau.
10. Le système affiche la liste des étudiants de ce niveau

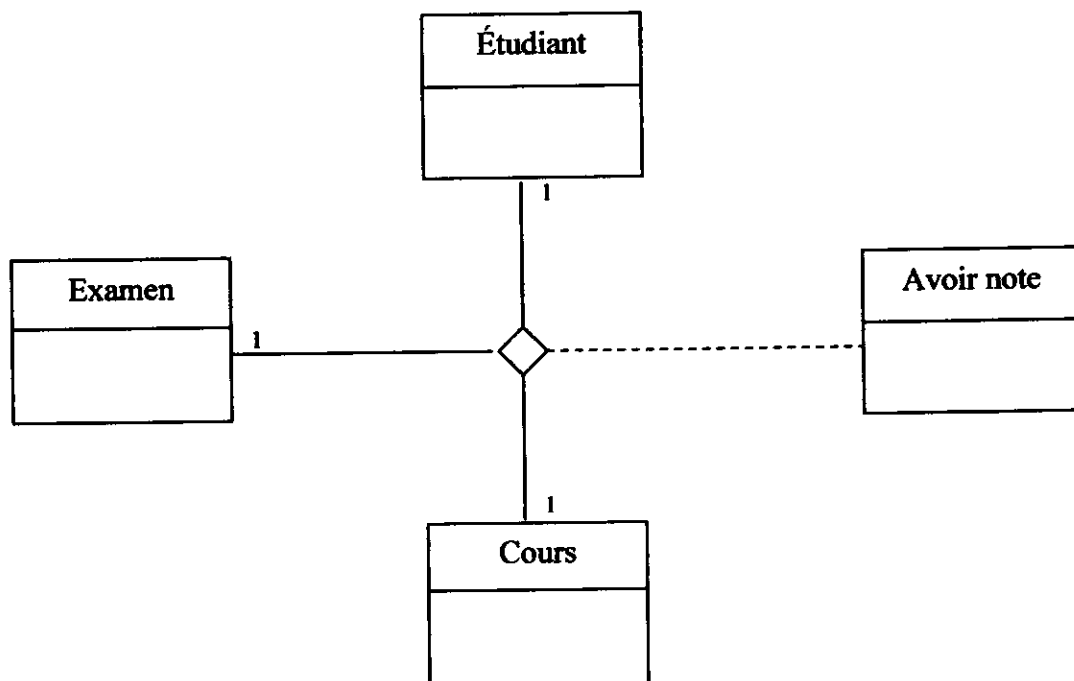
13.2 Diagramme de séquence:



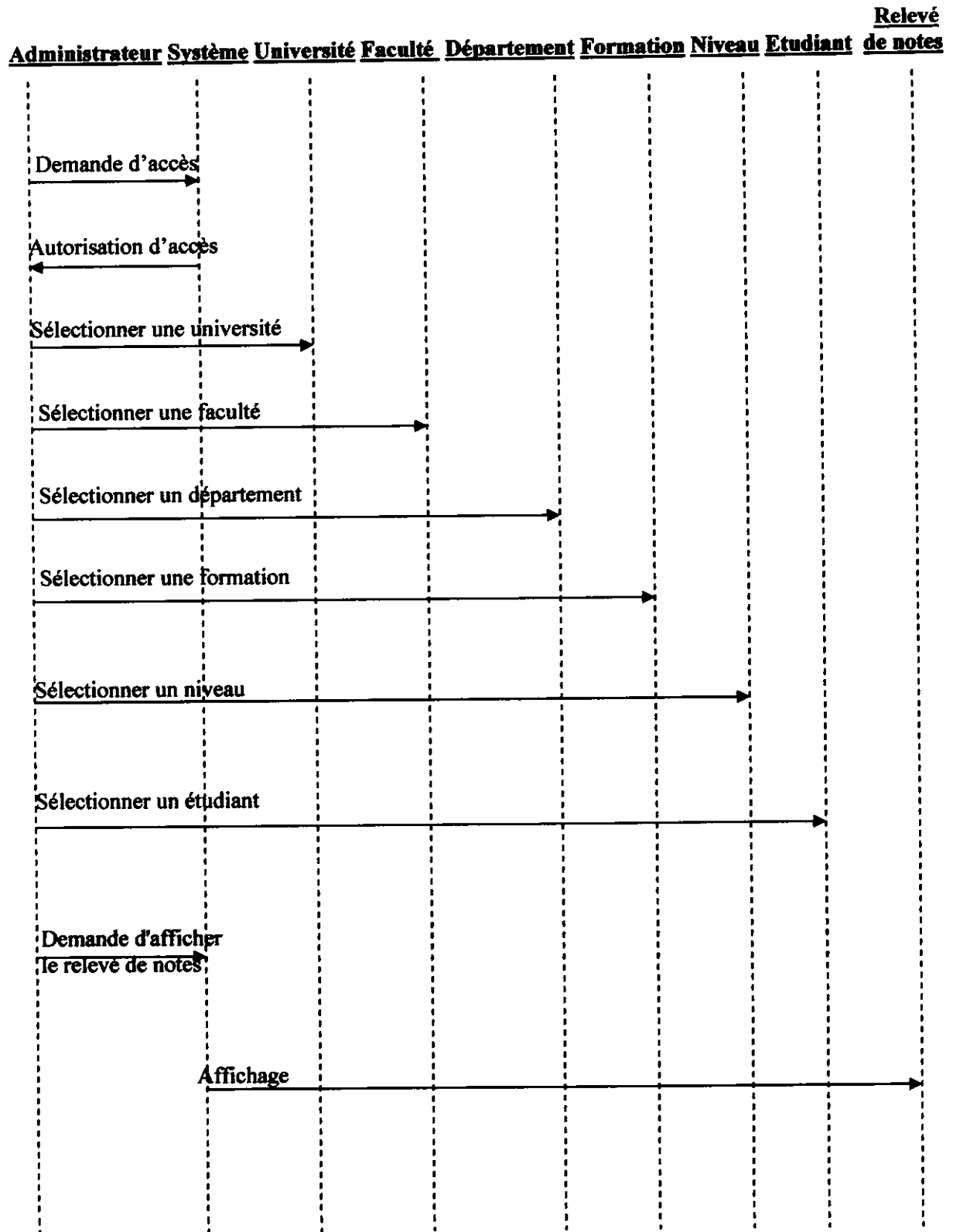
13.3 Diagramme d'activité :

14. Cas d'utilisation «Consulter les relevés de notes» :**14.1 Scénario:**

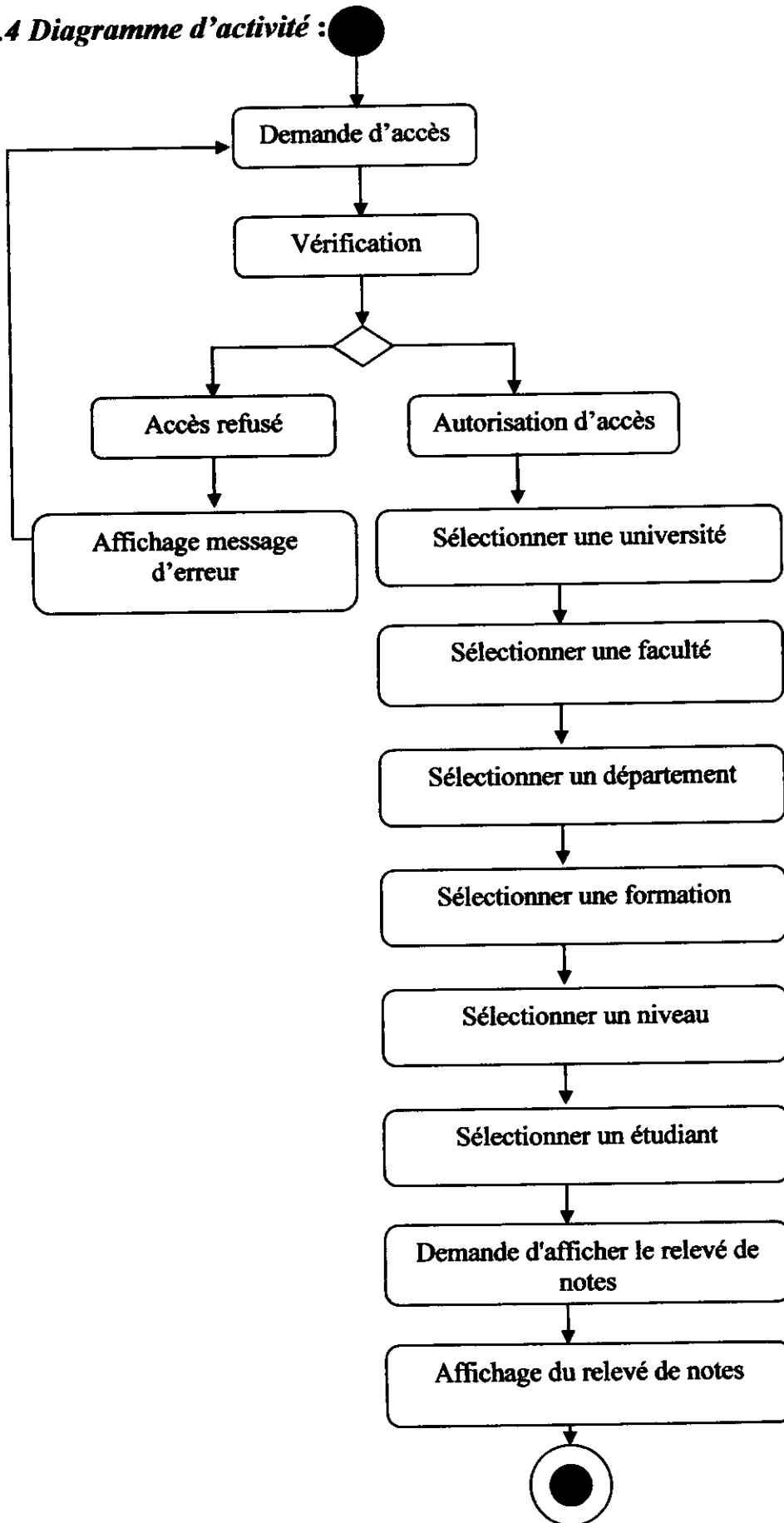
1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur sélectionne une université.
5. L'administrateur sélectionne une faculté.
6. L'administrateur sélectionne un département.
7. L'administrateur sélectionne une formation.
8. L'administrateur sélectionne un niveau.
9. L'administrateur sélectionne un étudiant
10. L'administrateur demande le relevé de notes de l'étudiant.
11. Le système affiche le relevé de notes.

14.2 Diagramme de classe :

14.3 Diagramme de séquence:



14.4 Diagramme d'activité :

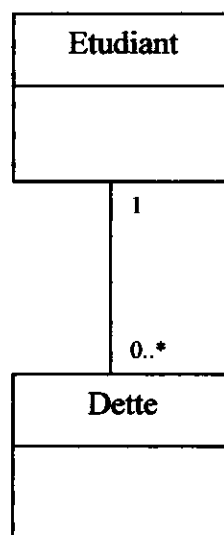


15. Cas d'utilisation «Gestion des dettes» :**15.1 Scénario:**

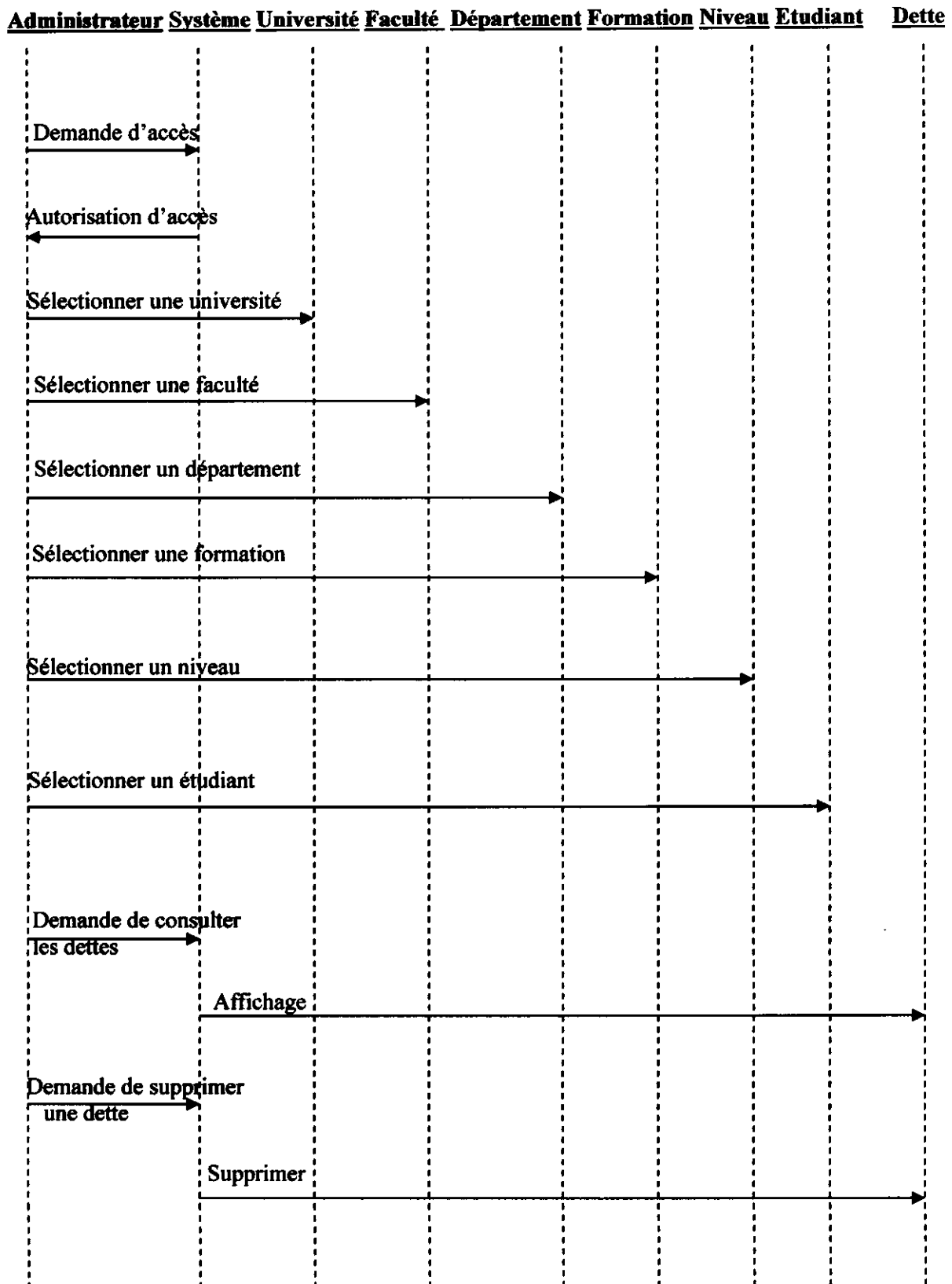
1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur sélectionne une université.
5. L'administrateur sélectionne une faculté.
6. L'administrateur sélectionne un département.
7. L'administrateur sélectionne une formation.
8. L'administrateur sélectionne un niveau.
9. L'administrateur sélectionne un étudiant
10. L'administrateur demande de consulter les dettes.
11. Le système affiche les dettes de l'étudiant (si l'étudiant a des dettes).

Supprimer une dette:

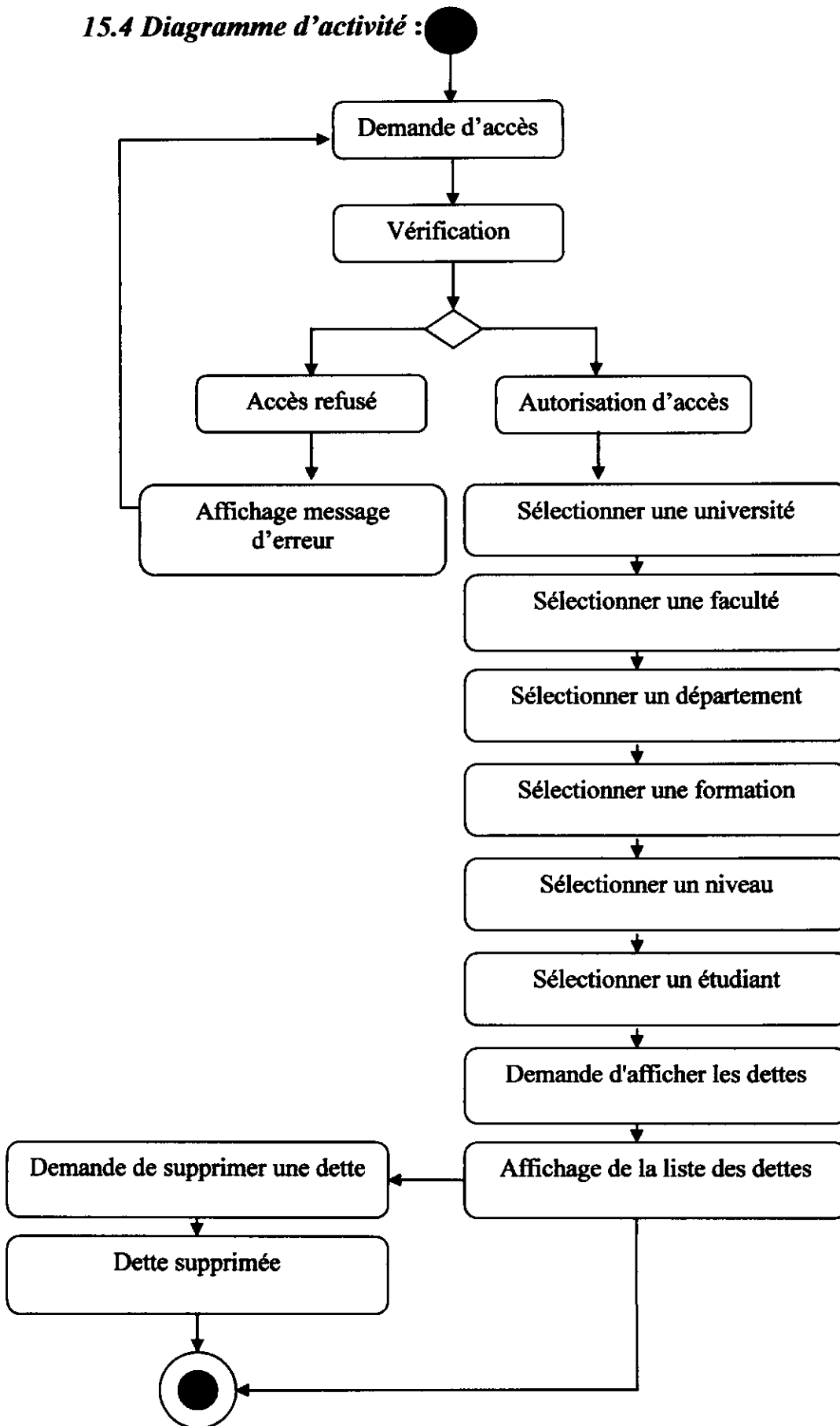
12. L'administrateur demande de supprimer une dette.
13. Le système supprime la dette.

15.2 Diagramme de classe :

15.3 Diagramme de séquence:



15.4 Diagramme d'activité :

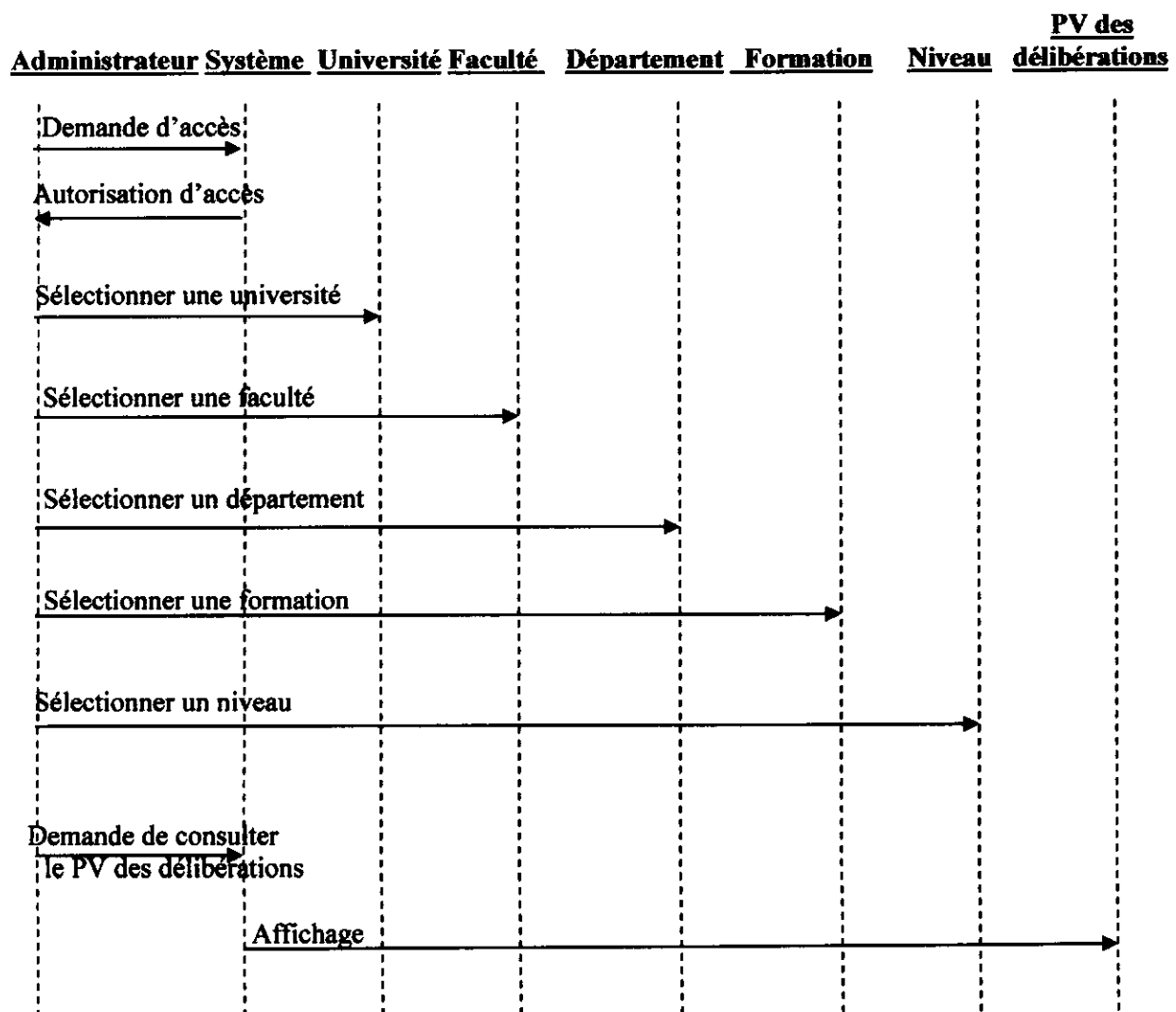


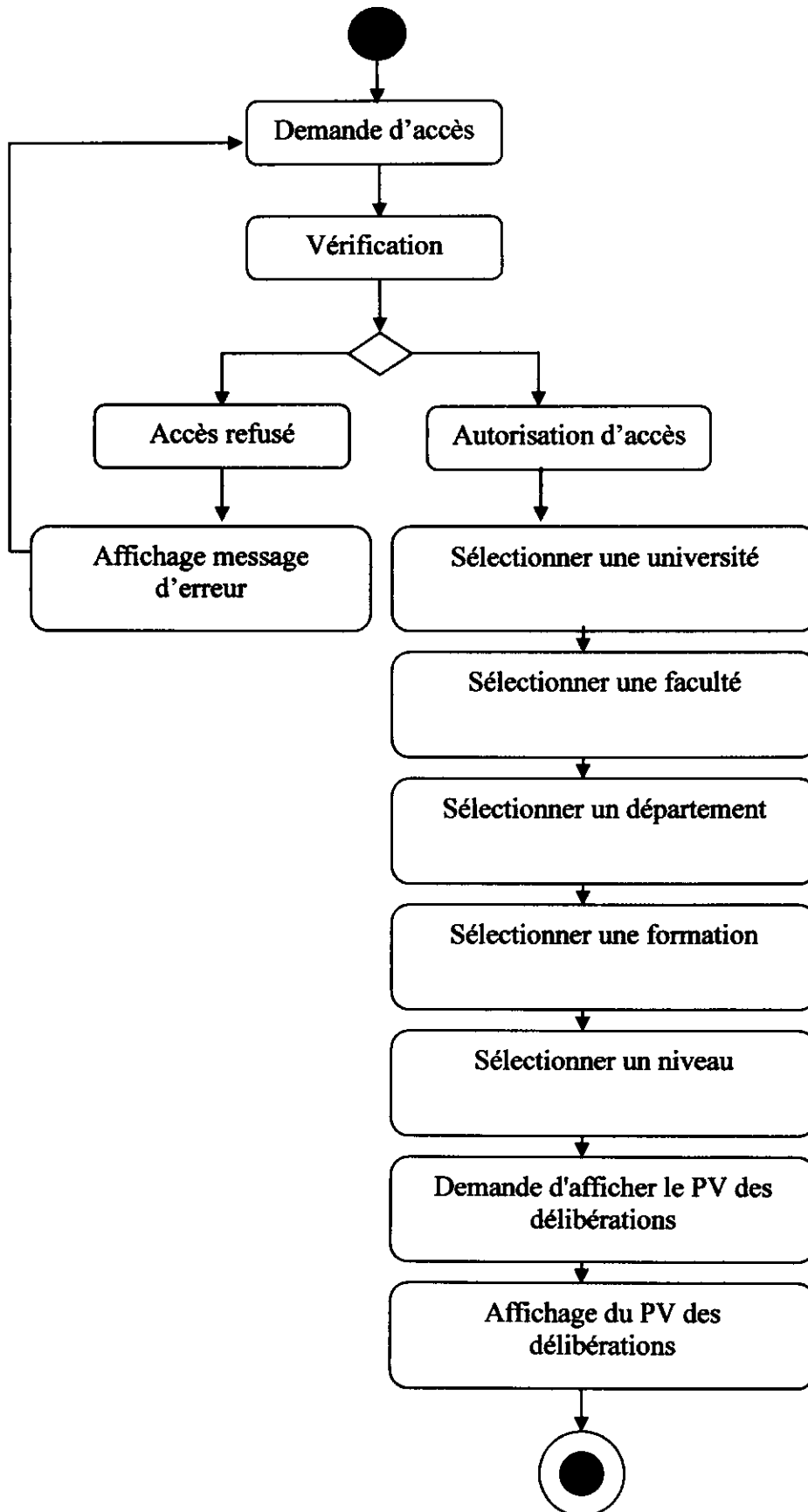
16. Cas d'utilisation « Consulter le PV des délibérations » :

16.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur sélectionne une université.
5. L'administrateur sélectionne une faculté.
6. L'administrateur sélectionne un département.
7. L'administrateur sélectionne une formation.
8. L'administrateur sélectionne un niveau.
9. L'administrateur demande de consulter le PV des délibérations.
10. Le système affiche la page.

16.2 Diagramme de séquence:



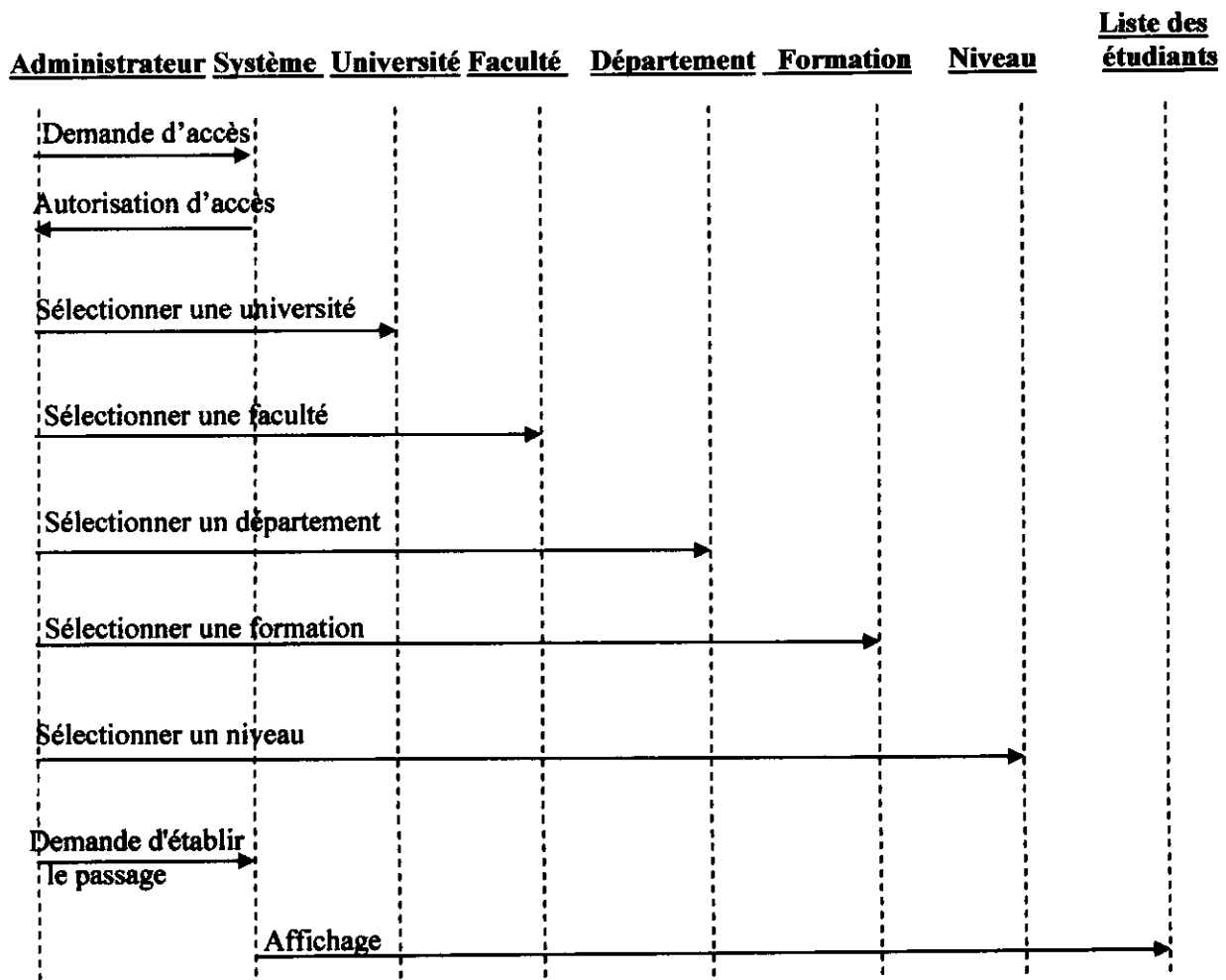
16.3 Diagramme d'activité :

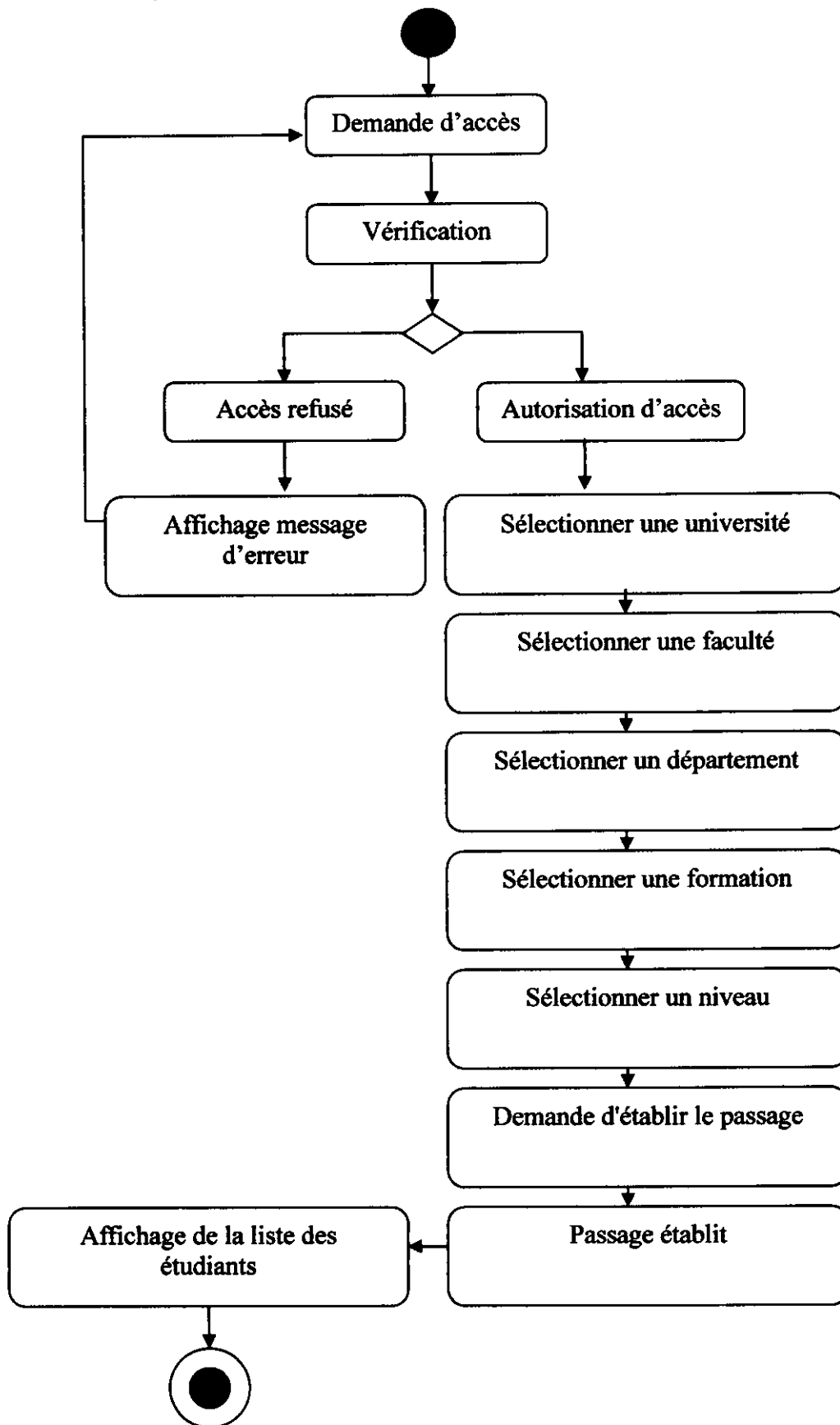
17. Cas d'utilisation «Etablir le passage » :

17.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur sélectionne une université.
5. L'administrateur sélectionne une faculté.
6. L'administrateur sélectionne un département.
7. L'administrateur sélectionne une formation.
8. L'administrateur sélectionne un niveau.
9. L'administrateur demande d'établir le passage des étudiants (si toutes les notes existent).
10. Le système établit le passage et affiche la liste des étudiant avec leur nouveau niveau.

17.2 Diagramme de séquence:



17.3 Diagramme d'activité :

18. Cas d'utilisation «Gestion des outils des cours » :**18.1 Scénario:**

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande d'accéder aux outils des cours.
5. Le système affiche la page.

A. Activer un outil :

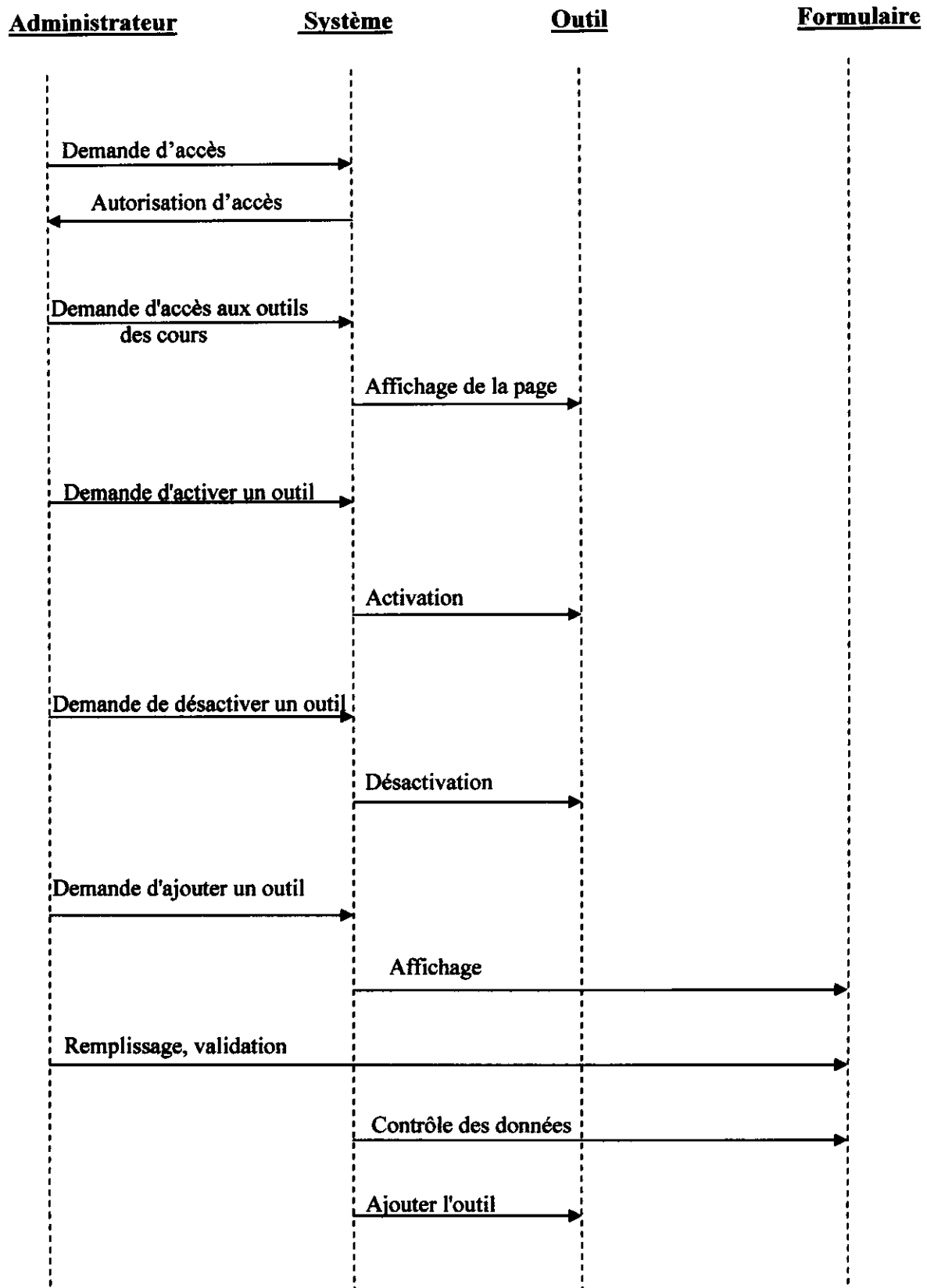
6. L'administrateur demande d'activer un outil.
7. Le système active l'outil.

B. Désactiver un outil :

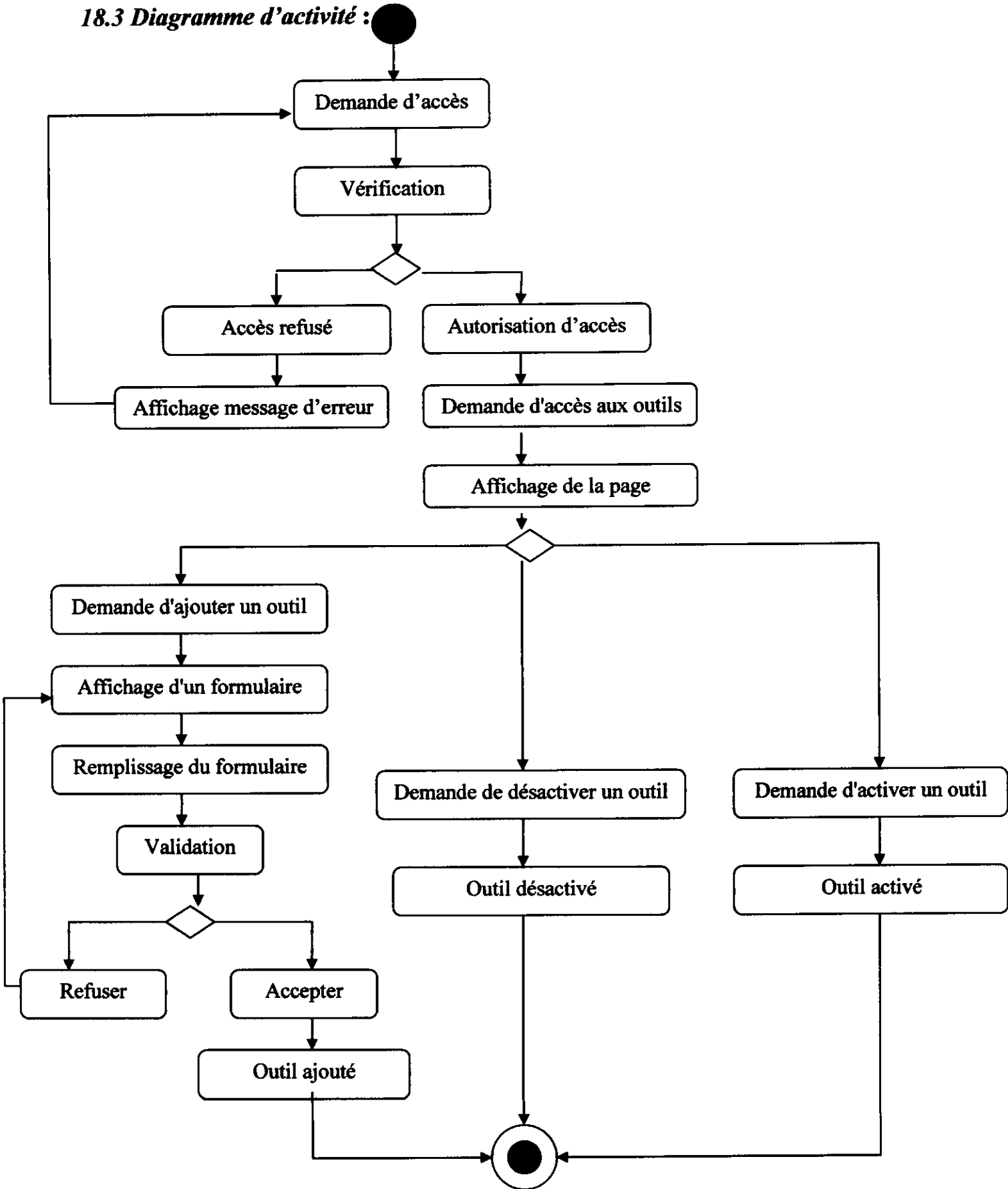
6. L'administrateur demande de désactiver un outil.
7. Le système désactive l'outil.

C. Ajouter un outil :

6. L'administrateur demande d'ajouter un outil.
7. Le système affiche un formulaire.
8. L'administrateur remplit le formulaire, puis valide.
9. Le système ajoute l'outil.

18.2 Diagramme de séquence:

18.3 Diagramme d'activité :



19. Cas d'utilisation « Agenda » :**19.1 Scénario:**

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande d'accéder à l'agenda.
5. Le système affiche la page.

A. Lire un événement :

6. L'administrateur demande de voir les événements de l'agenda.
7. Le système affiche la liste des événements.
8. L'administrateur sélectionne un des événements.
9. Le système affiche l'événement.

B. Ajouter un événement :

6. L'administrateur demande d'ajouter un événement.
7. Le système affiche un formulaire.
8. L'administrateur remplit le formulaire, puis valide.
9. Le système ajoute l'événement.

C. Supprimer un événement :

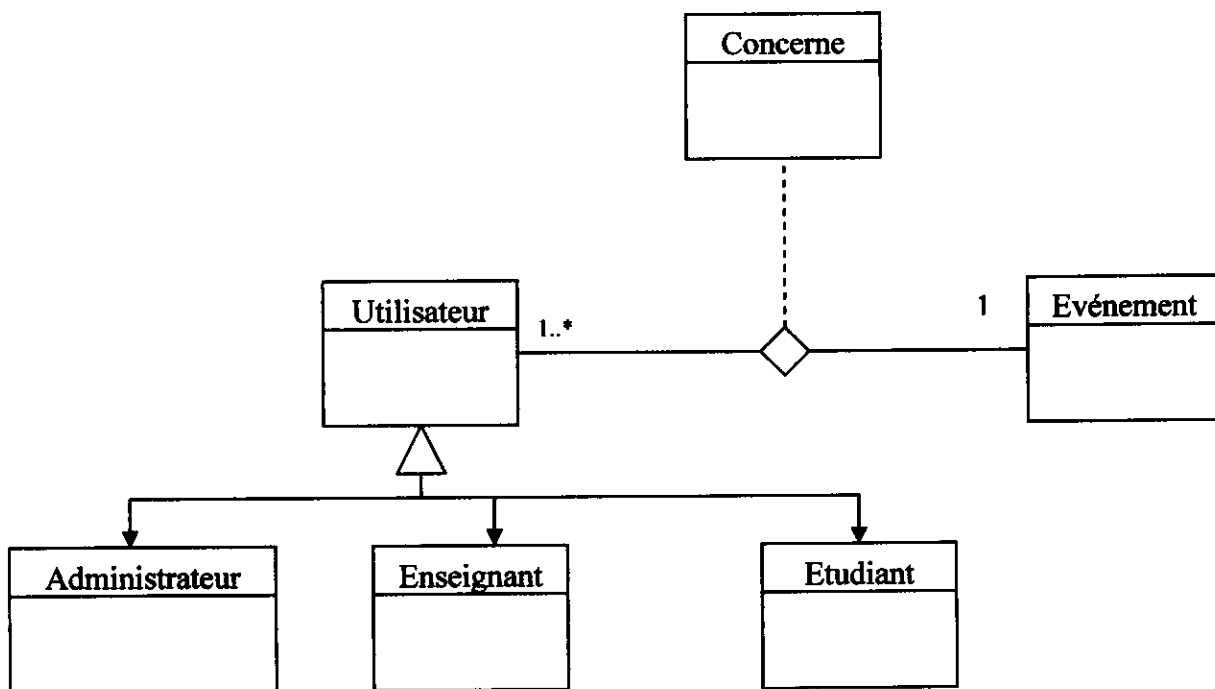
6. L'administrateur demande de supprimer un événement.
7. Le système supprime l'événement.

D. Modifier un événement :

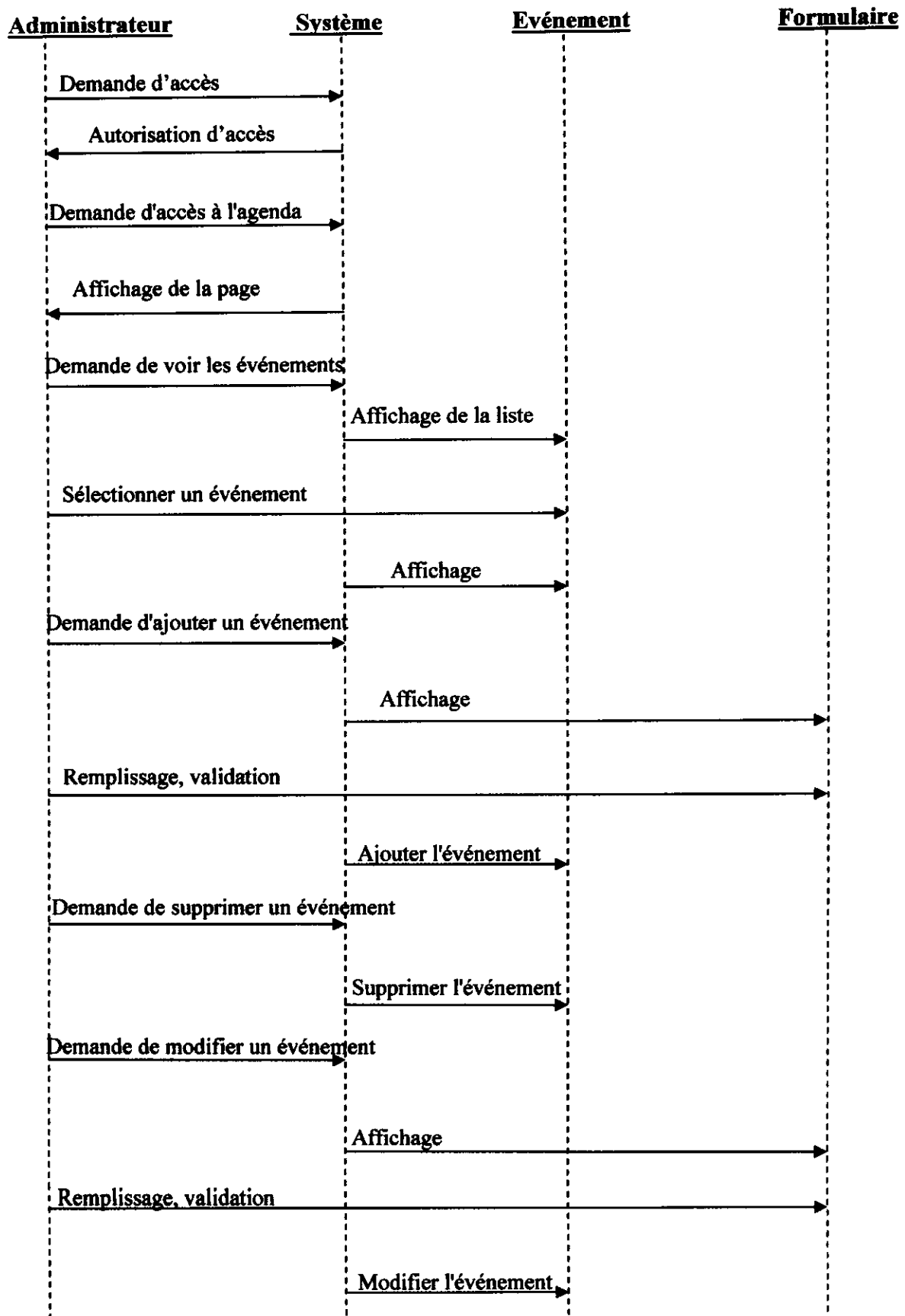
6. L'administrateur demande de modifier un événement.
7. Le système affiche un formulaire.
8. L'administrateur remplit le formulaire, puis valide.
9. Le système modifie l'événement.

Remarque :

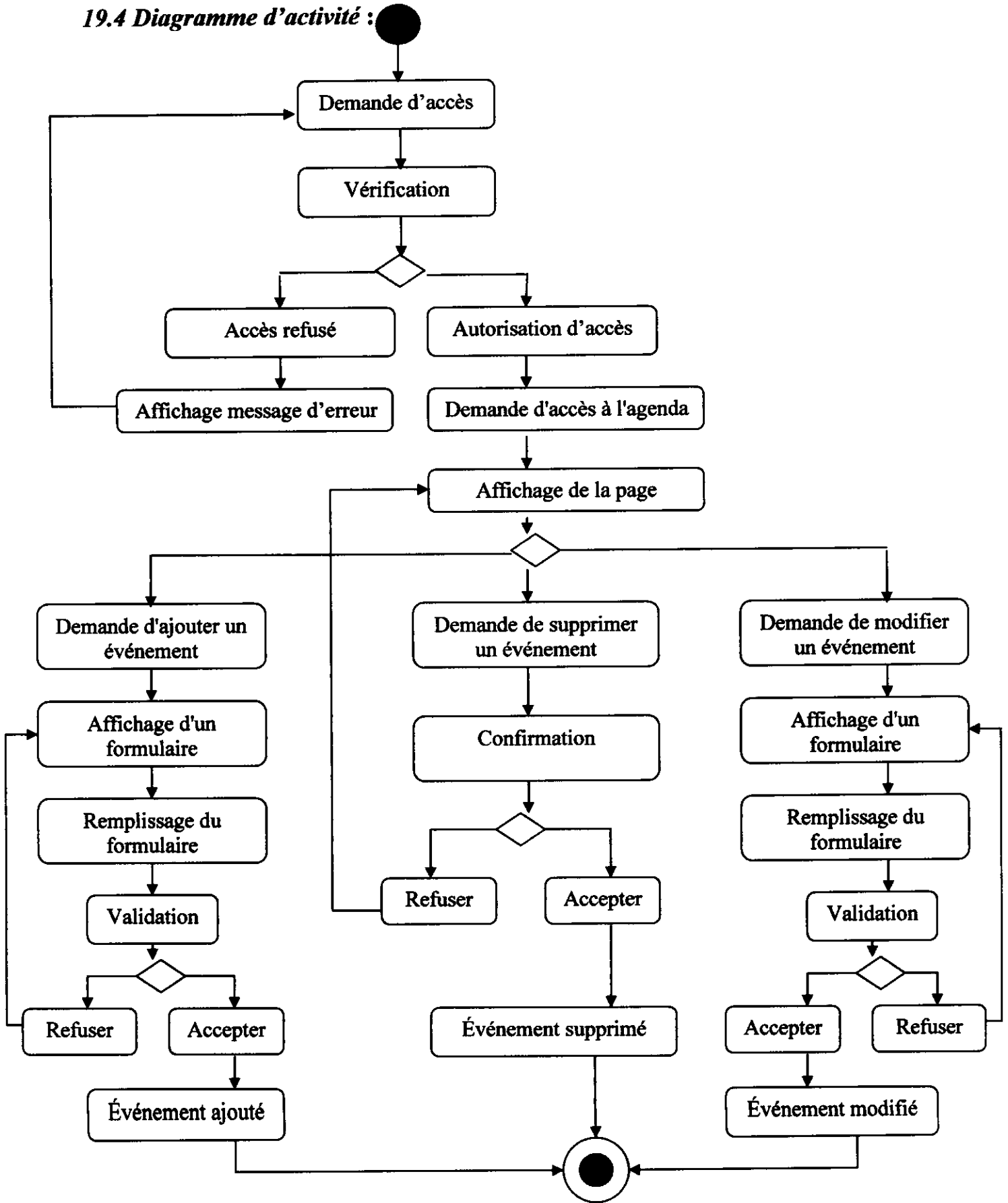
L'administrateur peut aussi exploiter l'agenda comme un calendrier.

19.2 Diagramme de classe :

19.3 Diagramme de séquence:



19.4 Diagramme d'activité :



20. Cas d'utilisation «Forums » :**20.1 Scénario:**

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande d'accéder aux forums.
5. Le système affiche la page.

A. Créer un forum :

6. L'administrateur demande de créer un forum.
7. Le système affiche un formulaire.
8. L'administrateur remplit le formulaire, puis valide.
9. Le système crée le forum.

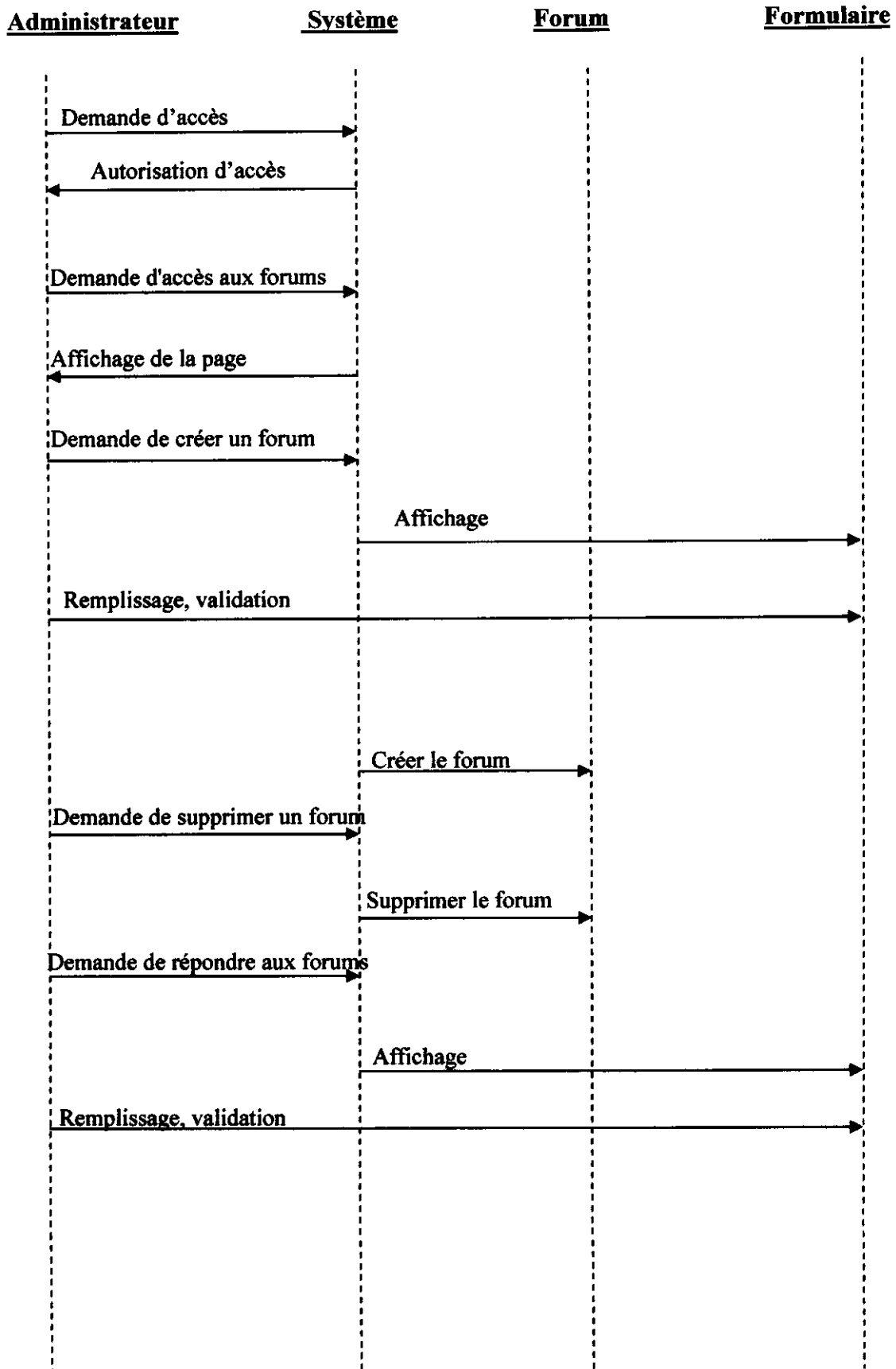
B. Supprimer un forum :

6. L'administrateur demande de supprimer un forum.
7. Le système supprime le forum.

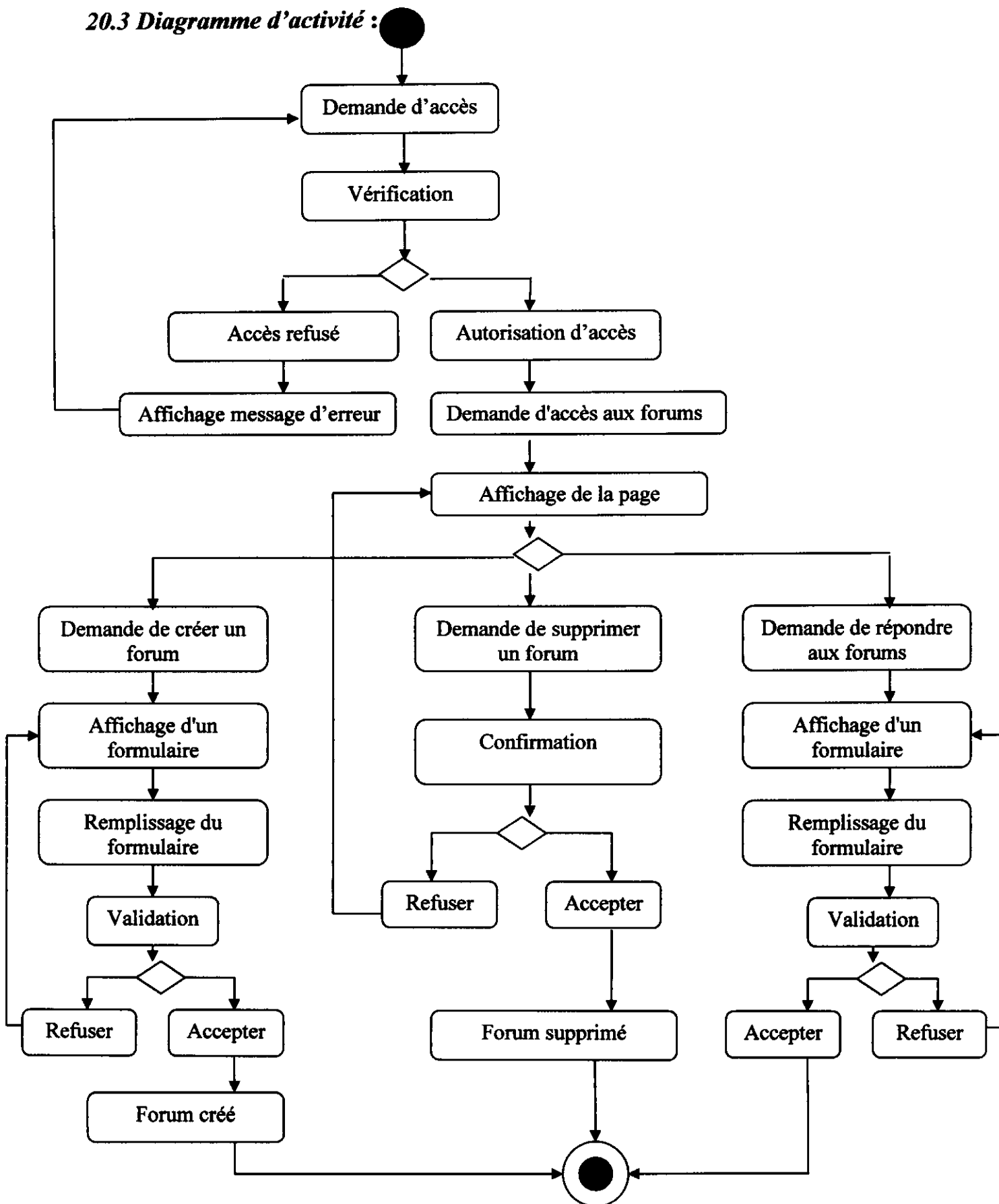
C. Répondre aux forums :

6. L'administrateur demande de répondre à un forum.
7. Le système affiche un formulaire.
8. L'administrateur remplit le formulaire, puis valide.

20.2 Diagramme de séquence:



20.3 Diagramme d'activité :



21. Cas d'utilisation «Annonces » :**21.1 Scénario:**

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande d'accéder aux annonces.
5. Le système affiche la page.

A. Ajouter une annonce :

6. L'administrateur demande d'ajouter une annonce.
7. Le système affiche un formulaire.
8. L'administrateur remplit le formulaire, puis valide.
9. Le système ajoute l'annonce.

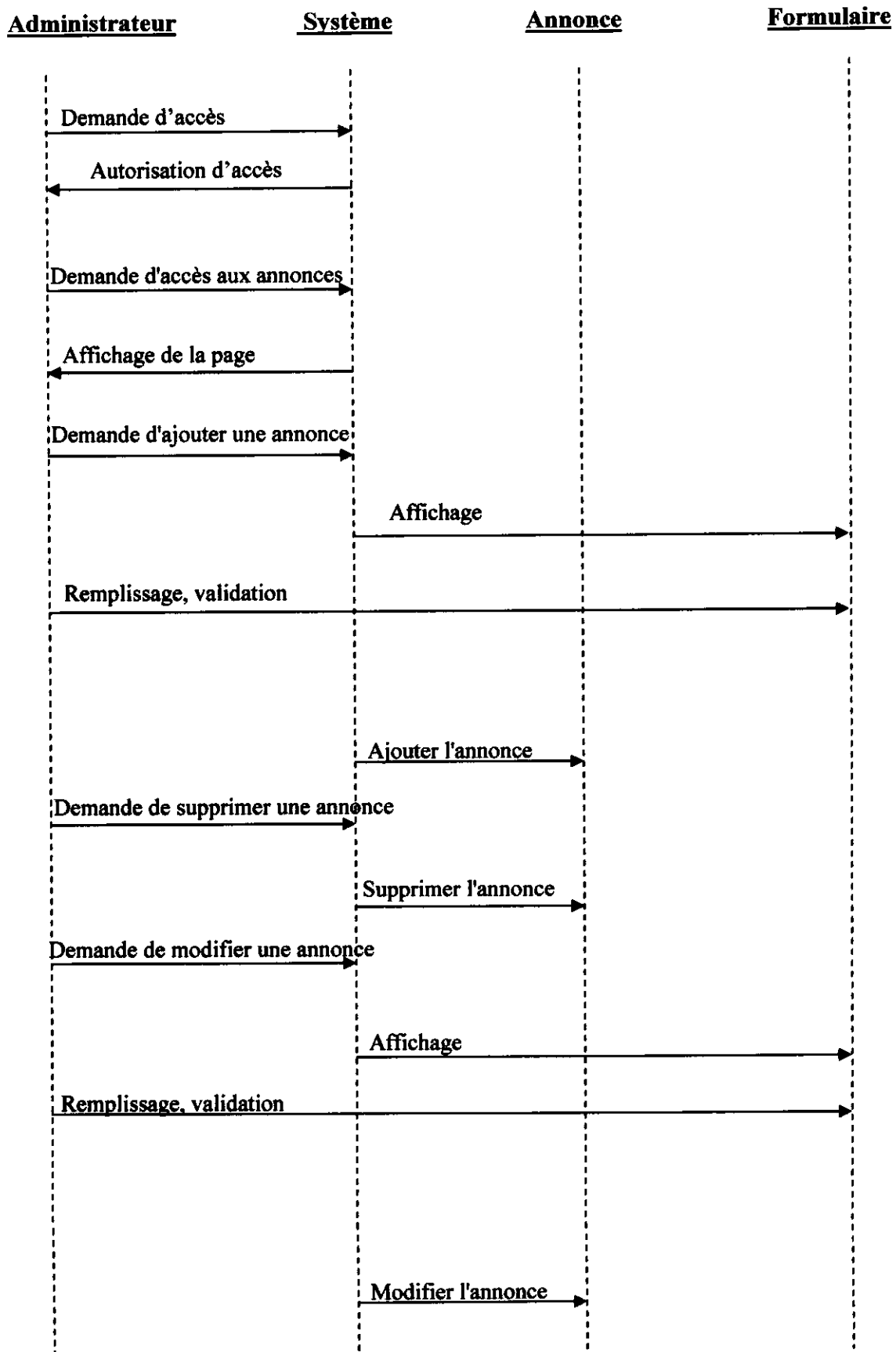
B. Supprimer une annonce:

6. L'administrateur demande de supprimer une annonce.
7. Le système supprime l'annonce.

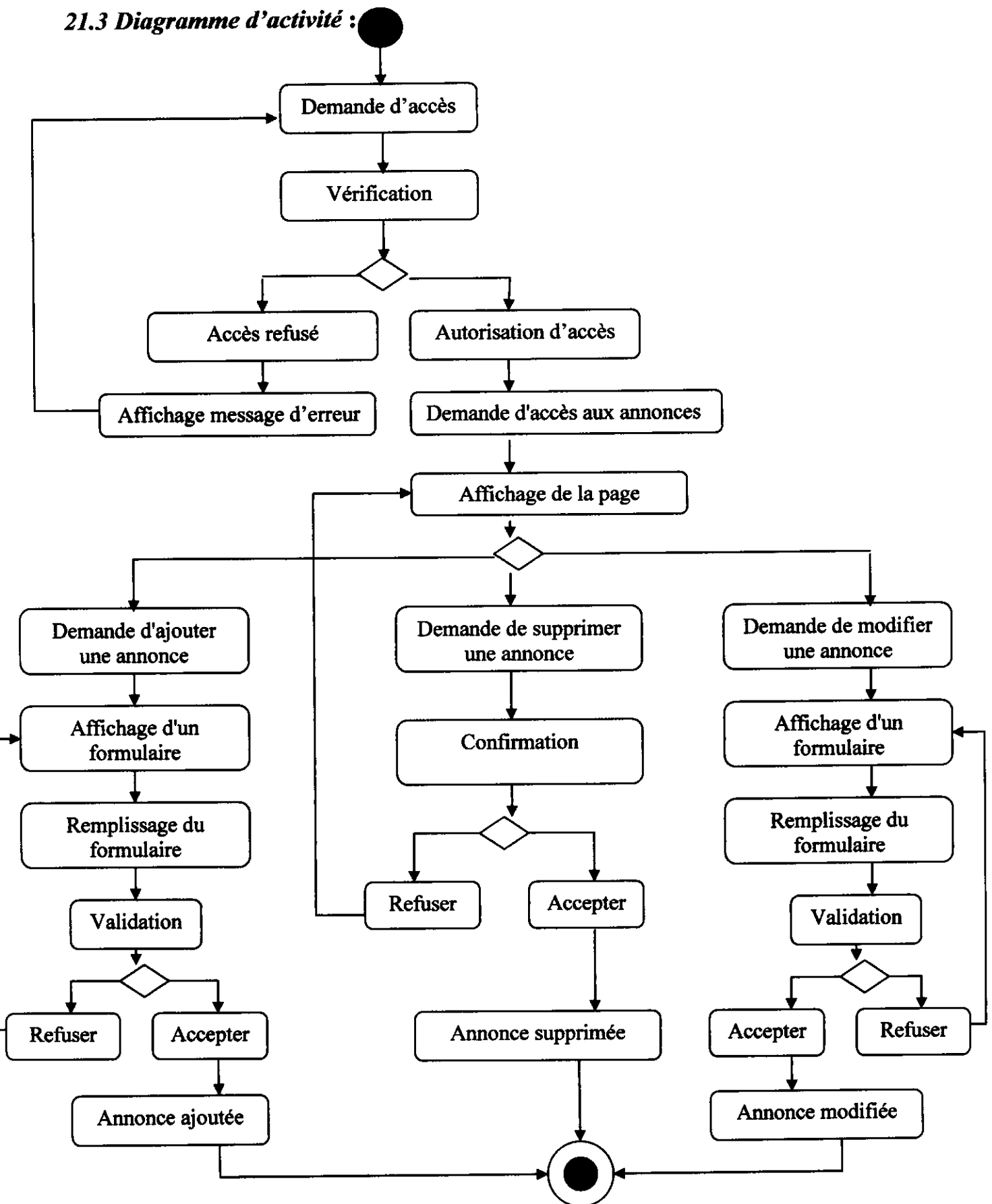
C. Modifier une annonce:

6. L'administrateur demande de modifier une annonce.
7. Le système affiche un formulaire.
8. L'administrateur remplit le formulaire, puis valide.
9. Le système modifie l'annonce.

21.2 Diagramme de séquence:



21.3 Diagramme d'activité :



22. Cas d'utilisation «Messagerie » :

22.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande d'accéder aux messages.
5. Le système affiche la page.

A. Ecriture d'un message :

6. L'administrateur demande d'écrire un message.
7. Le système affiche un formulaire.
8. L'administrateur écrit le message, puis valide.

B. Lecture d'un message:

6. L'administrateur demande de voir ses messages.
7. Le système affiche les messages.

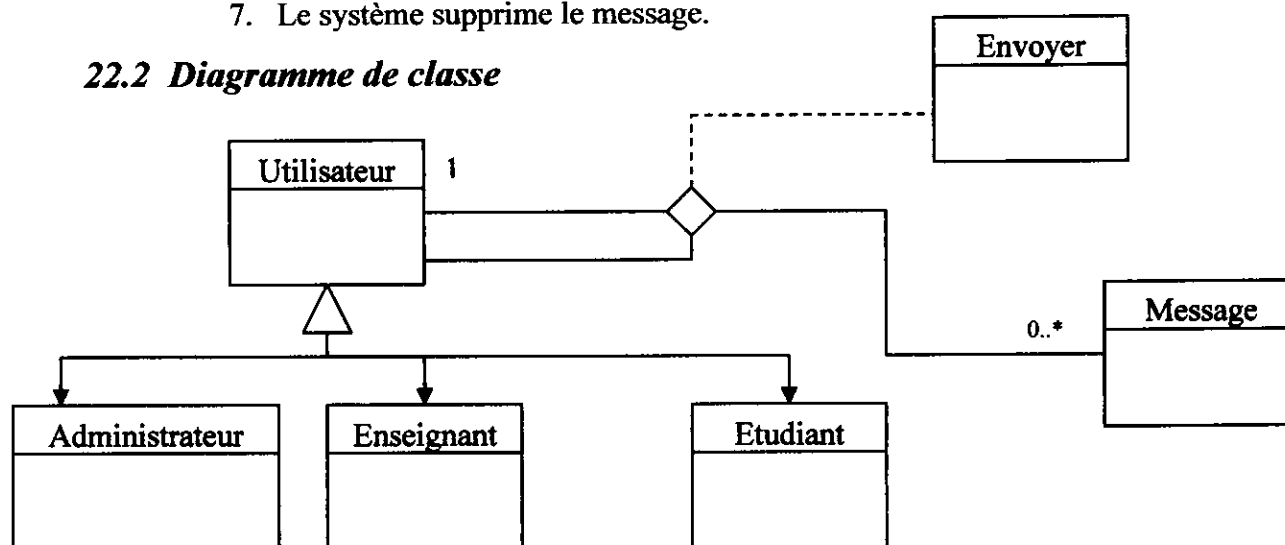
C. Répondre aux messages:

6. L'administrateur demande de répondre à un message.
7. Le système affiche un formulaire.
8. L'administrateur écrit sa réponse, puis valide.

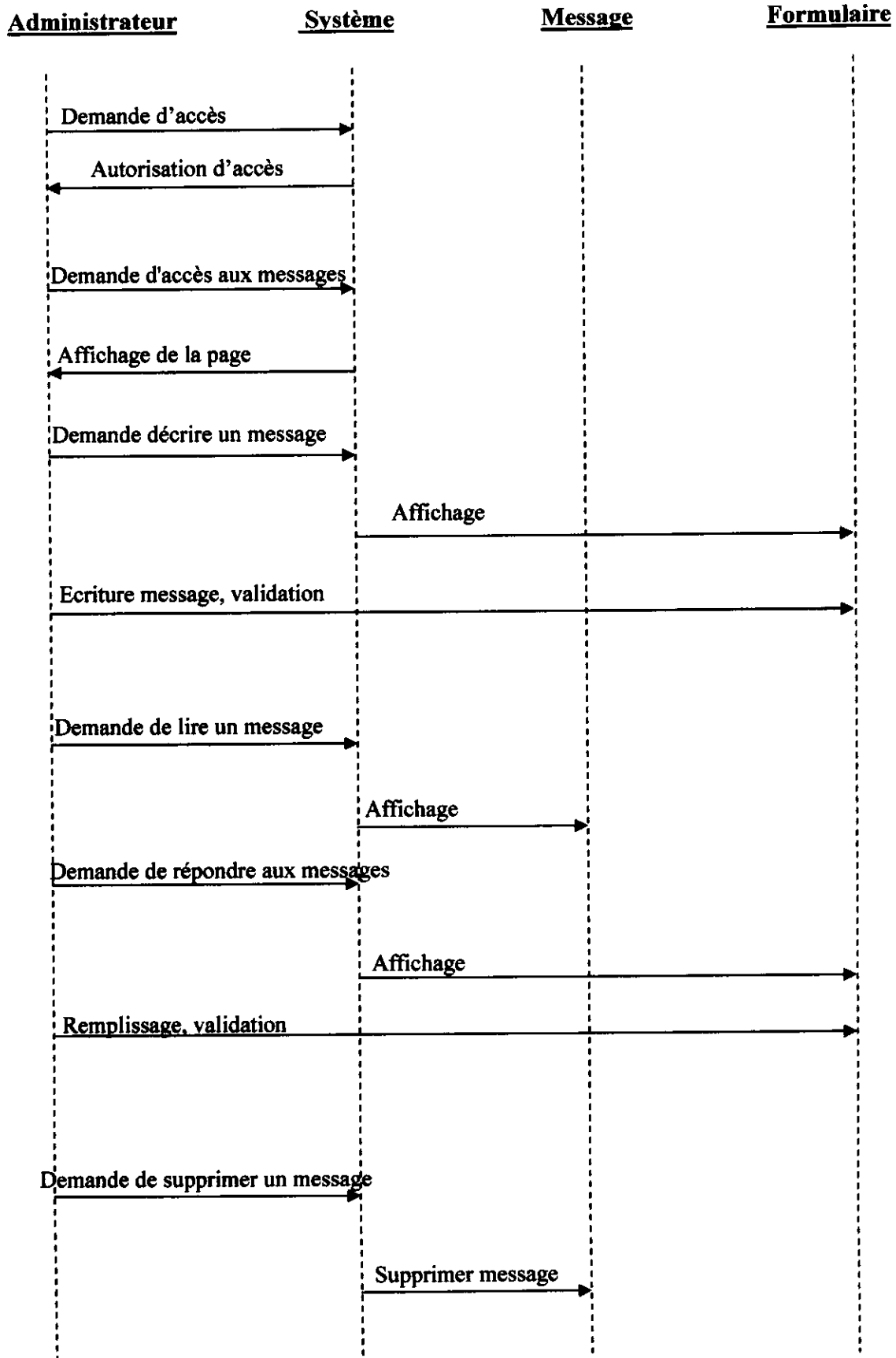
D. Supprimer un message:

6. L'administrateur demande de supprimer un message.
7. Le système supprime le message.

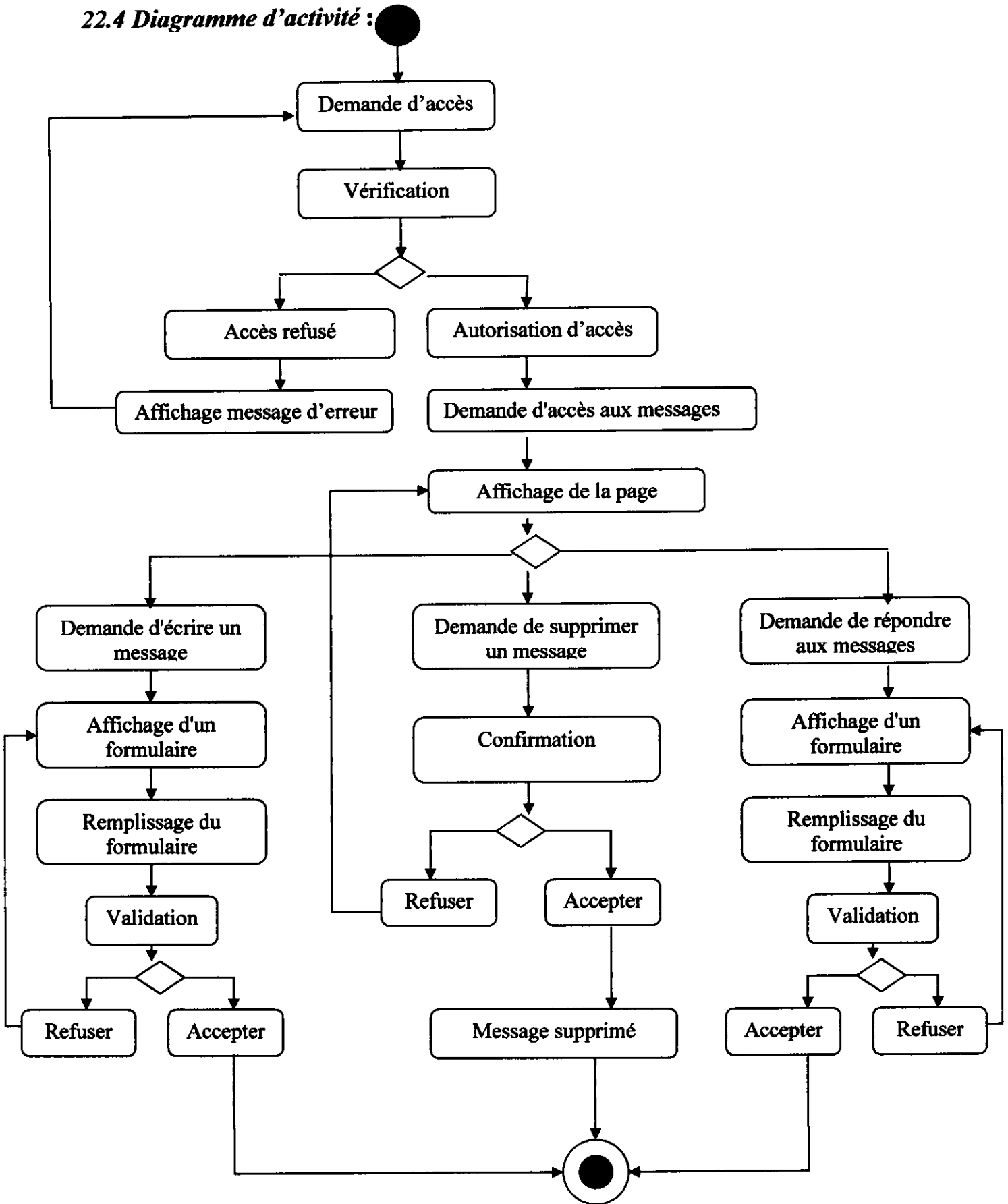
22.2 Diagramme de classe



22.3 Diagramme de séquence:



22.4 Diagramme d'activité :

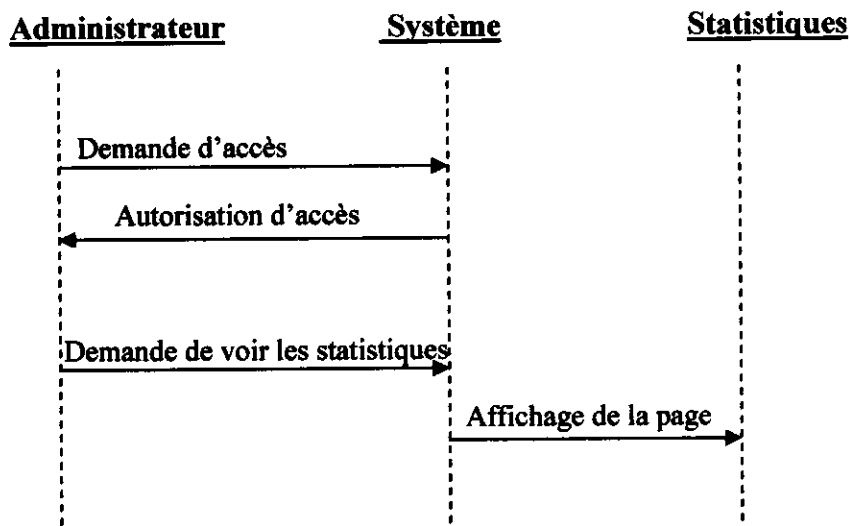


23. Cas d'utilisation « Analyser les statistiques » :

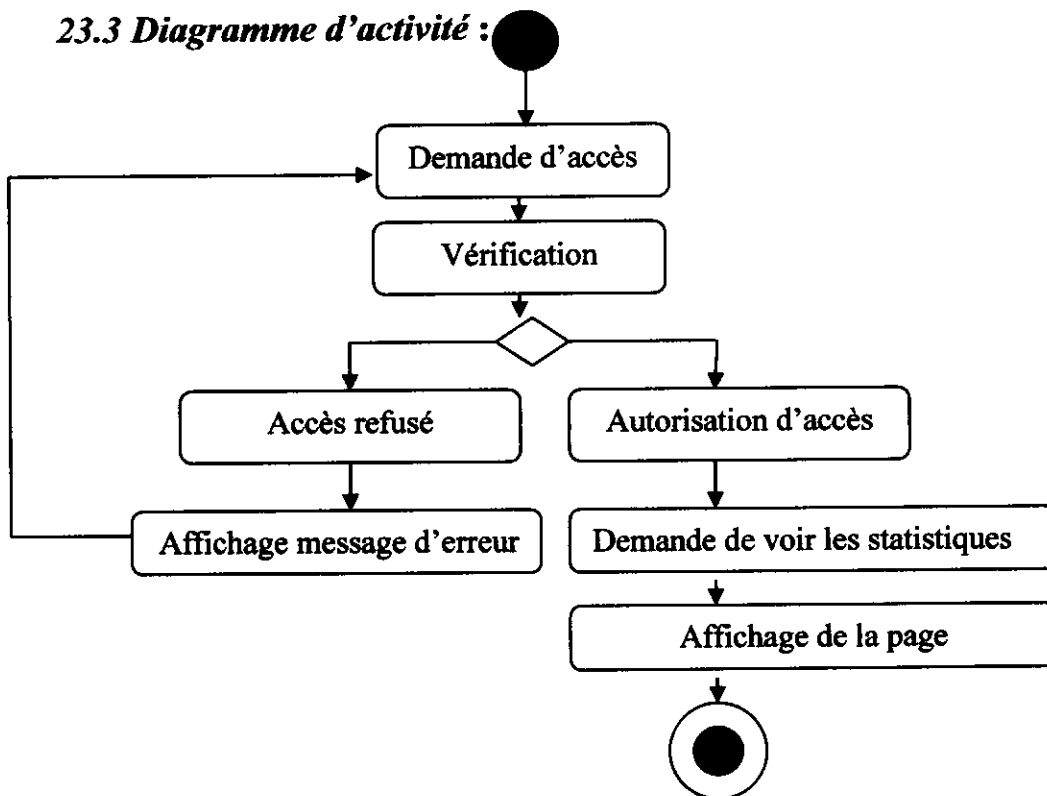
23.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande d'analyser les statistiques.
5. Le système affiche la page.

23.2 Diagramme de séquence:



23.3 Diagramme d'activité :

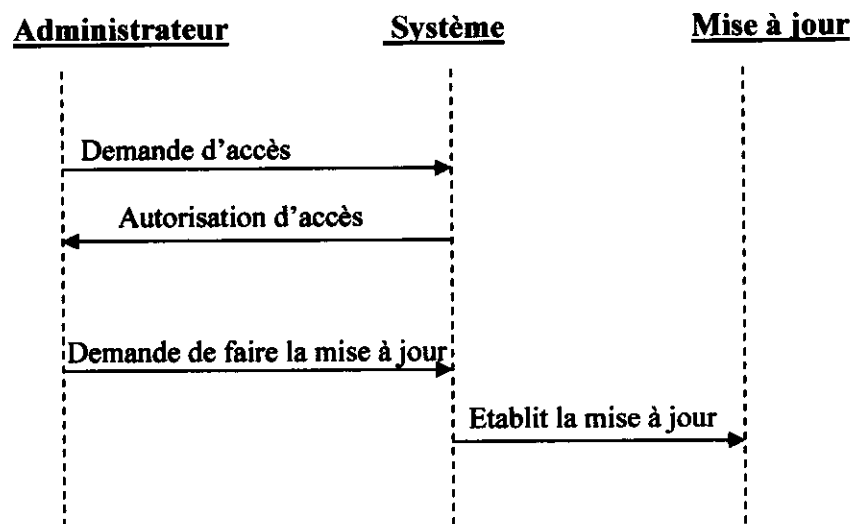


24. Cas d'utilisation «Mise à jour » :

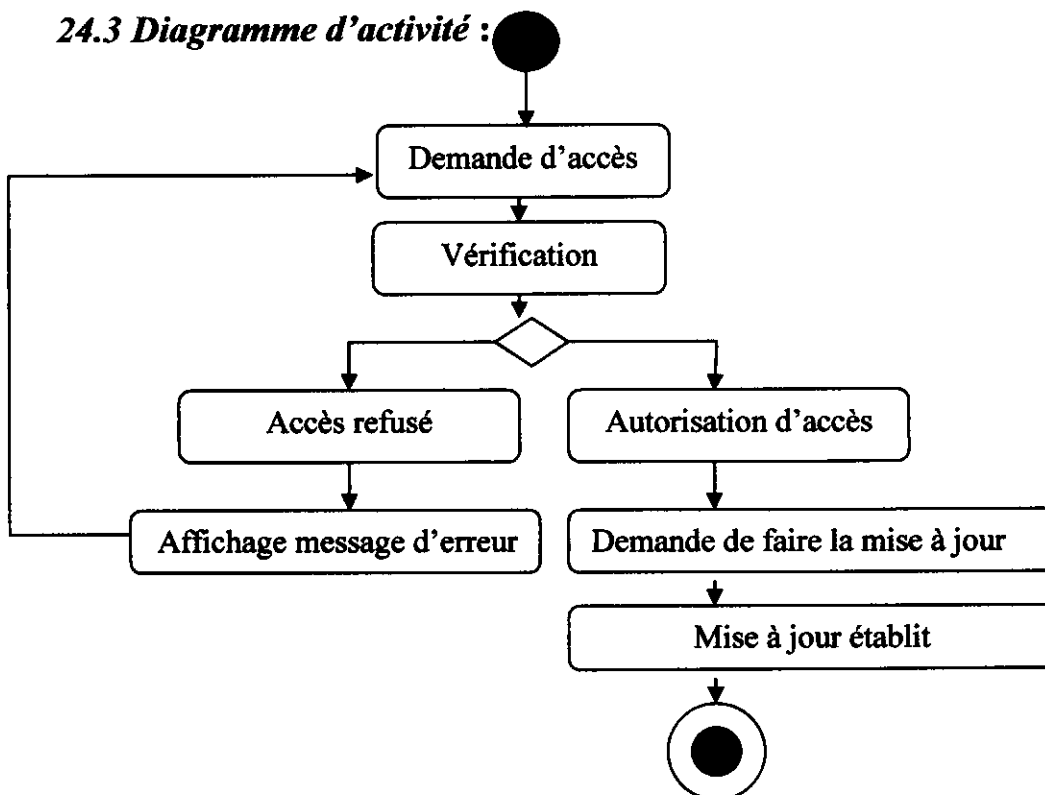
24.1 Scénario:

1. L'administrateur envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'administrateur.
4. L'administrateur demande la mise à jour.
5. Le système établit la mise à jour.

24.2 Diagramme de séquence:



24.3 Diagramme d'activité :



25. Cas d'utilisation «Ouverture et fermeture de l'année scolaire » :

La fermeture d'une année scolaire se fait par l'établissement du passage des étudiant, cette étape permet d'affecter chaque étudiant à son niveau dans l'année scolaire prochaine. L'ouverture d'une année scolaire se fait par la détermination de l'année d'étude ensuite l'administrateur a le choix:

- Il peut insérer de nouvelles universités, nouvelles facultés, de nouveaux départements, de nouvelles formations, de nouveaux enseignants, de nouveaux étudiants, de nouveaux cours..., dans le cas par exemple ou c'est la première utilisation de ce logiciel.
- Il peut récupérer les données de l'année précédente que ce soit les université, les facultés, les départements, les formations, les unités d'enseignement, les cours, les étudiants, les enseignants..., cette récupération des données est possible parce que notre logiciel garde les bases de données de toutes les années scolaires, bien sur l'administrateur peut toujours supprimer la base de données de n'importe quel année.

Remarque:

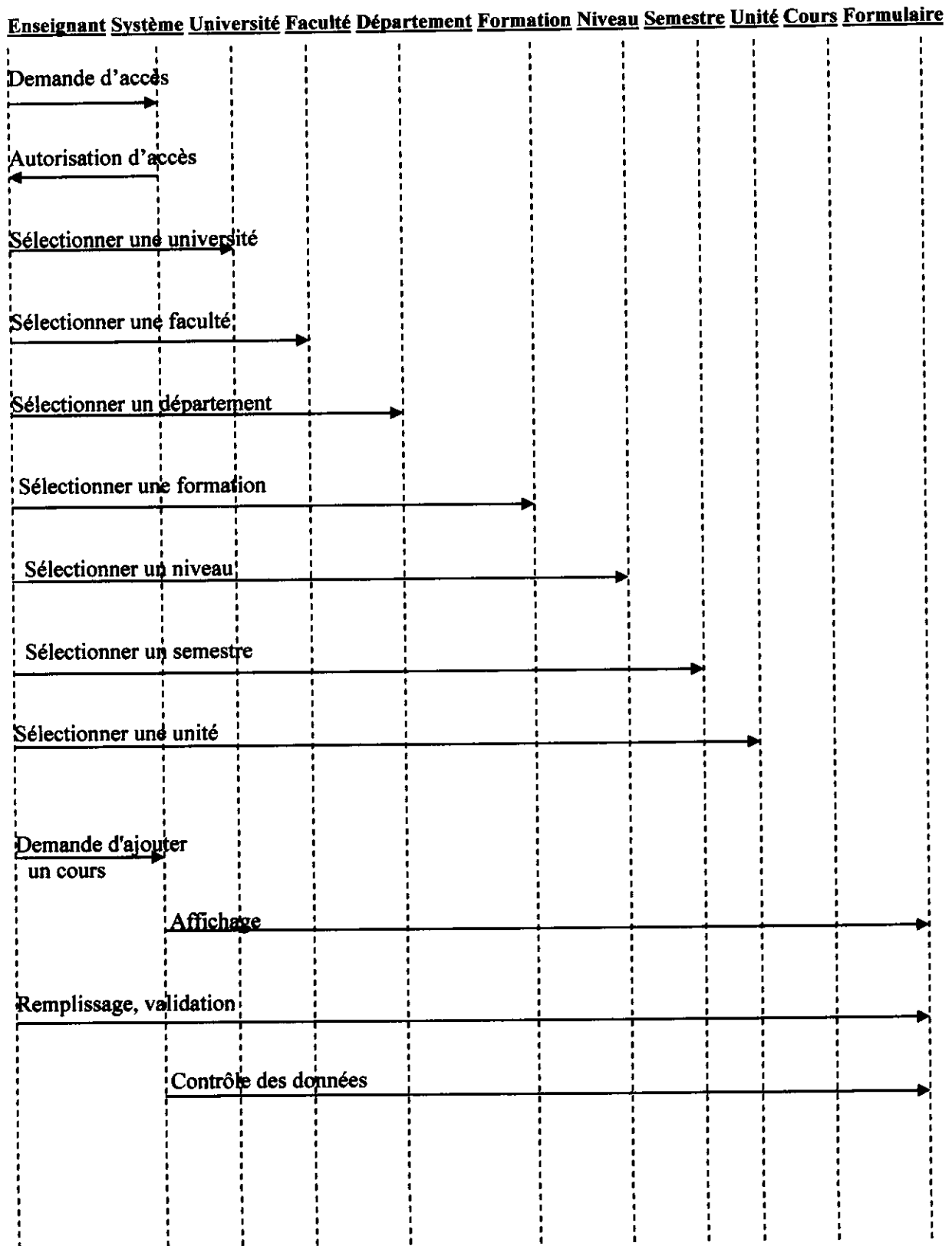
On peut restaurer le site de n'importe quel année scolaire tant que sa base de données existe. La sauvegarde des bases de données permet de résoudre beaucoup de problèmes dont :

- ✓ Le problème des dettes : dans le système LMD l'étudiant peut passer à un niveau supérieur en ayant des dettes, notre logiciel garde en mémoire toutes les dettes depuis le premier niveau et comme ça on pourra savoir lors de l'obtention du diplôme si l'étudiant a des dettes ou non .
- ✓ Les étudiants ajournés et qui ont des notes de cours supérieures ou égales à dix et dont il faudra garder en mémoire.
- ✓ Le plus important dans tous ça c'est qu'on peut avoir tout le parcours de l'étudiant depuis le premier niveau et la première formation (grâce à la restauration des sites des années précédentes) et qui peut être très utiles pour l'étudiant (exemple pour le CV).

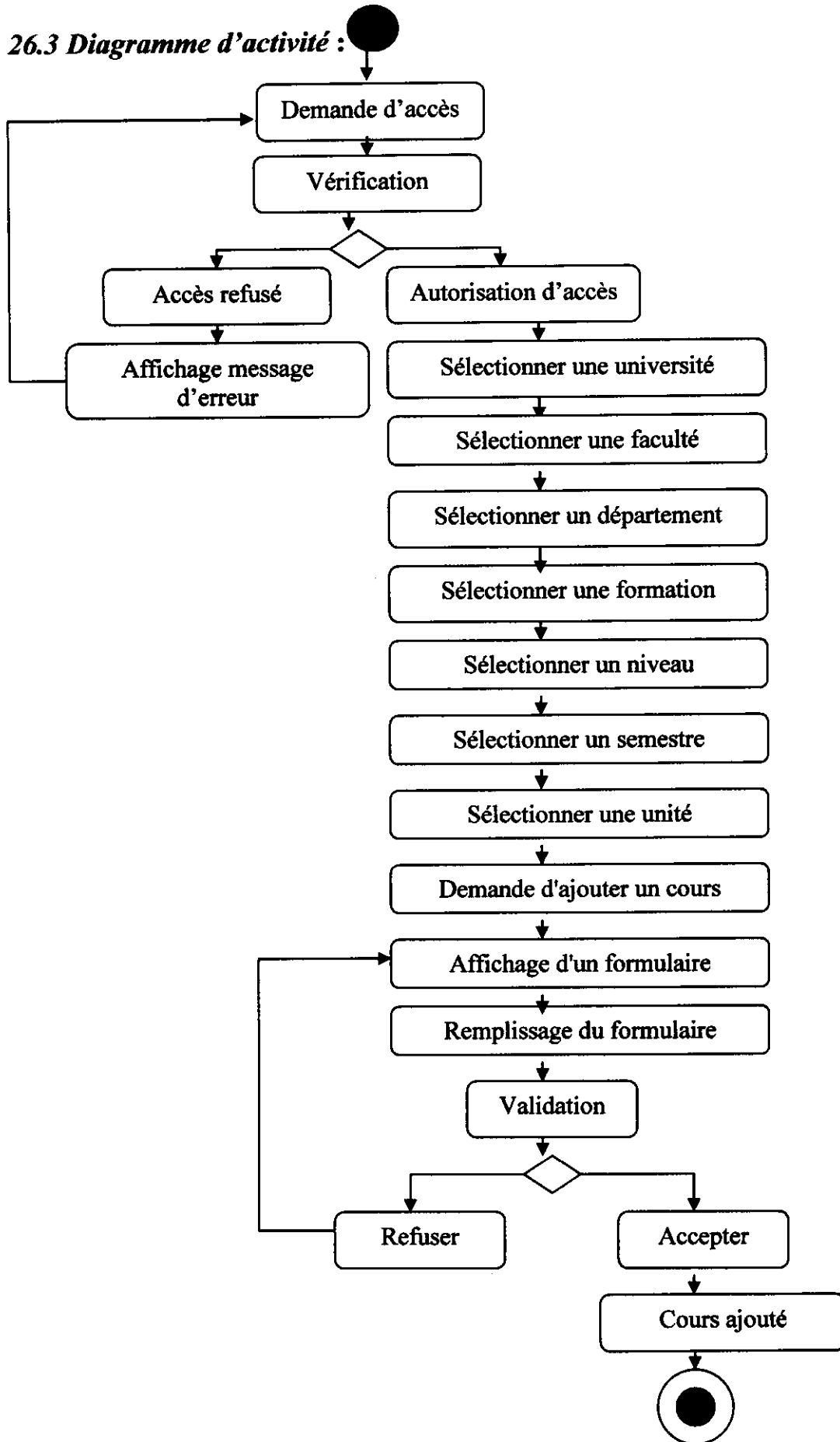
❖ L'enseignant :**26. Cas d'utilisation «Créer un cours » :****26.1 Scénario:**

1. L'enseignant envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'enseignant.
4. L'enseignant sélectionne une université.
5. L'enseignant sélectionne une faculté.
6. L'enseignant sélectionne un département.
7. L'enseignant sélectionne une formation.
8. L'enseignant sélectionne un niveau.
9. L'enseignant sélectionne un semestre.
10. L'enseignant sélectionne une unité d'enseignant.
11. L'enseignant demande d'ajouter un cours.
12. Le système affiche un formulaire vide.
13. L'enseignant saisit les paramètres du cours puis valide.
14. Le système contrôle les données saisies et crée le cours.

26.2 Diagramme de séquence:



26.3 Diagramme d'activité :



27. Cas d'utilisation «Gestion de ses cours » :**27.1 Scénario:**

1. L'enseignant envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'enseignant.
4. L'enseignant demande d'accéder à l'un de ses cours.
5. Le système affiche la page.

A. Ajouter un texte :

6. L'enseignant demande d'ajouter un texte.
7. Le système affiche un formulaire.
8. L'enseignant écrit le texte, puis valide.
9. Le système ajoute le texte.

B. Faire la description d'un cours :

6. L'enseignant demande de faire la description du cours.
7. Le système affiche un formulaire.
8. L'enseignant remplit le formulaire, puis valide.
9. Le système ajoute la description.

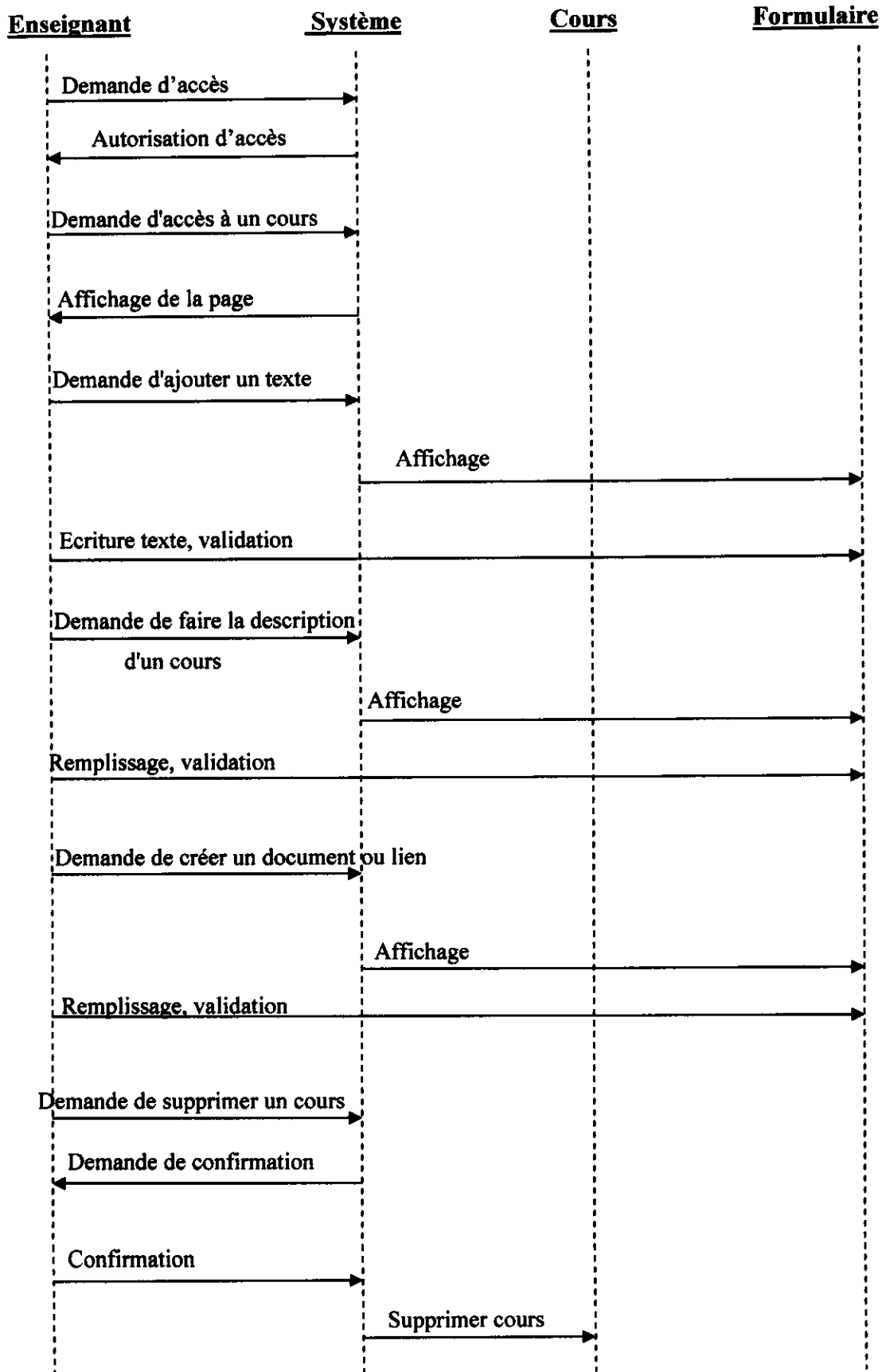
C. Créer des documents et des liens :

6. L'enseignant demande de créer un document ou un lien.
7. Le système affiche un formulaire.
8. L'enseignant remplit le formulaire, puis valide.
9. Le système ajoute le document ou le lien.

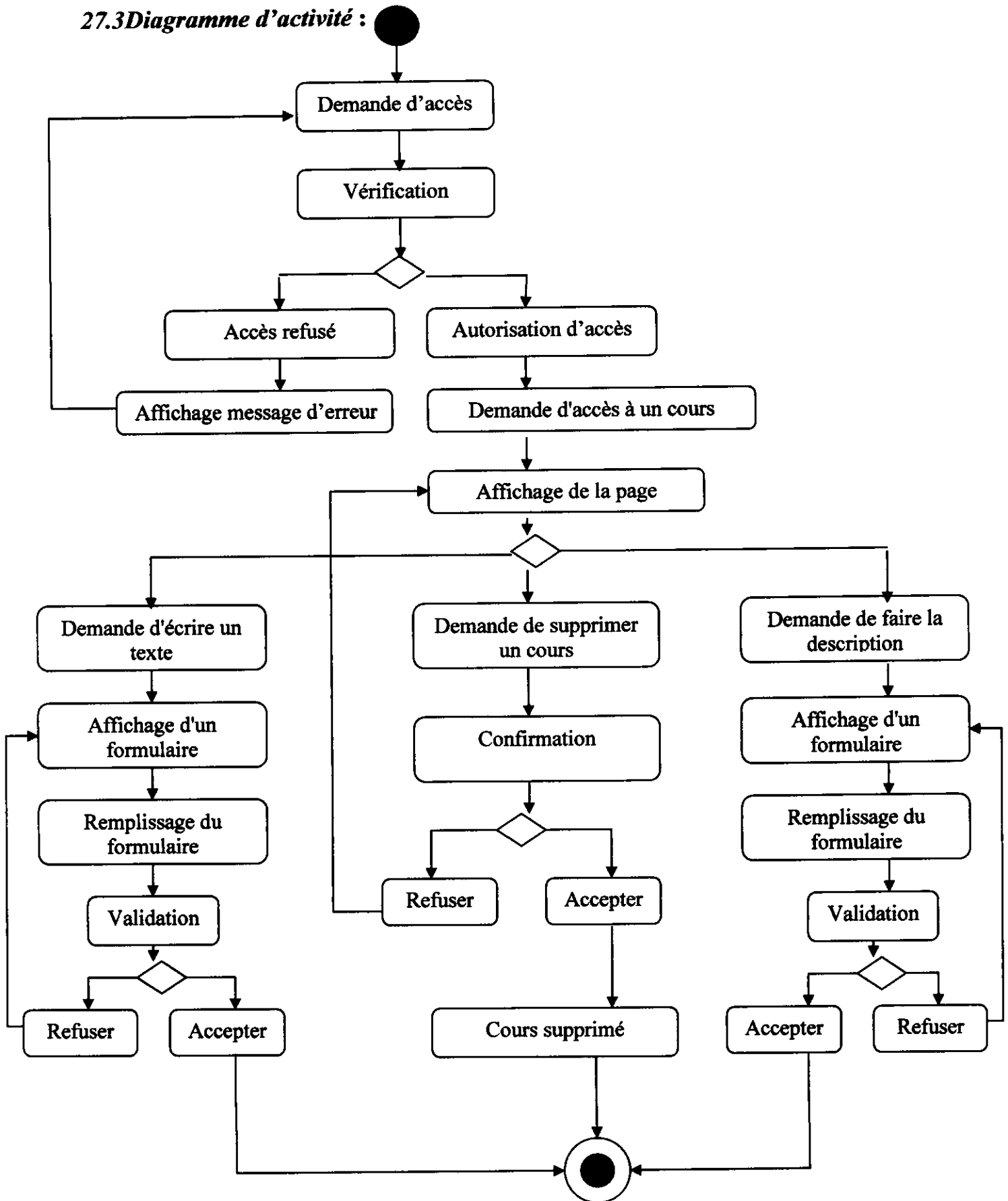
D. Supprimer un cours :

6. L'enseignant demande de supprimer un cours.
7. Le système demande la confirmation.
8. L'enseignant confirme la suppression.
9. Le système supprime le cours.

27.2 Diagramme de séquence:



27.3 Diagramme d'activité :



28. Cas d'utilisation «Préparer les examens » :**28.1 Scénario:**

1. L'enseignant envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'enseignant.
4. L'enseignant demande d'accéder à l'un de ses cours.
5. Le système affiche la page.
6. L'enseignant demande d'accéder aux examens.
7. Le système affiche la page.
8. L'enseignant demande d'ajouter un examen.
9. Le système affiche un formulaire.
10. L'enseignant remplit le formulaire, puis valide.
11. Le système ajoute l'examen.

A. Ajouter un exercice :

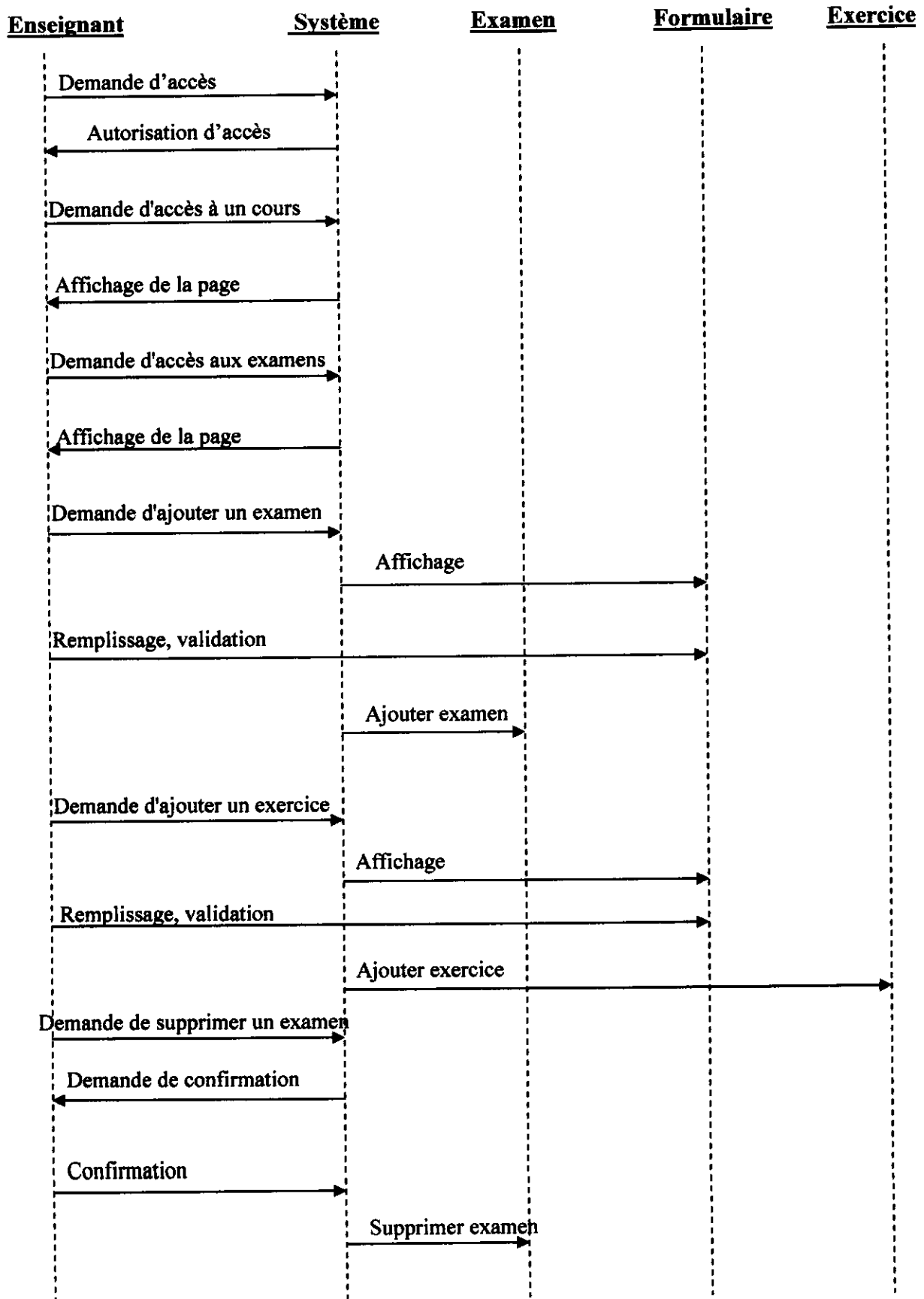
12. L'enseignant demande d'ajouter un exercice à un examen.
13. Le système affiche un formulaire avec les types de réponses proposées.
14. L'enseignant remplit le formulaire et fait le choix du type de réponse, puis valide.
15. L'enseignant édite les réponses avec l'évaluation.
16. Le système ajoute l'exercice.

B. Supprimer un examen :

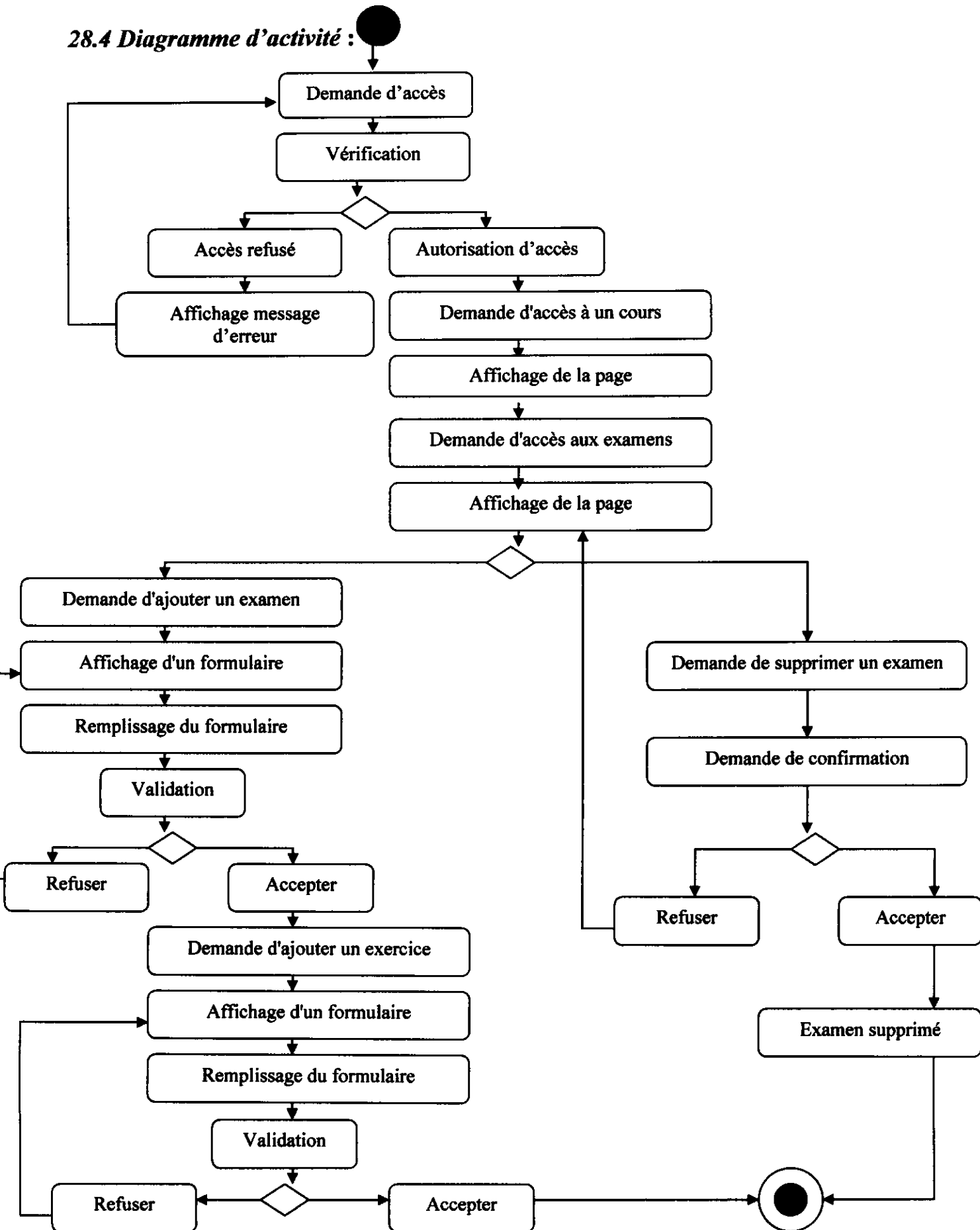
12. L'enseignant demande de supprimer un examen.
13. Le système demande la confirmation.
14. L'enseignant confirme la suppression.
15. Le système supprime l'examen.

28.2 Diagramme de classe :

28.3 Diagramme de séquence:



28.4 Diagramme d'activité :



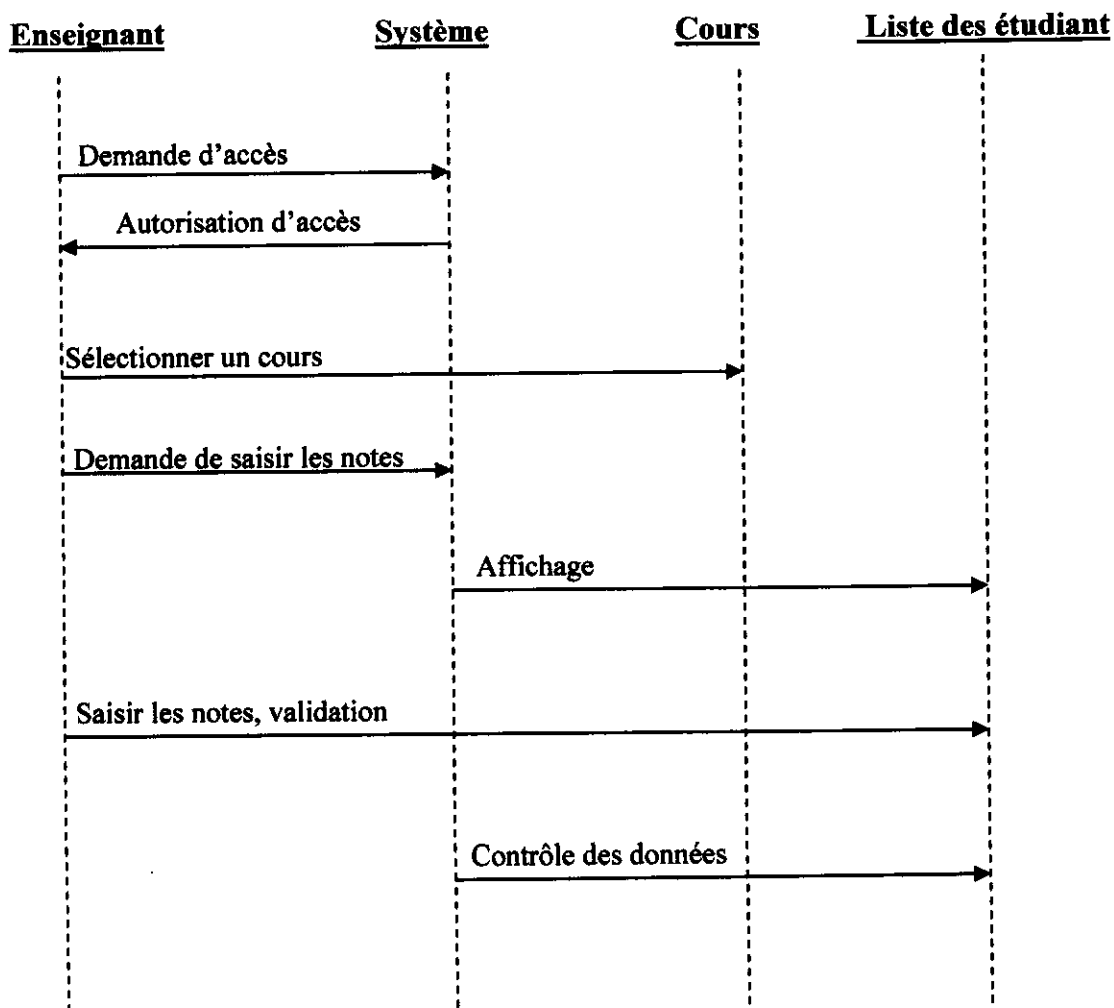
29. Cas d'utilisation «Saisir les notes des examens » :

29.1 Scénario:

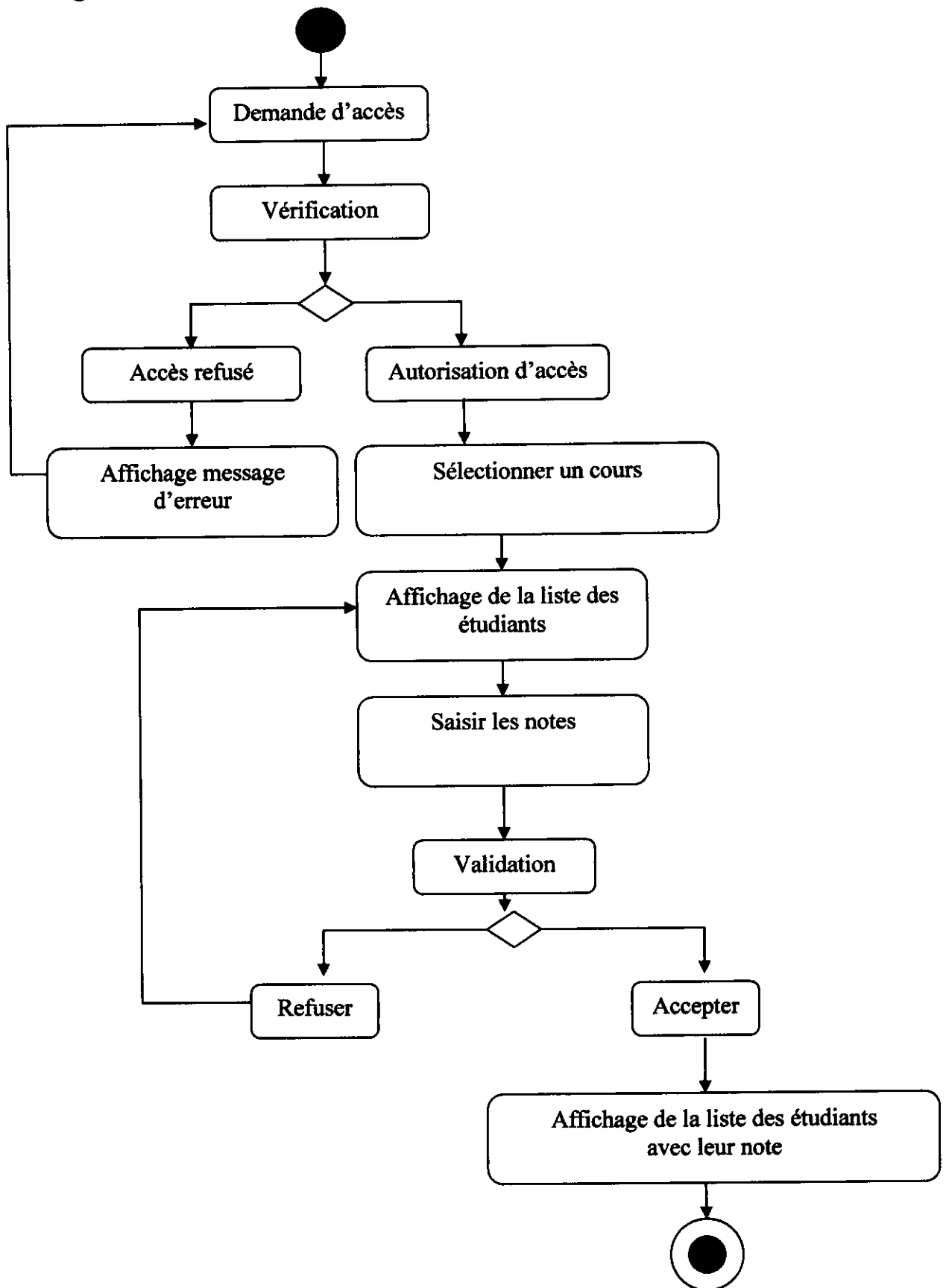
1. L'enseignant envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'enseignant.
4. L'enseignant sélectionne un cours dont il a le droit de saisir les notes.
5. Le système affiche la liste des étudiants.
6. L'enseignant saisit les notes, puis valide.
7. Le système contrôle les notes saisies et affiche la liste des étudiants avec leur note.

L'enseignant peut à tout moment modifier les notes, comme ça l'étudiant peut faire un recours en envoyant un message à l'enseignant.

29.2 Diagramme de séquence:



29.3 Diagramme d'activité :



Remarque:

Le reste des cas d'utilisation de l'enseignant sont identiques à ceux de l'administrateur sauf pour les dettes, l'enseignant peut seulement consulter les dettes et c'est à l'administrateur de les supprimer.

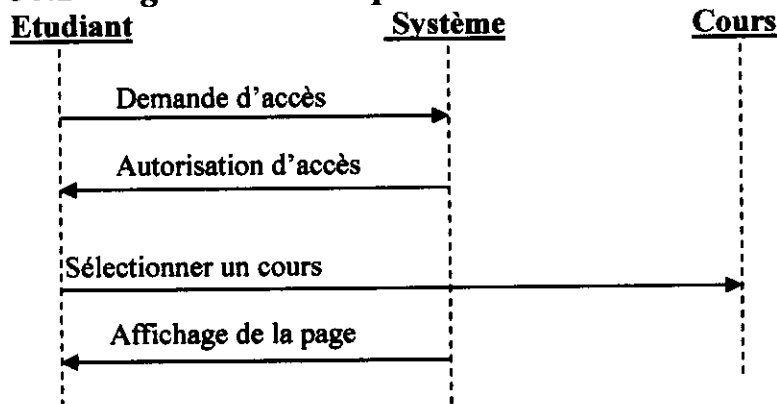
❖ L'étudiant :

30. Cas d'utilisation «Accède à ses cours » :

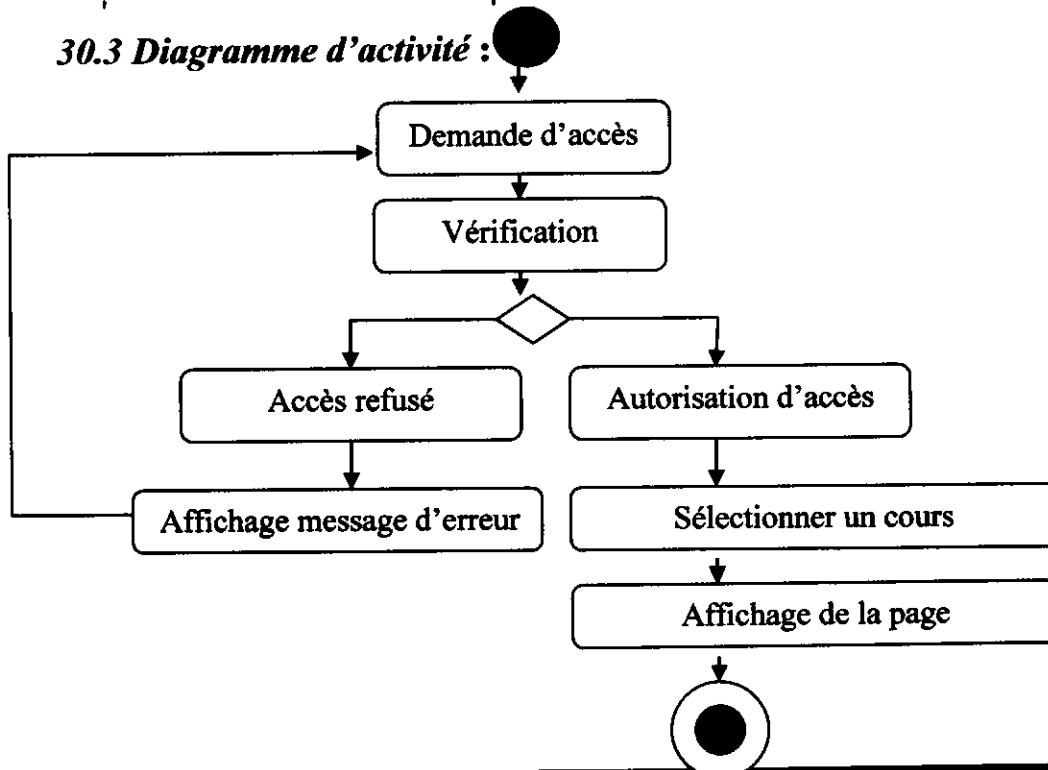
30.1 Scénario:

1. L'étudiant envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'étudiant.
4. L'étudiant sélectionne un de ses cours.
5. Le système affiche la page.

30.2 Diagramme de séquence:

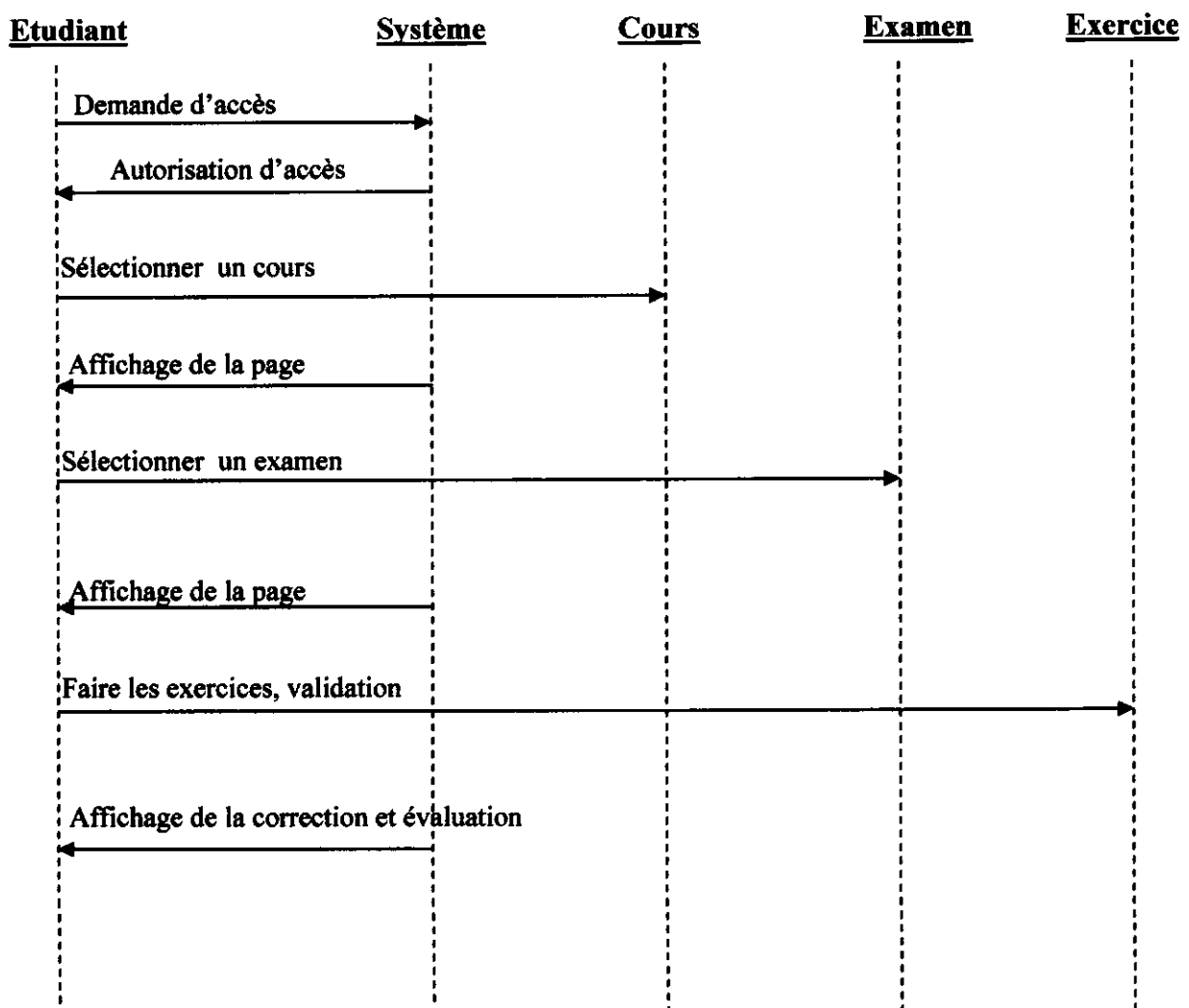


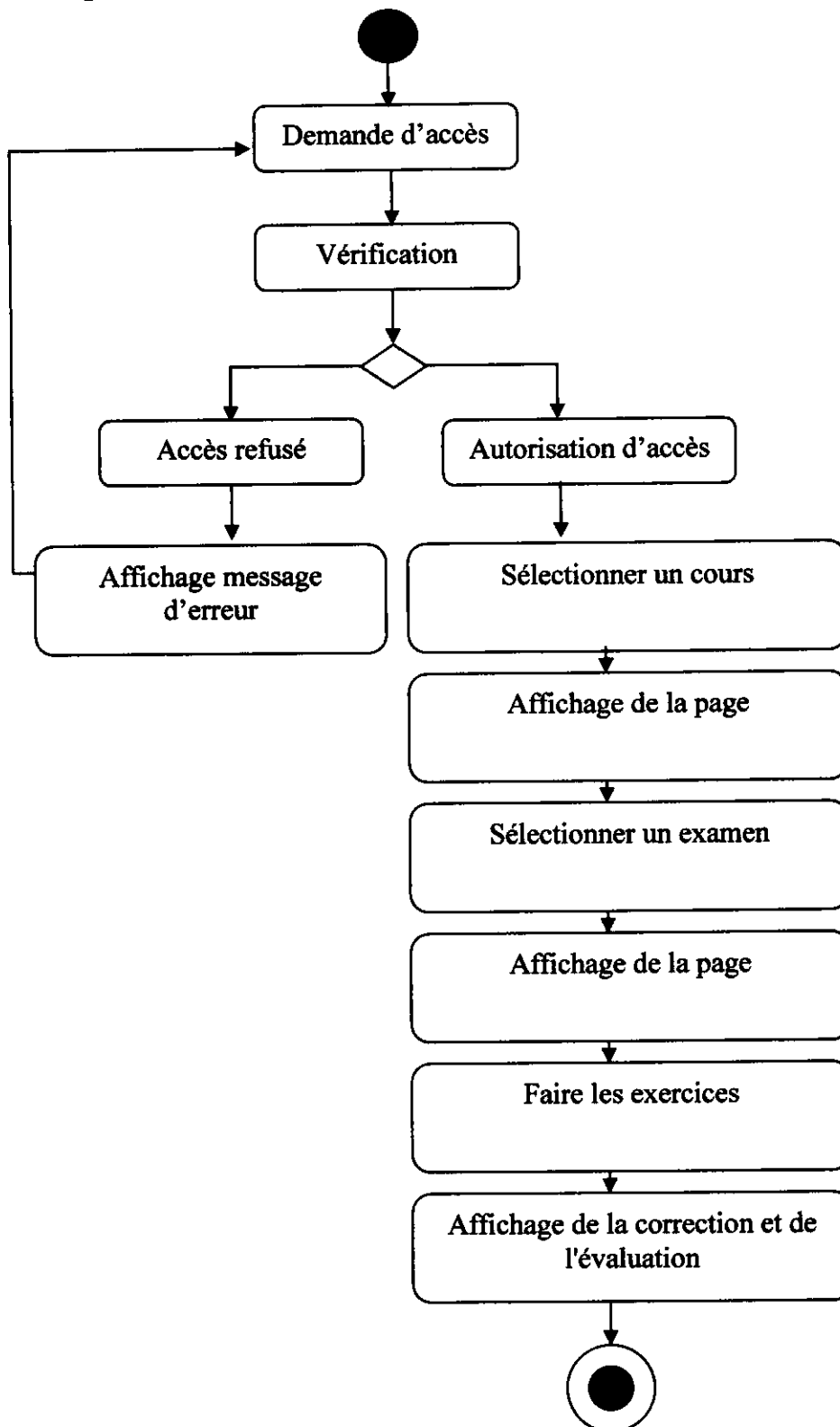
30.3 Diagramme d'activité :



31. Cas d'utilisation « Passer les examens » :**31.1 Scénario:**

1. L'étudiant envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'étudiant.
4. L'étudiant sélectionne un de ses cours.
5. Le système affiche la page.
6. L'étudiant sélectionne un examen.
7. Le système affiche la page.
8. L'étudiant fait les exercices, puis valide.
9. Le système affiche la correction et l'évaluation.

**31.2 Diagramme de séquence:**

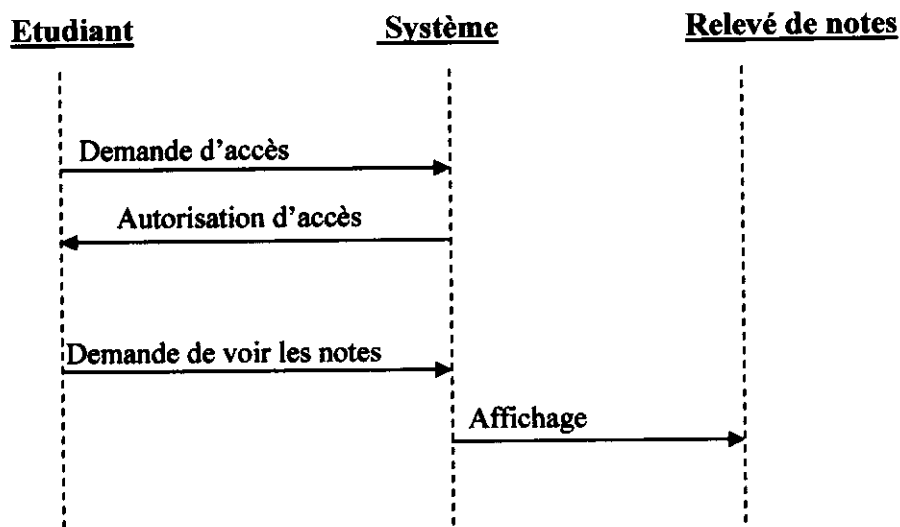
31.3 Diagramme d'activité :

32. Cas d'utilisation « Consulter le relevé de notes » :

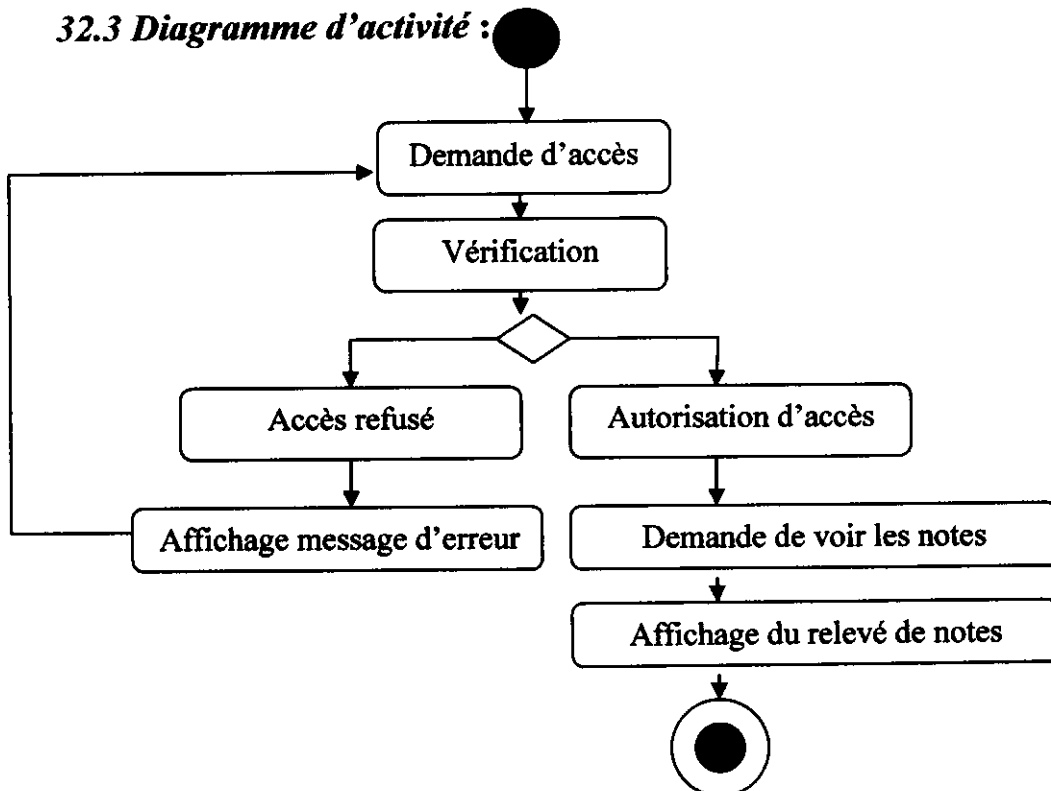
32.1 Scénario:

1. L'étudiant envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'étudiant.
4. L'étudiant demande de voir ses notes.
5. Le système affiche la page.

32.2 Diagramme de séquence:

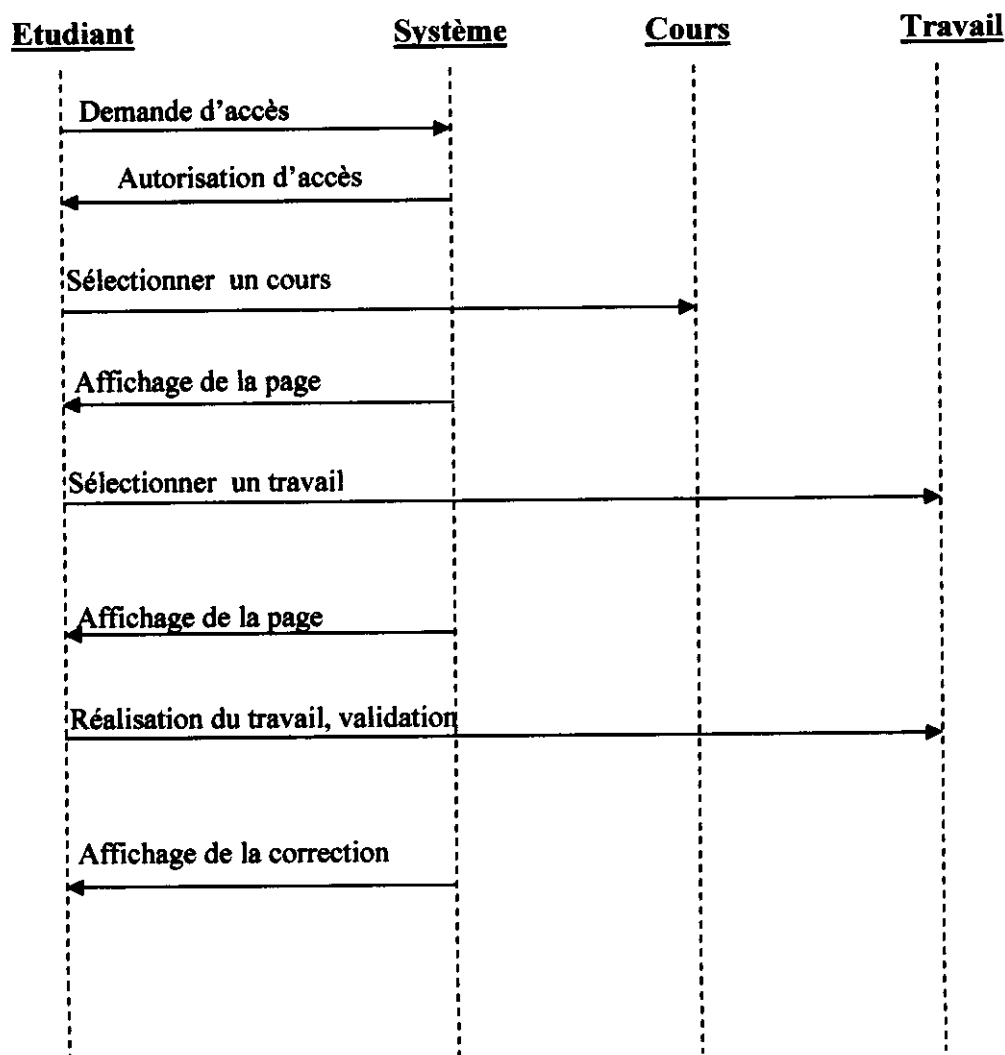


32.3 Diagramme d'activité :

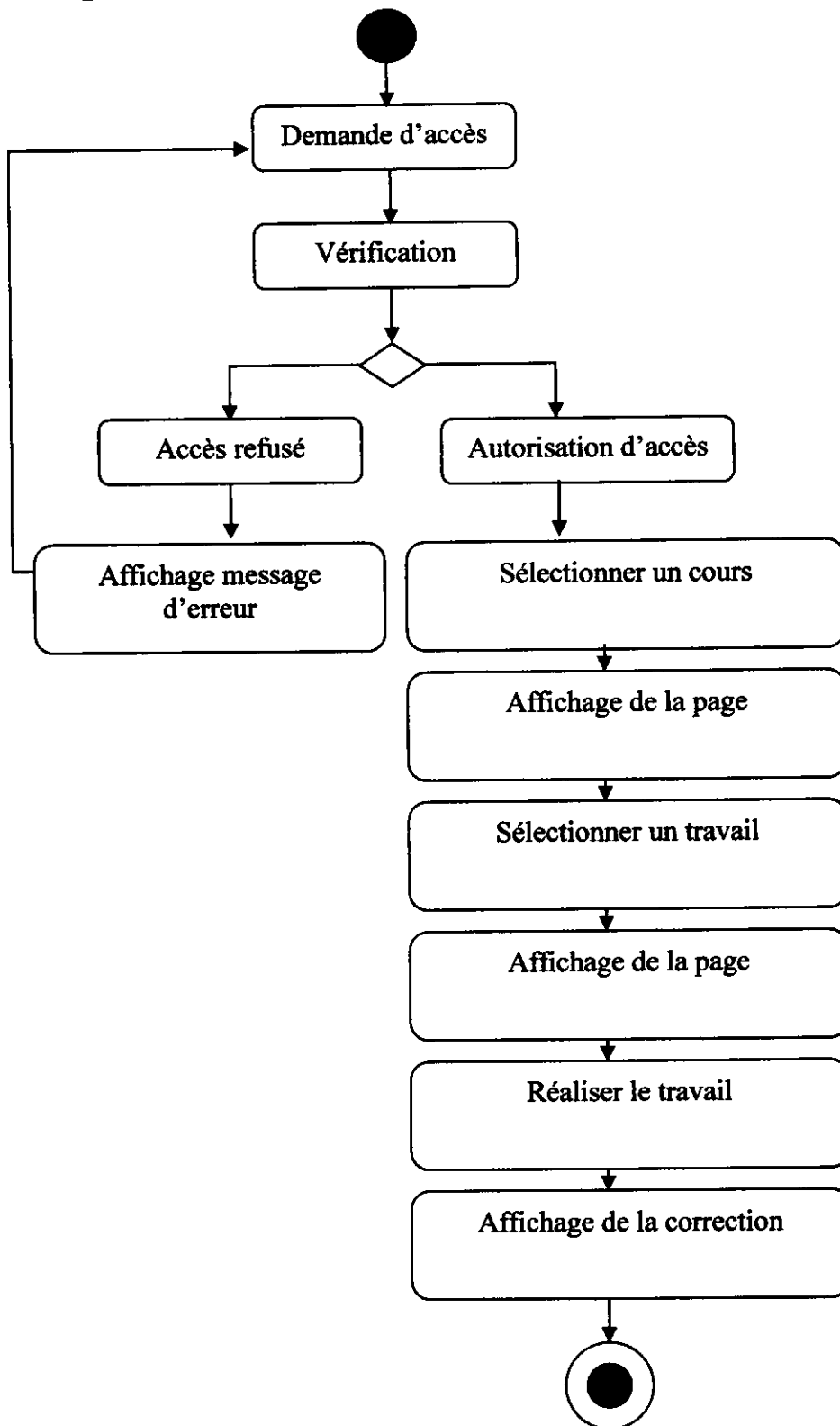


33. Cas d'utilisation «Réaliser les travaux » :**33.1 Scénario:**

1. L'étudiant envoie ses paramètres d'identification.
2. Le système vérifie cette identité.
3. Le système donne l'accès à l'étudiant.
4. L'étudiant sélectionne un de ses cours.
5. Le système affiche la page.
6. L'étudiant sélectionne un travail.
7. Le système affiche la page.
8. L'étudiant réalise le travail, puis valide.
9. Le système affiche la correction.

33.2 Diagramme de séquence:

33.3 Diagramme d'activité :



Remarque:

Pour l'agenda et les annonces l'étudiant peut seulement faire des consultations et ne peut que répondre aux forums.

❖ Visiteur :**34. Cas d'utilisation «Consulter la liste des facultés d'une université » :****34.1 Scénario:**

1. Sélectionner une université.
2. Le système affiche la liste des facultés.

35. Cas d'utilisation «Consulter la liste des départements d'une faculté » :**35.1 Scénario:**

1. Sélectionner une faculté.
2. Le système affiche la liste des départements.

36. Cas d'utilisation «Consulter la liste des formations d'un département» :**36.1 Scénario:**

1. Sélectionner un département.
2. Le système affiche la liste des formations.

37. Cas d'utilisation «Consulter la liste des niveaux d'une formation» :**37.1 Scénario:**

1. Sélectionner une formation.
2. Le système affiche la liste des niveaux.

38. Cas d'utilisation «Consulter la liste des semestre d'un niveau» :**38.1 Scénario:**

1. Sélectionner un niveau.
2. Le système affiche la liste des semestres.

39. Cas d'utilisation «Consulter la liste des unités d'enseignement d'un semestre» :**39.1 Scénario:**

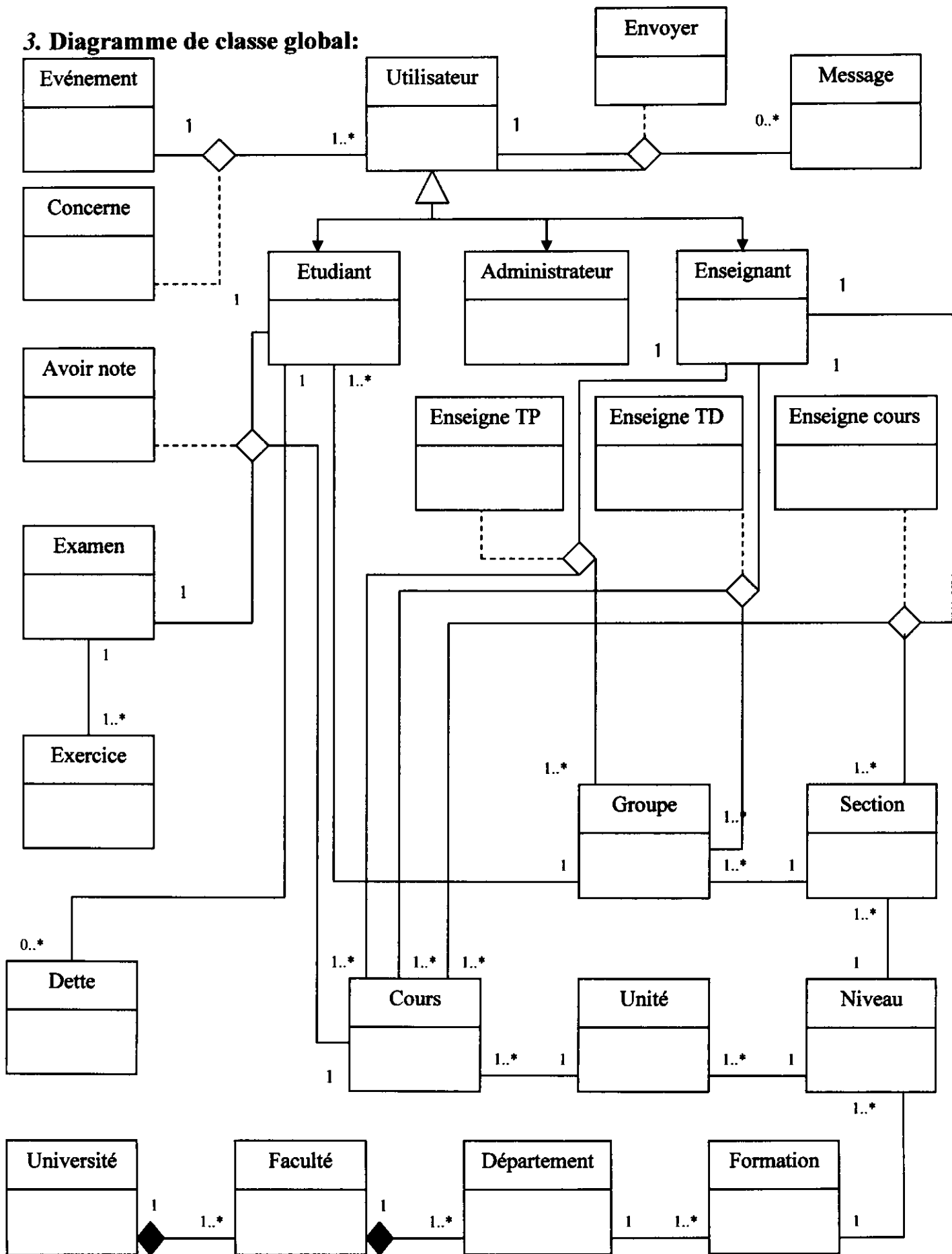
1. Sélectionner un semestre.
2. Le système affiche la liste des unités d'enseignement.

40. Cas d'utilisation «Consulter la liste des cours d'une unité d'enseignement» :

40.1 Scénario:

1. Sélectionner une unité d'enseignement.
2. Le système affiche la liste des cours.

3. Diagramme de classe global:



4. Description du diagramme de classe:

Nom de la classe	Attributs de la classe	Désignation des attributs	Méthodes
Administrateur	id_adm user_adm pass_adm mail_adm nom_adm prénom_adm	Identifiant Username Mot de passe Email Nom Prénom	- Ajouter - Supprimer
Enseignant	id_ens nom_ens prénom_ens user_ens pass_ens mail_ens	Identifiant Nom Prénom Username Mot de passe Email	# Recruter enseignant # Supprimer enseignant # Affecter privilège # Affecter enseignant a un cours
Etudiant	id_et nom_et prénom_et user_et pass_et mail_et code_fo code_niv code_sec code_gr	Identifiant Nom Prénom Username Mot de passe Email Code formation Code niveau Code section Code groupe	# Ajouter étudiant # Supprimer étudiant
Université	code_u nom_u ville_u	Code université Nom de l'université Ville de l'université	# Ajouter université # Supprimer université
Faculté	code_f nom_f code_u	Code faculté Nom de la faculté Code université	# Ajouter faculté # Supprimer faculté
Département	code_d nom_d code_f	Code département Nom du département Code faculté	# Ajouter département # Supprimer département
Formation	code_fo nom_fo code_d	Code formation Nom de la formation Code département	# Ajouter formation # Supprimer formation
Niveau	code_niv nom_niv code_fo	Code niveau Nom du niveau Code formation	# Ajouter niveau # Supprimer niveau

	nb_sem cre_sem	Nombre de semestres Crédit par semestre	
Section	code_sec code_niv num_sec	Code section Code niveau Numéro de section	# Ajouter section # Supprimer section
Groupe	code_gr code_sec num_gr	Code groupe Code section Numéro du groupe	# Ajouter groupe # Supprimer groupe
Unité	code_uni nom_uni code_niv semestre	Code unité Nom de l'unité Code niveau semestre	# Ajouter unité # Supprimer unité
Cours	code_c nom_c code_uni coeff crédit ptd ptp	Code cours Nom du cours Code unité Coefficient Crédit Possède TD Possède TP	# Ajouter cours # Supprimer cours
Dette	code_dette id_et code_c code_niv Note	Code dette Identifiant de l'étudiant Code cours Code niveau note	# Supprimer dette
Examen	code_exa nom_exa code_c type	Code examen Nom de l'examen Code cours type	# Ajouter examen # Supprimer examen
Exercice	code_exo nom_exo code_exa	Code exercice Nom de l'exercice Code examen	# Ajouter exercice # Supprimer exercice
Message	code_mes mess date_mes heure_mes	Code message Le message Date d'envoi du message Heure d'envoi du message	# Ecrire message # Envoyer message # Répondre à un message # Supprimer message

Événement	code_ev eve date_ev heure_ev	Code événement L'événement Date de l'événement Heure de l'événement	# Ajouter événement # Supprimer événement
Enseigne cours	id_ens code_c code_sec	Identifiant de l'enseignant Code cours Code section	
Enseigne TD	id_ens code_c code_gr	Identifiant de l'enseignant Code cours Code groupe	
Enseigne TP	id_ens code_c code_gr	Identifiant de l'enseignant Code cours Code groupe	
Envoyer	id_user1 id_user2 code_mes	Identifiant de l'utilisateur1 Identifiant de l'utilisateur2 Code message	
Concerne	code_ev id_user	Code événement Identifiant de l'utilisateur	
Avoir note	id_et code_exa code_c note	Identifiant de l'étudiant Code examen Code cours Note	

Chapitre 4 :

Implémentation

ET

Réalisation

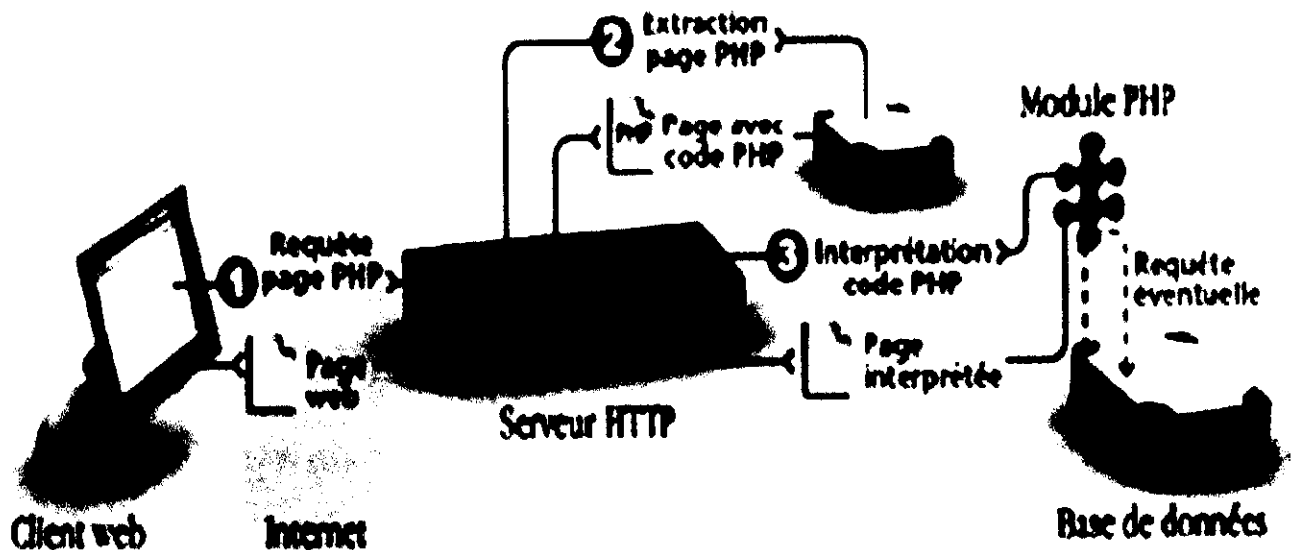
1. Introduction:[w-05]

L'architecture 3-tier est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application ou système à modéliser, et présenter cette application comme un empilement de trois couches dont le rôle est clairement défini :

- ◆ ***la présentation des données*** : correspondant à l'affichage, la restitution sur le poste de travail et le dialogue avec l'utilisateur.
- ◆ ***le traitement métier des données*** : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative.
- ◆ ***l'accès aux données persistantes*** : correspondant aux données qui sont destinées à être conservées longtemps, voire de manière définitive.

Ce modèle d'architecture 3-tier a pour objectif de répondre aux préoccupations suivantes :

- ✓ Allégement du poste de travail.
- ✓ Prise en compte de l'hétérogénéité des plates-formes (serveurs, clients, langages, etc.).
- ✓ Meilleure répartition de la charge entre différents serveurs d'application.
- ✓ L'activation à distance (entre la station et le serveur d'application) des objets et de leurs méthodes.
- ✓ Cette architecture ouverte permet également de répartir les objets sur différents serveurs d'application.

Détail de l'architecture trois tiers

- 1- Un client envoie une requête pour obtenir une page PHP.
- 2- Le serveur http qui traite cette demande extrait la page voulue.
- 3- Le serveur http confie ces pages à son module PHP pour l'interprétation.

A ce stade le serveur peut éventuellement envoyer des requêtes vers une base de données comme MYSQL, la page Web interprétée est retournée au client.

Le fonctionnement de PHP et ses caractéristiques techniques reposent sur le flux depuis le navigateur client qui fait une requête.

2. Les trois couches:

2.1 Couche Présentation (premier niveau):

Elle correspond à l'*Interface Homme Machine*, en informatique, elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée en (HTML) pour être exploitée par un *navigateur Web*.

La couche présentation relaie les requêtes de l'utilisateur à destination de la couche métier et présente les informations renvoyées par les traitements de cette couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche inférieure.

2.2 Couche Métier / Business (second niveau):

Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la « logique », et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs.

Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche.

La couche métier offre des services applicatifs à la couche présentation. Pour fournir ces services, elle s'appuie sur les données du système.

2.3 Couche Accès aux données (troisième niveau):

Elle gère l'accès aux données du système. Ces données peuvent être propres au système, ou gérées par un autre système. La couche métier n'a pas à s'adapter à ces deux cas, ils sont transparents pour elle, et elle accède aux données de manière uniforme.

3. Stratégie de développement:

Il faut savoir que notre projet qui est l'extension d'une plateforme n'est pas une simple configuration de modules existants mais un travail sur le code source de la plateforme, en effet notre projet consiste à étendre un système existant pour l'adapter à un système d'enseignement particulier, pour cela on est passé par les étapes suivantes :

1. L'étude de la plateforme.
2. Déploiement de la plateforme.
3. Etude et compréhension du code source de la plateforme (dans un contexte où il n'y a pas de documentation sur le code source).
4. Après avoir étudié tout le code source on a commencé notre extension par l'élaboration de programmes "greffés" au code source de la plateforme, et ce travail a été fait de manière à s'adapter aux nouvelles versions de la plateforme « Claroline », en effet on a commencé le travail sur la version (1.8.1) et en mois de mars est apparu la version (1.8.4) et le 31 juillet est apparu la version (1.8.6) et sur les deux versions l'extension s'adapte parfaitement.

On a choisit pour illustrer notre travail le système LMD car il va être généralisé dans toutes les universités algériennes, et son fonctionnement est beaucoup plus complexe que l'ancien système, mais on peut adapter notre travail à n'importe quel autre système d'enseignement (lycées, primaires,...).

Notre logiciel permet de concevoir le site d'une ou de plusieurs universités à la fois donc il permet:

- ✓ **La centralisation** : gérer et contrôler toutes les universités, par exemple pour le ministère de l'enseignement supérieur.
- ✓ **La décentralisation** : gérer et contrôler une université.

4. Configuration du système:

Tout d'abord on doit remplacer le fichier «index.php » par celui de l'extension dans le dossier « claroline180 », et ajouter tous les fichier de l'extension dans le même emplacement du fichier «index.php » sauf pour le fichier «affich_fac.inc.php » qui doit être mis dans l'emplacement « claroline180\claroline\inc »

Pour qu'on puisse tester notre projet dans le système d'exploitation Windows, on doit mettre le dossier « claroline180 » dans le répertoire :

C:\Program Files\EasyPHP1-8\www\

Et dans le système d'exploitation Linux, on doit le mettre dans le répertoire

var/www/html

On remarque qu'il y a une différence entre les deux systèmes dans la syntaxe du langage Php.

Le dossier « claroline180 » contient les fichiers de programmation. Parmi lesquels :

- Le fichier «bdclaro.php »

```

1 <?php
2 require './claroline/inc/claro_init_global.inc.php'; // main init
3 include claro_get_conf_repository() . 'CLHOME.conf.php'; // conf file
4 require $includePath . '/claro_init_header.inc.php';
5
6 mysql_connect("localhost", "root", "");
7 mysql_select_db($GLOBALS['mainDbName']);
8
9
10 $sql =mysql_query('CREATE TABLE `universite` ('
11     . ' `id_u` INT(11) NOT NULL AUTO_INCREMENT, '
12     . ' `name_u` VARCHAR(100) NOT NULL, '
13     . ' `code_u` VARCHAR(12) NOT NULL, '
14     . ' `ville_u` VARCHAR(100) NOT NULL, '
15     . ' PRIMARY KEY (`id_u`), '
16     . ' UNIQUE (`code_u`)'
17     . ');');
18
19
20 $sql =mysql_query('CREATE TABLE `faculte` ('
21     . ' `id_f` INT(11) NOT NULL AUTO_INCREMENT, '
22     . ' `name_f` VARCHAR(100) NOT NULL, '
23     . ' `code_f` VARCHAR(12) NOT NULL, '
24     . ' `code_u` VARCHAR(12) NOT NULL, '
25     . ' PRIMARY KEY (`id_f`), '
26     . ' UNIQUE (`code_f`)'
27     . ');');
28
29
30 $sql =mysql_query('CREATE TABLE `departement` ('
31     . ' `id_d` INT(11) NOT NULL AUTO_INCREMENT, '
32     . ' `name_d` VARCHAR(100) NOT NULL, '
33     . ' `code_d` VARCHAR(12) NOT NULL, '
34     . ' `code_f` VARCHAR(12) NOT NULL, '

```

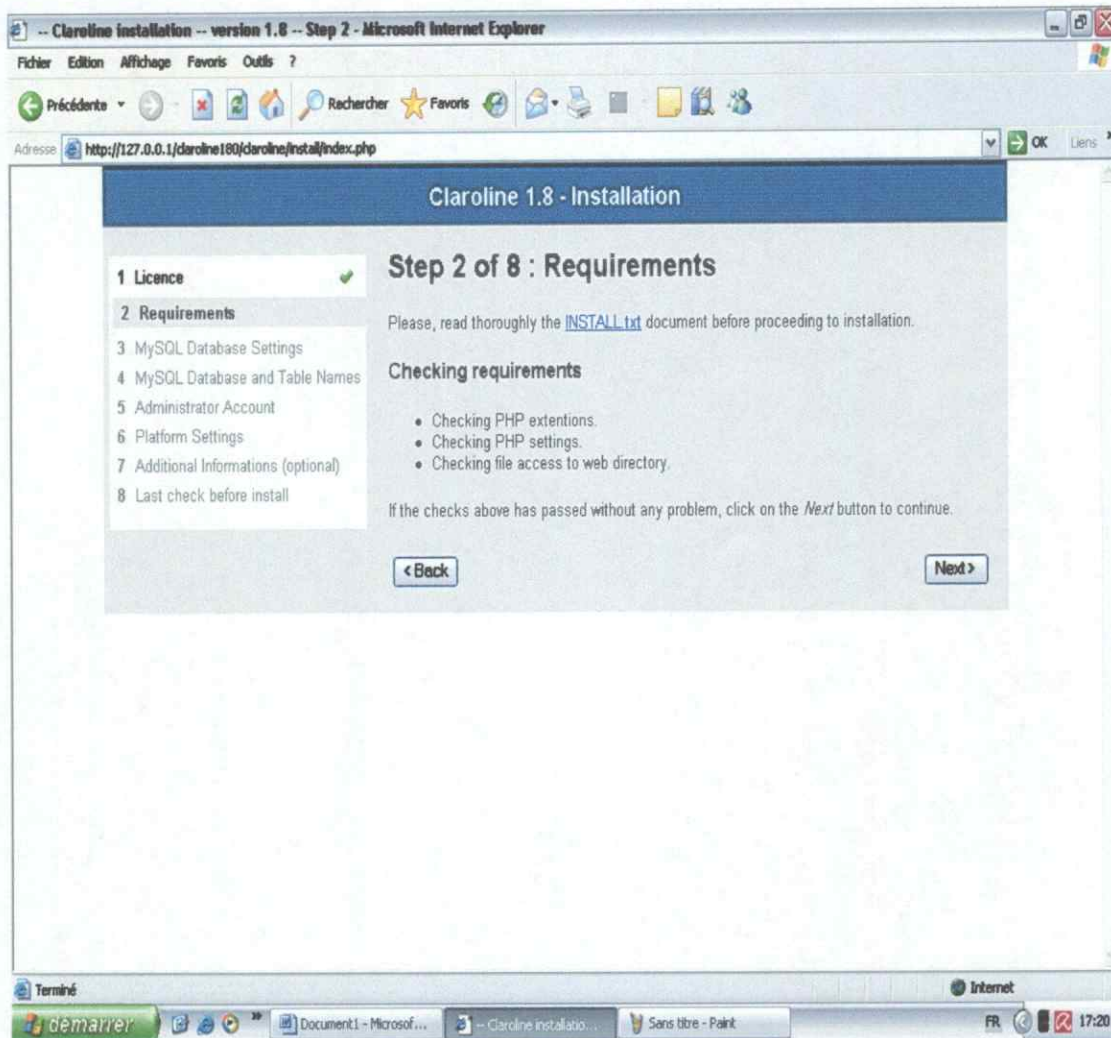
Le fichier : bdclaro.php

bdclaro.php : est un fichier qui sert a créer les tables de la base de données pour l'extension.

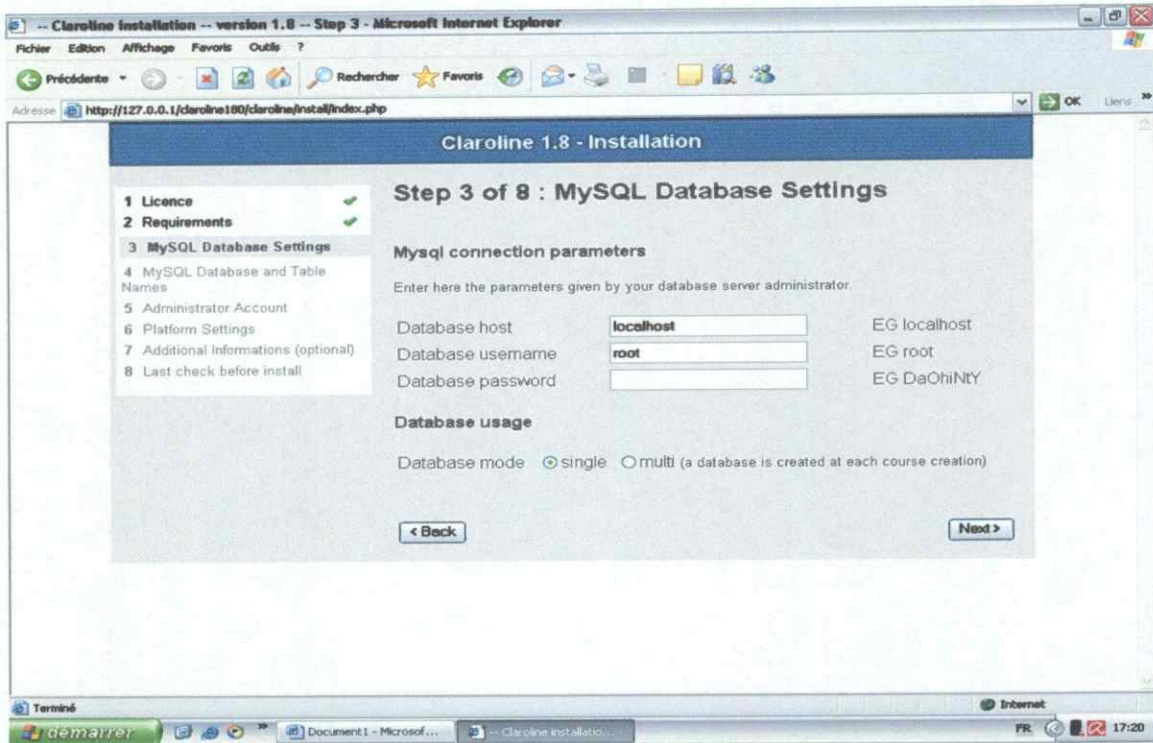
5. Installation du système:

Pour installer le système on doit passer par les étapes suivantes :

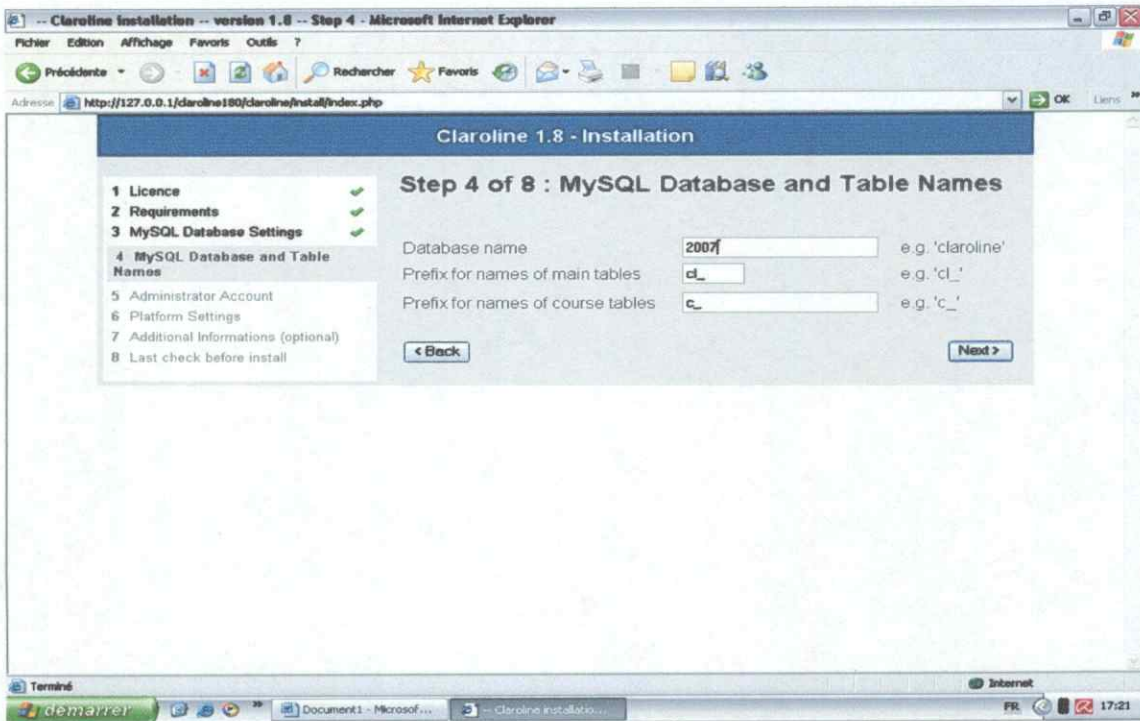
1. Vérification des paramètres de configuration.



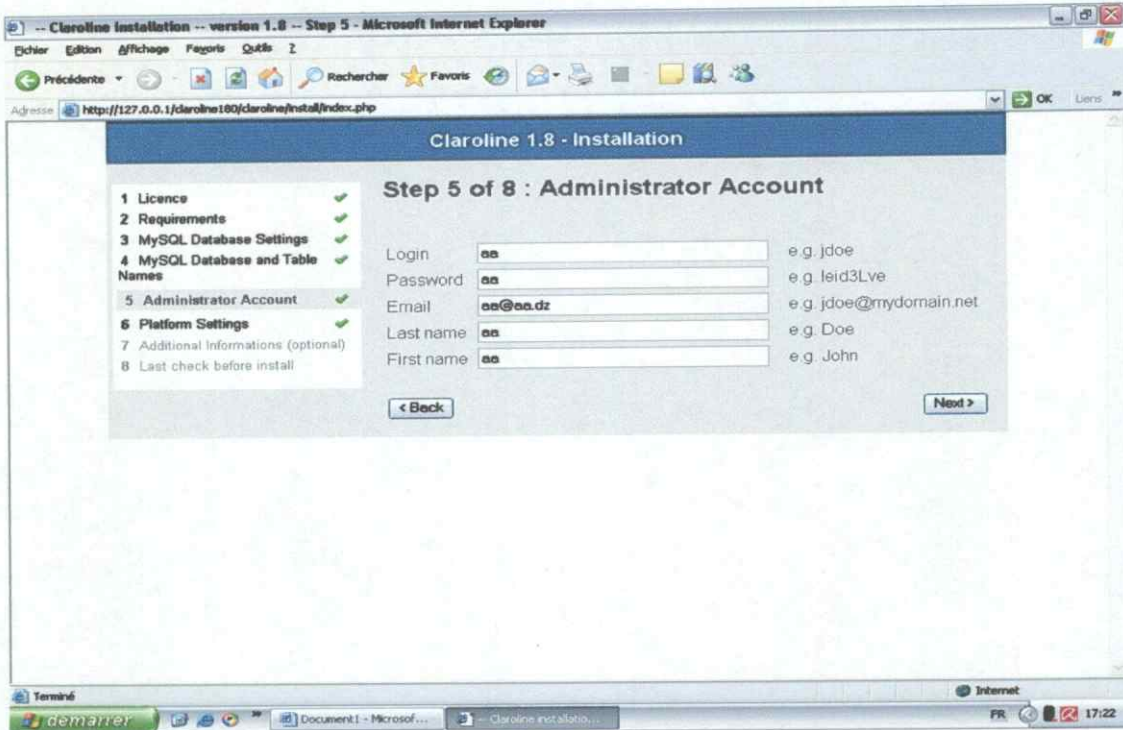
2. Vérification des paramètres relatifs à la base de données Mysql .



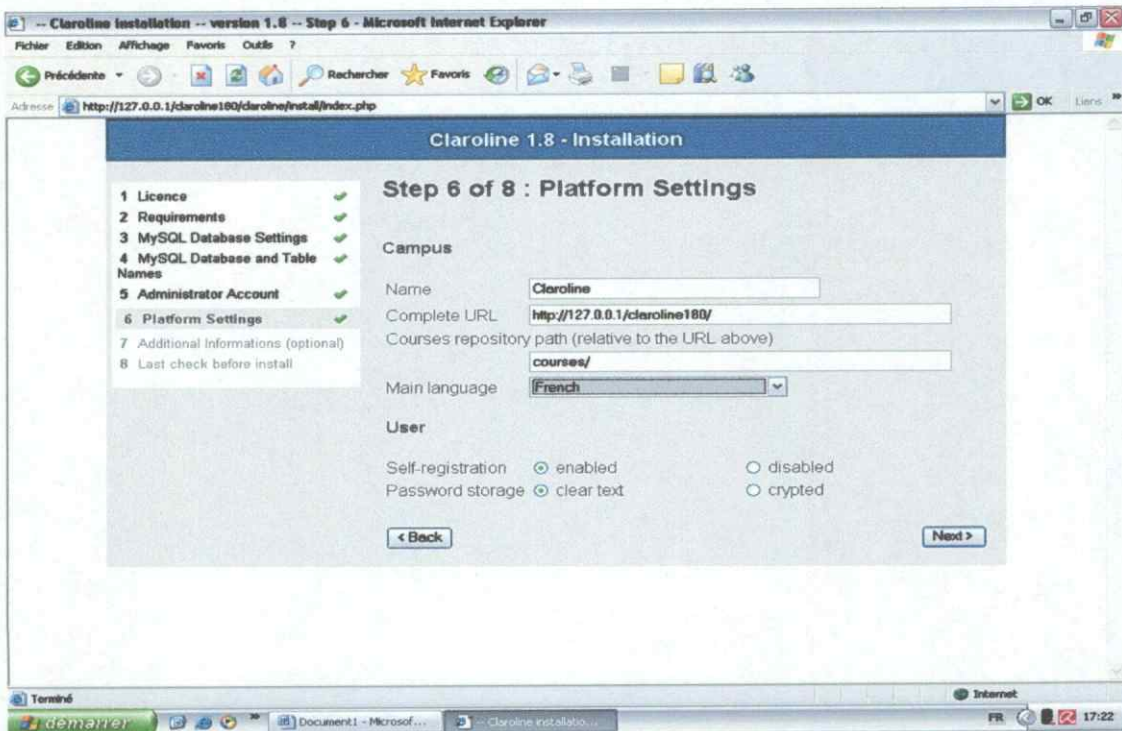
3. Détermination de l'année scolaire.



4. Détermination des paramètres de l'administrateur.



5. Détermination des paramètres de la plateforme.



6. Présentation du logiciel:

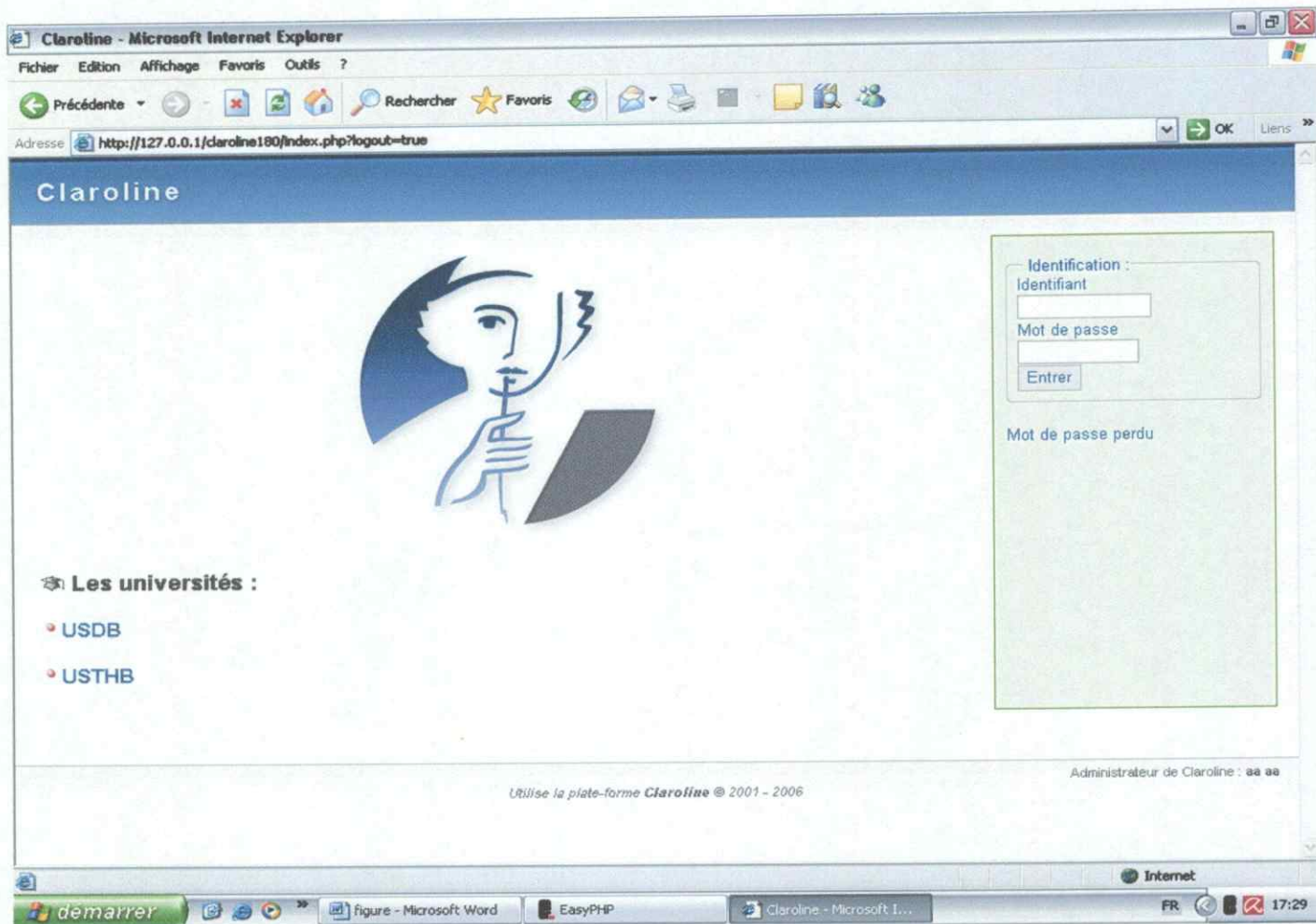


Figure1 : page d'accueil.

La figure 1 représente la page d'accueil du site, n'importe quelle personne peut visiter ce site pour s'informer sur les universités et les formations proposées. Pour cela les universités sont représentées par des liens qui mènent vers les facultés.

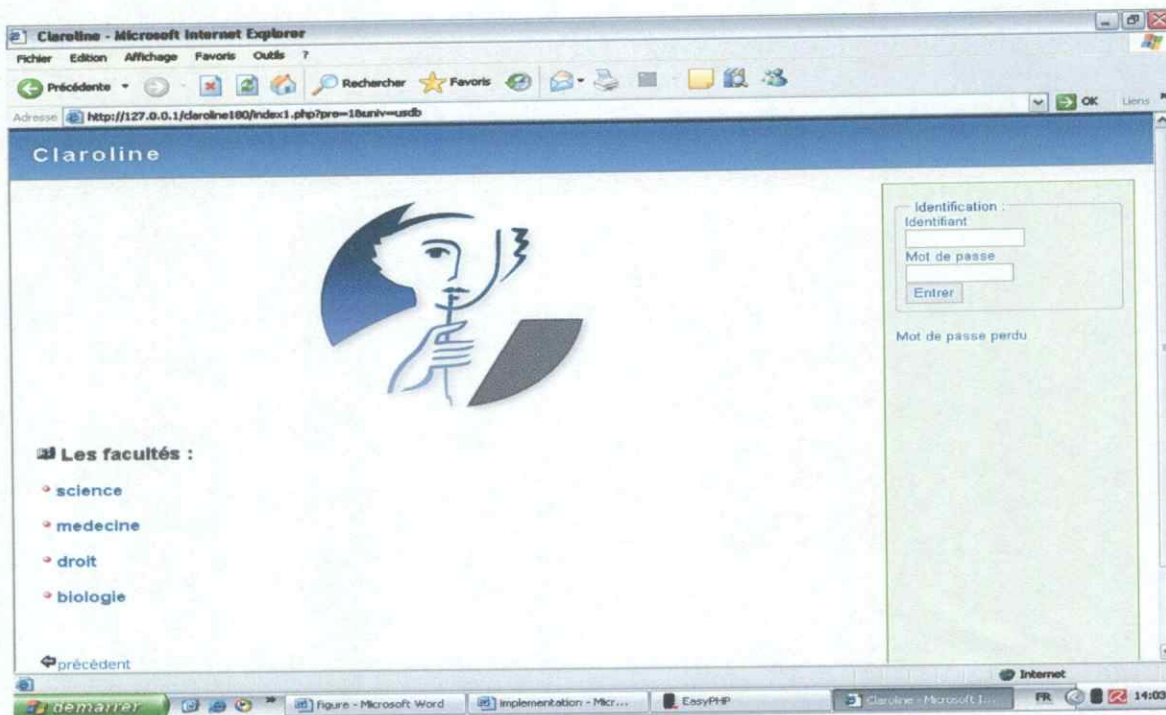


Figure2 : les facultés d'une université.

Les facultés sont représentées par des liens qui mènent vers les départements.

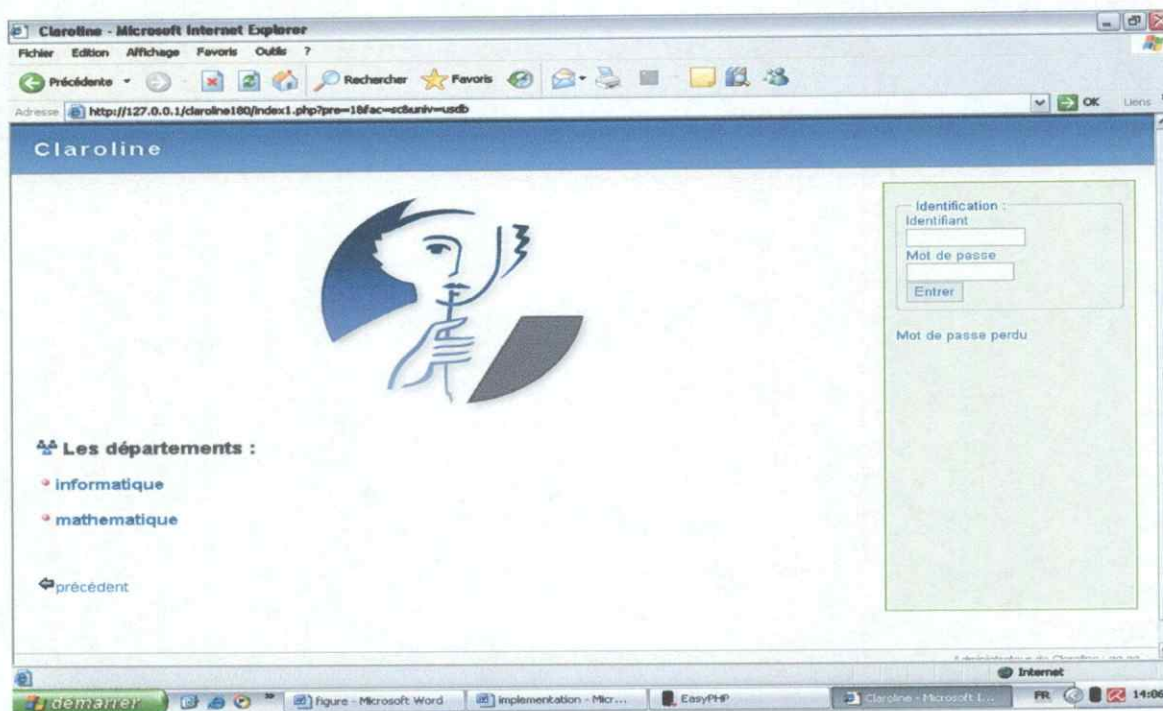


Figure3 : les départements d'une faculté.

Les départements sont représentés par des liens qui mènent vers les formations.

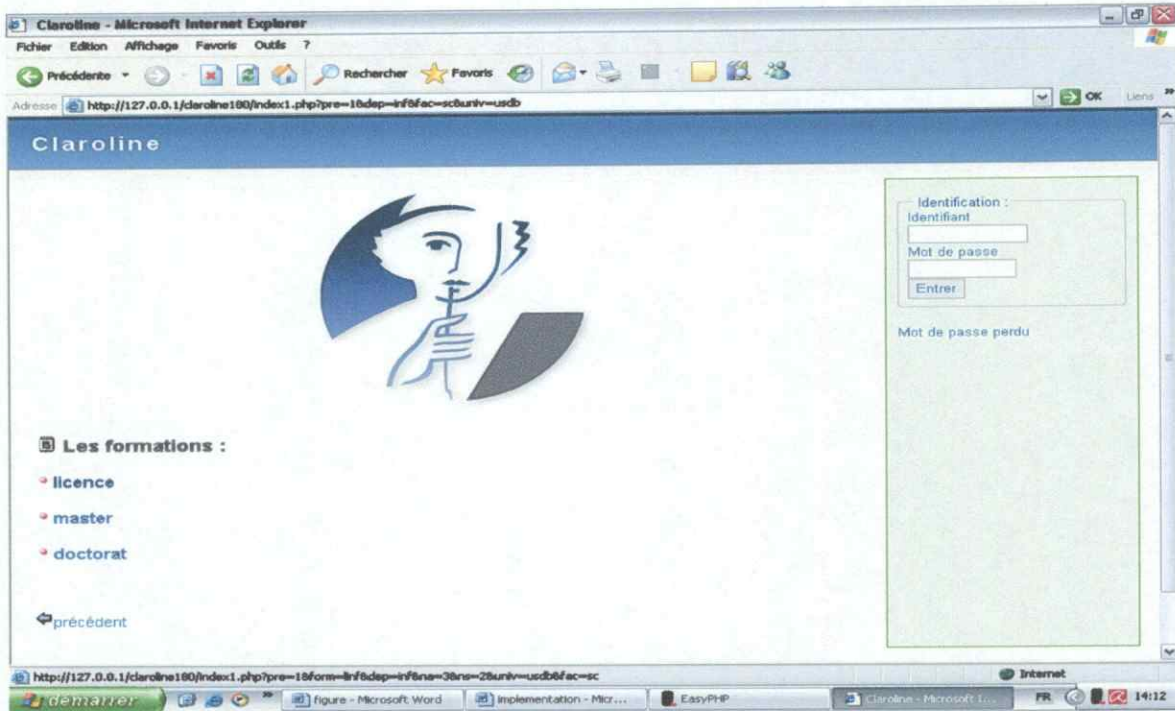


Figure4 : les formations d'un département.

Les formations sont représentées par des liens qui mènent vers les niveaux.

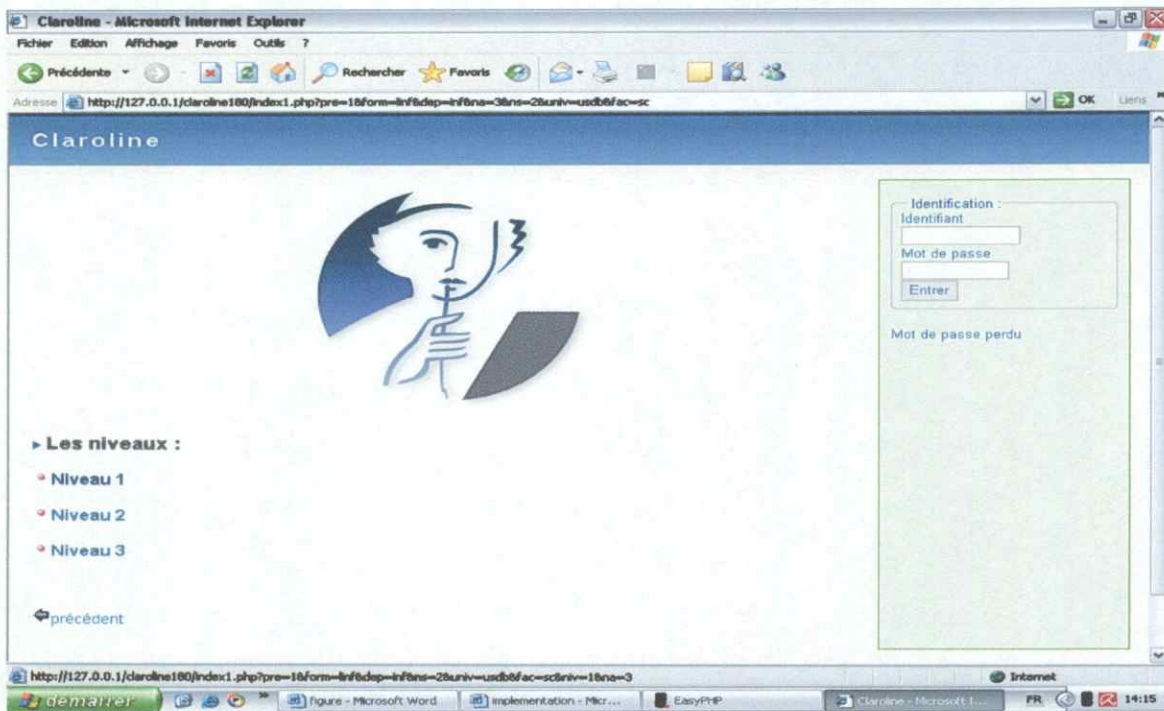


Figure5 : les niveaux d'une formation.

Les niveaux sont représentés par des liens qui mènent vers les semestres.

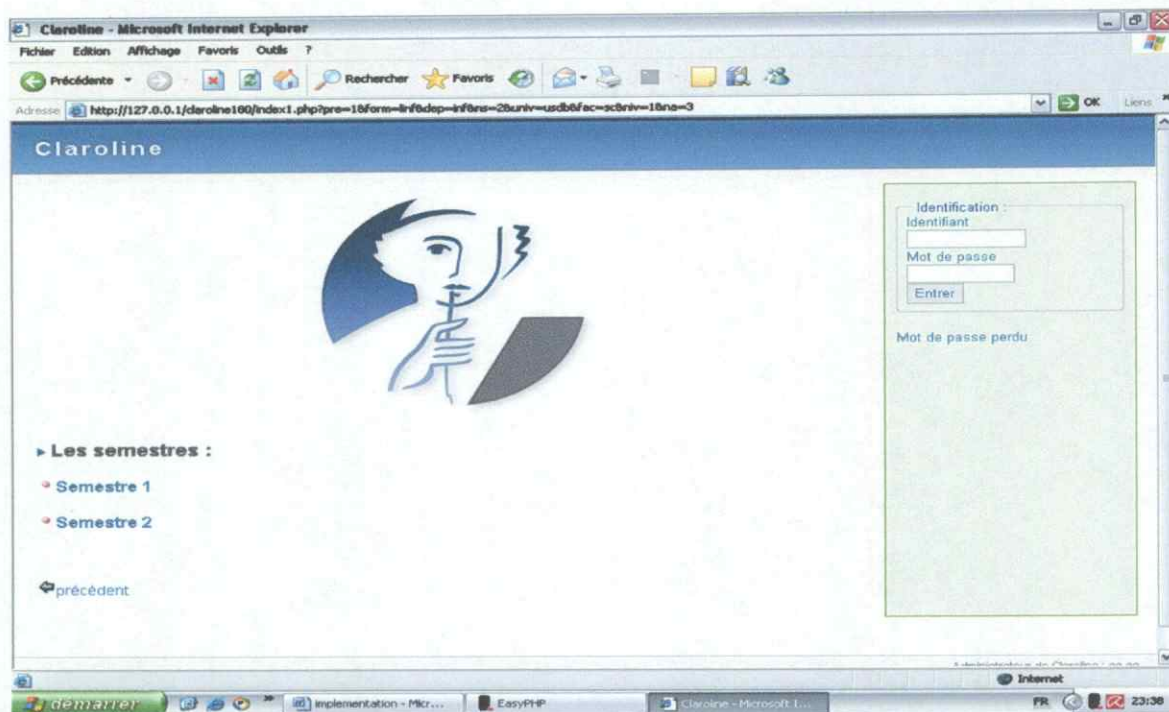


Figure6 : les semestres d'un niveau.

Les semestres sont représentés par des liens qui mènent vers les unités et les cours.

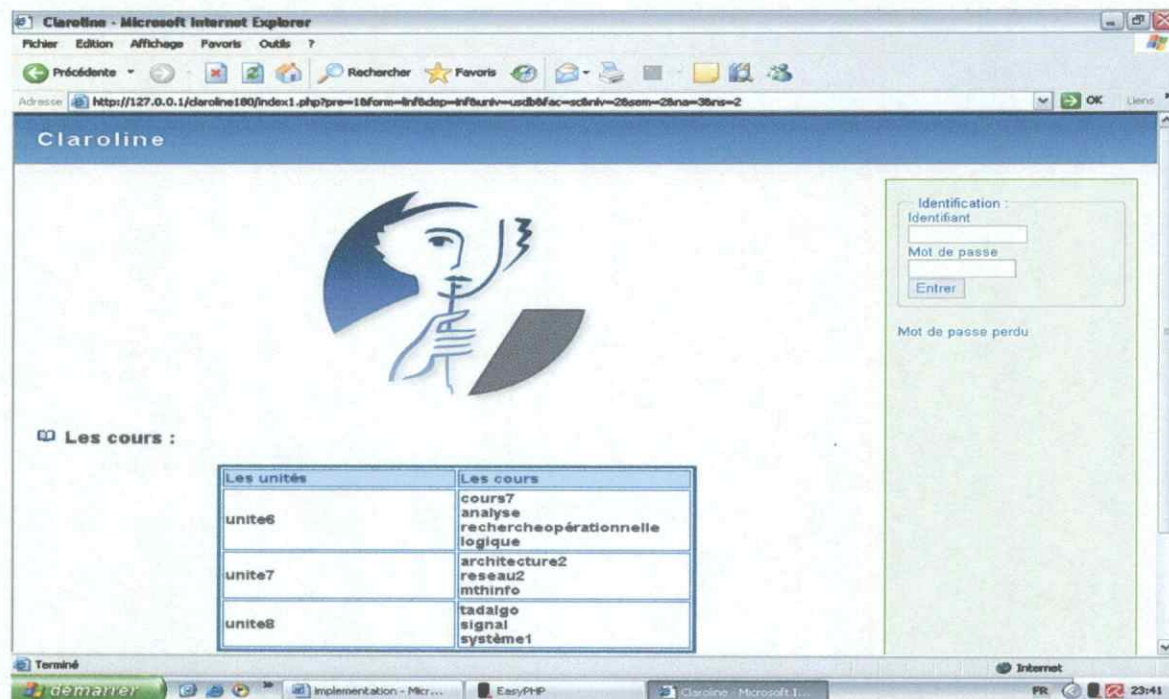


Figure7 : les unités et les cours d'un semestre.

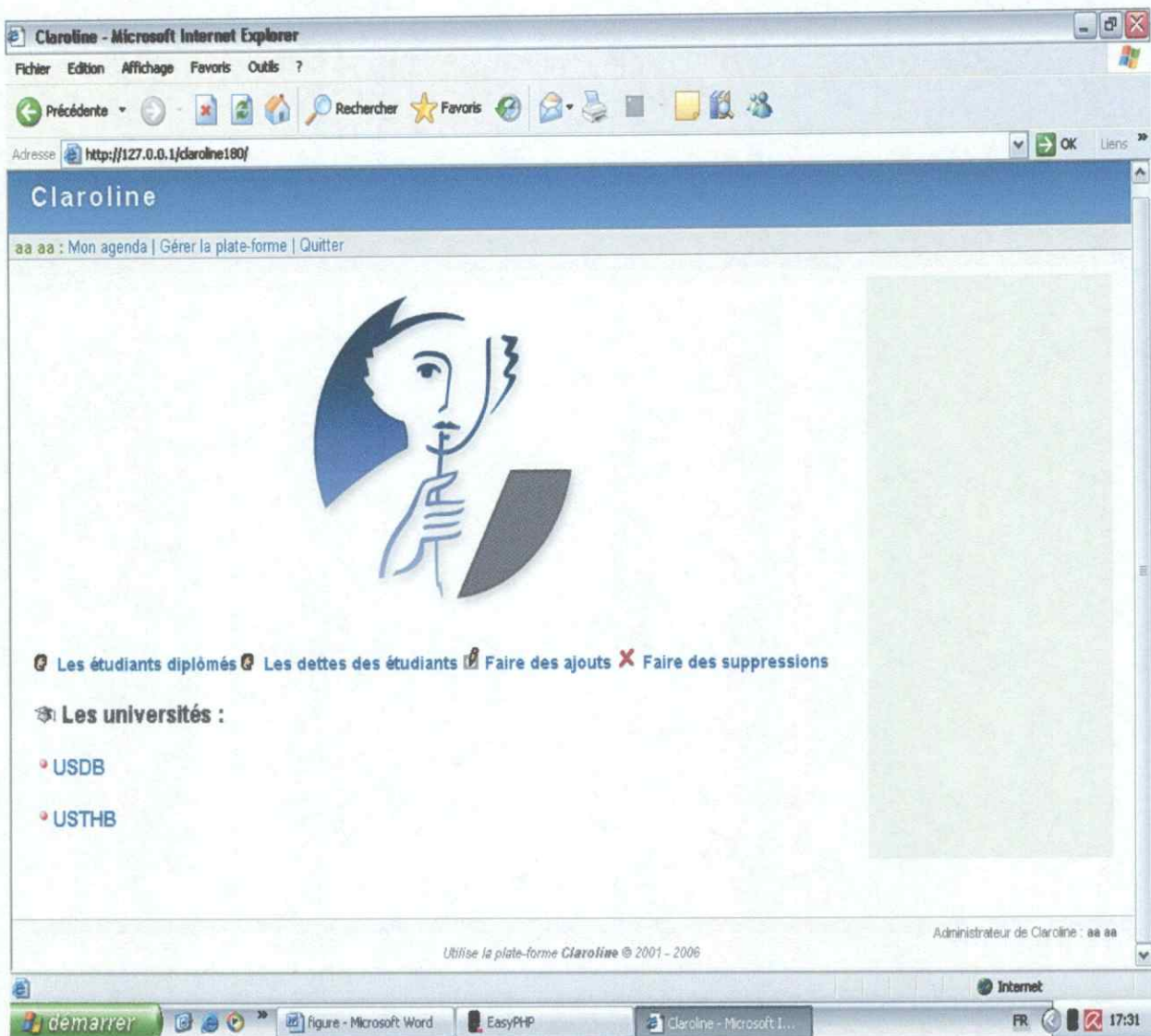


Figure8 : page d'accueil de l'administrateur.

En introduisant l'identifiant et le mot de passe, l'administrateur accède a son espace privé où il peut faire:

- Des ajouts.
- Des suppressions.
- Gérer les étudiants diplômés qui ont des dettes.
- Gérer les dettes des étudiants.
- Gérer la plateforme.
- Consulter l'agenda.
- Consulter la liste des étudiants d'un niveau.
- Consulter la liste des cours d'un semestre.
- Accéder aux cours.
- Consulter le PV des délibérations.

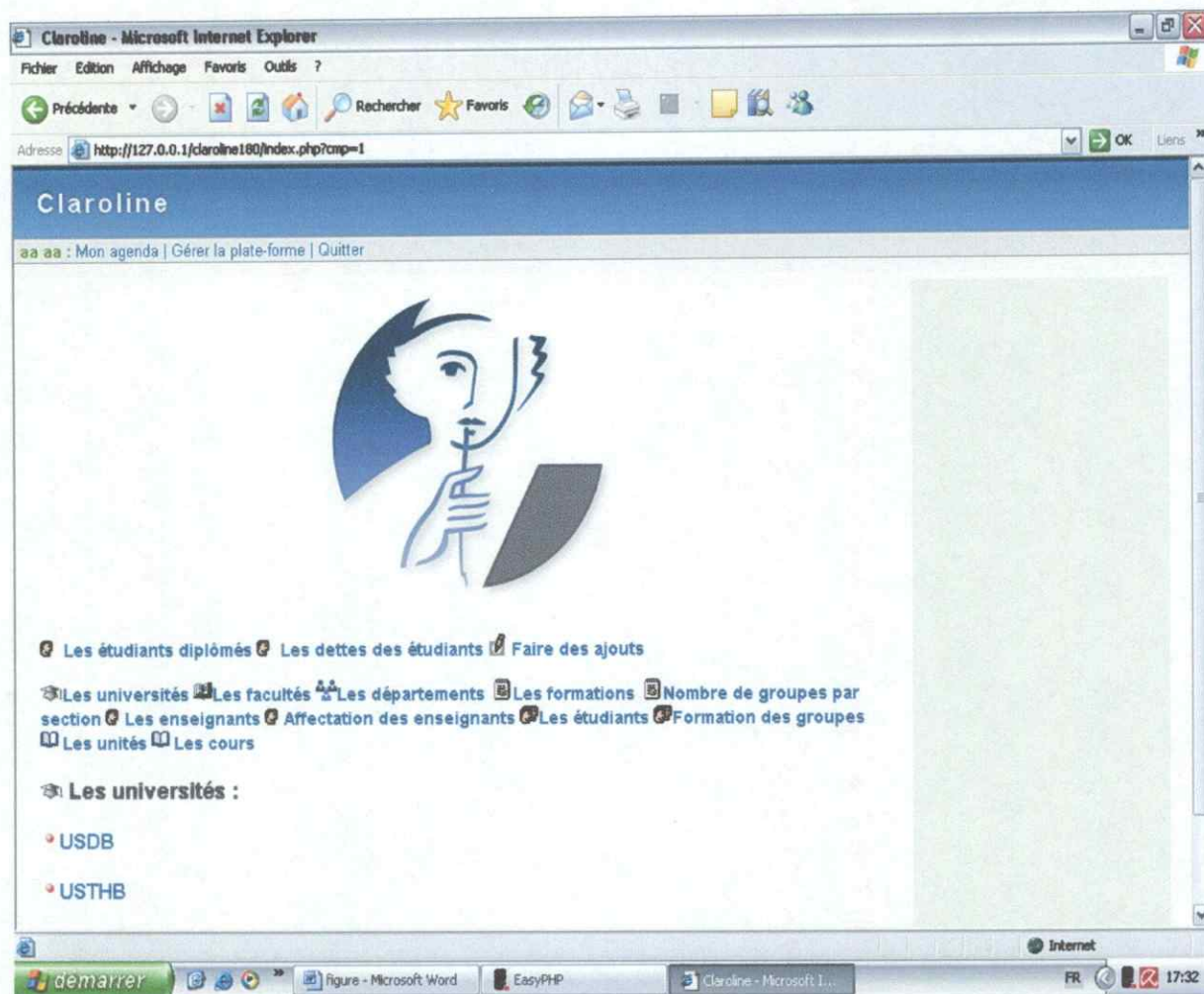


Figure9 : les ajouts.

L'administrateur peut :

- Ajouter une université.
- Ajouter une faculté.
- Ajouter un département.
- Ajouter une formation.
- Déterminer le nombre de sections et de groupes par niveau.
- Ajouter un enseignant.
- Affectation des enseignants à leurs cours.
- Ajouter un étudiant.
- Affectation des étudiants à leur section et leur groupe.
- Ajouter une unité d'enseignement.
- Ajouter un cours.

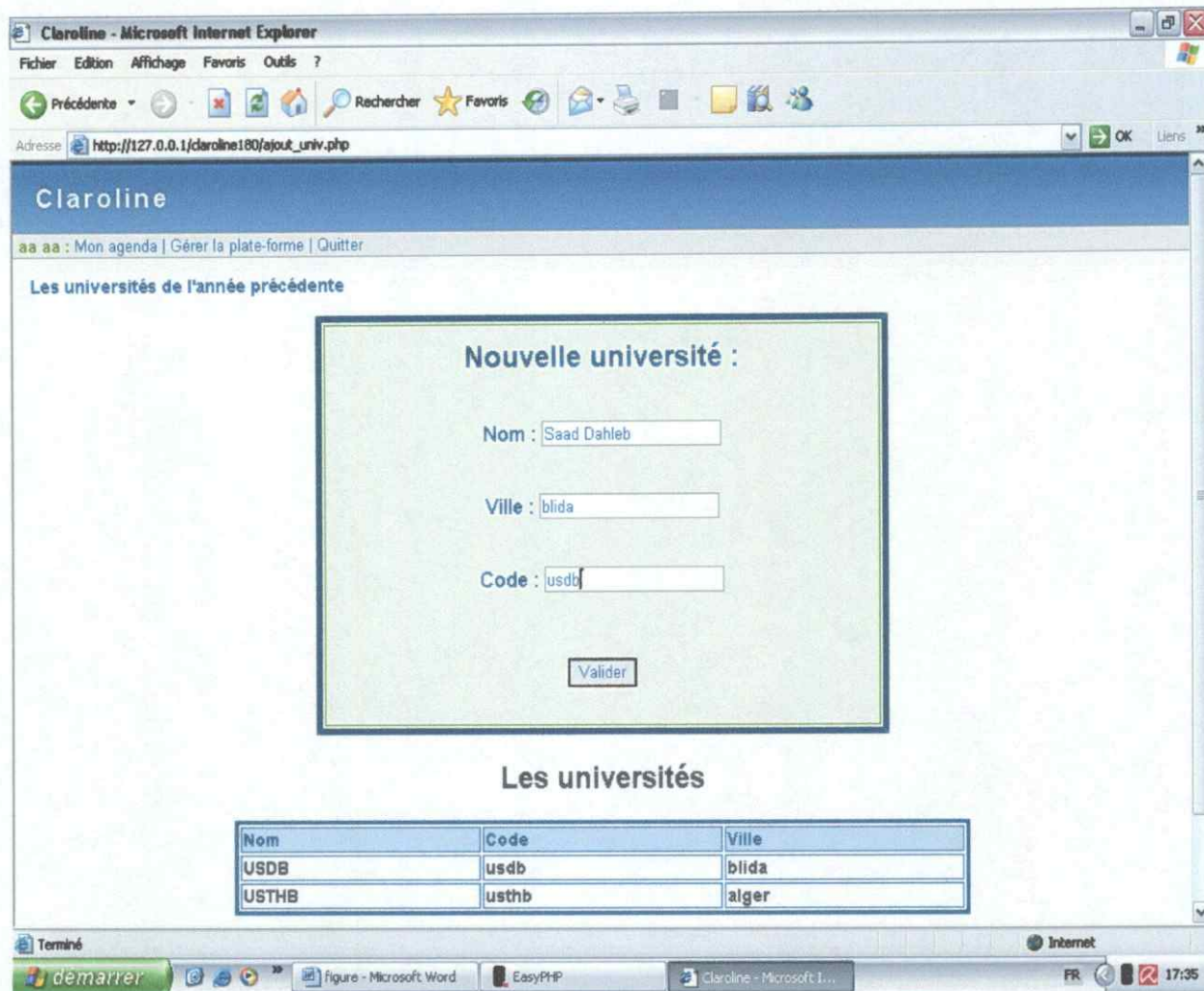


Figure10 : ajouter une université.

Pour ajouter une nouvelle université l'administrateur remplit le formulaire puis valide.

Après le contrôle des données saisies, et dans le cas où elles sont conformes, l'université est ajoutée et apparaît dans la liste des universités.

L'administrateur a la possibilité de récupérer les universités de l'année précédente par le lien « **Les universités de l'année précédente** ».

Même procédure pour ajouter les facultés, les départements, les formations, les unités, les cours, les enseignants et les étudiants.

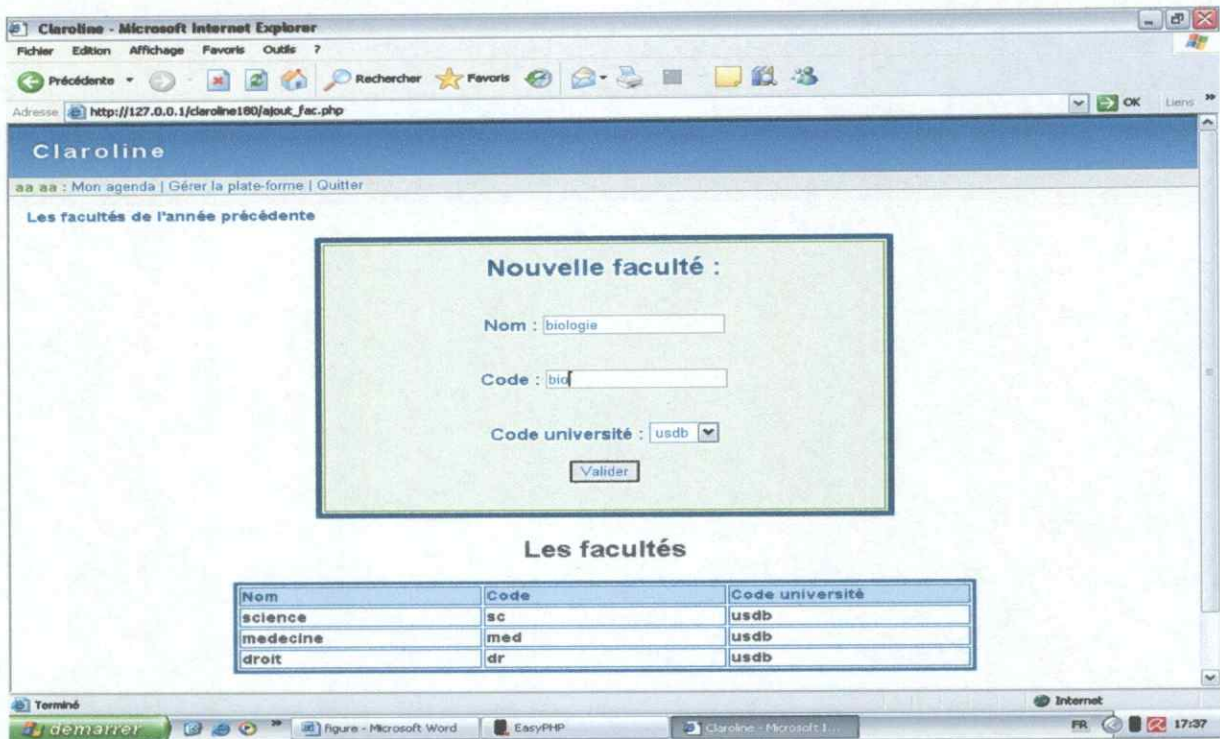


Figure11 : ajouter une faculté.

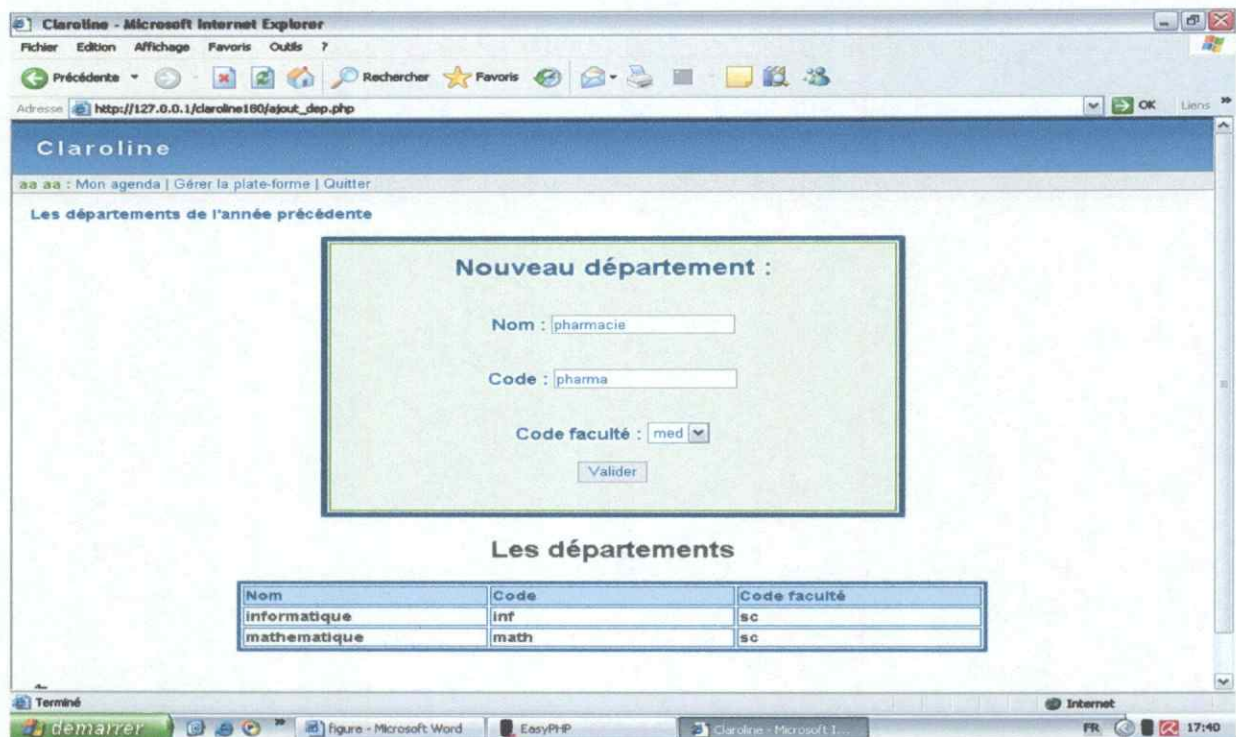


Figure12 : ajouter un département.

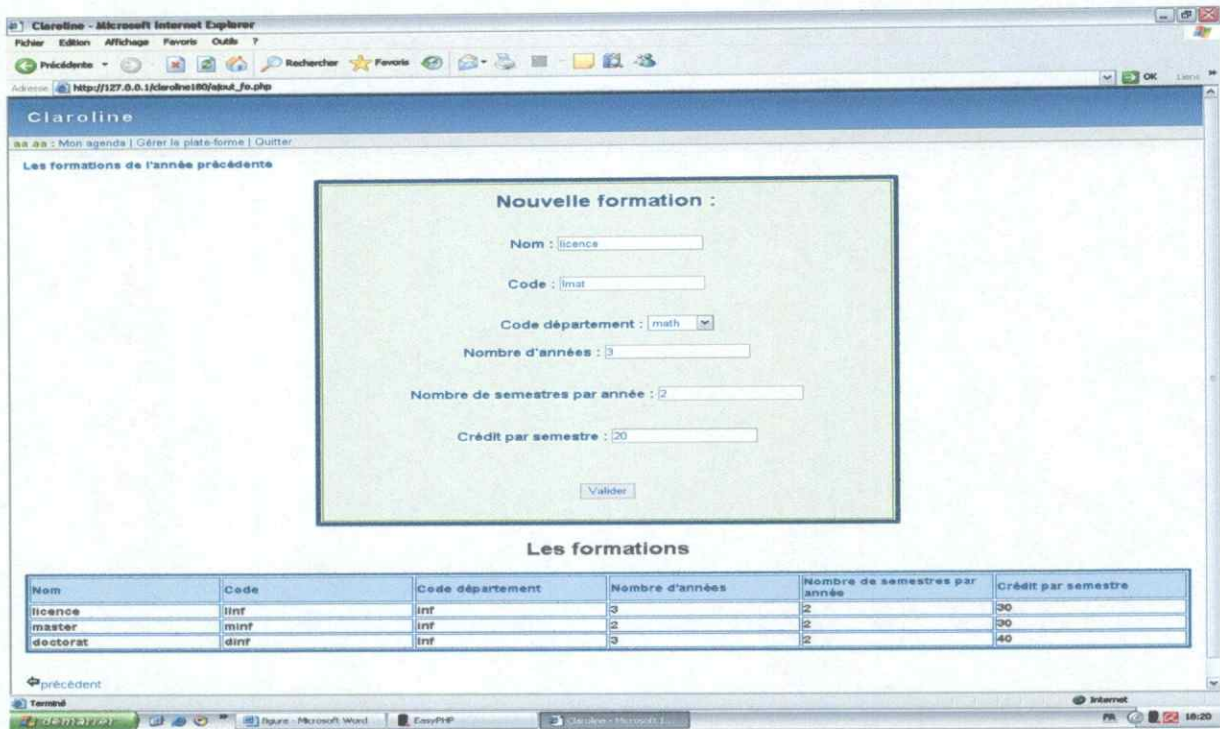


Figure13 : ajouter une formation.

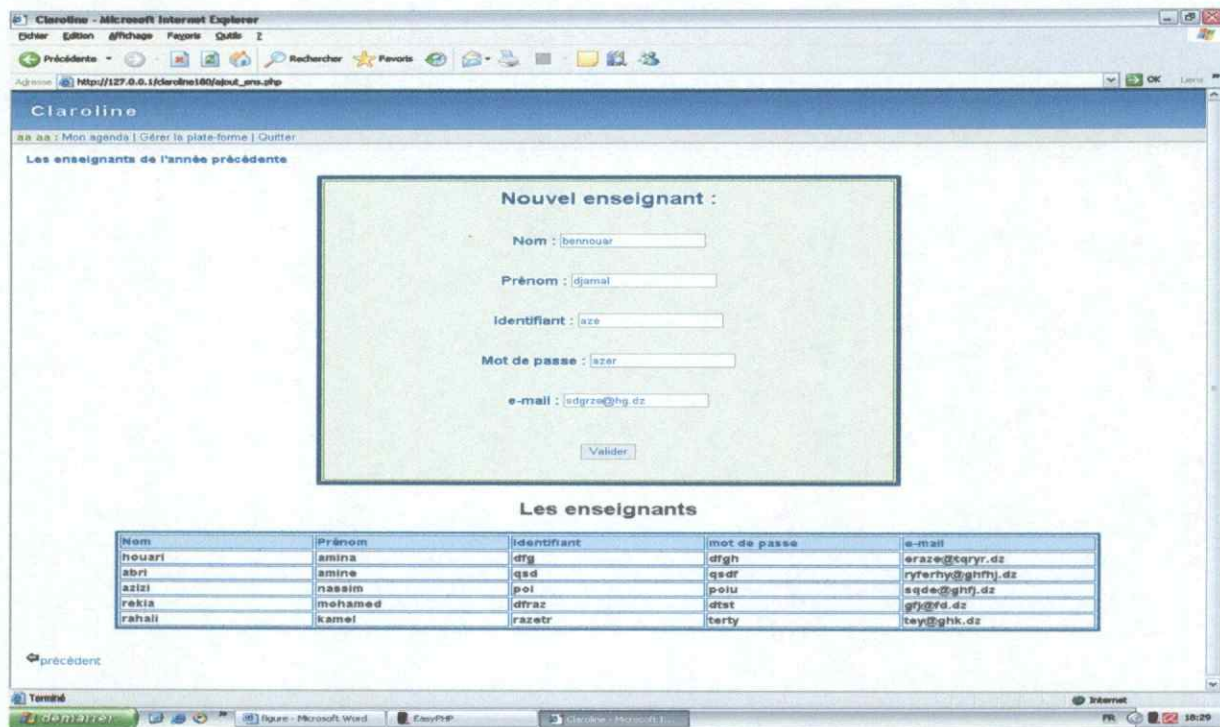


Figure14 : ajouter un enseignant.

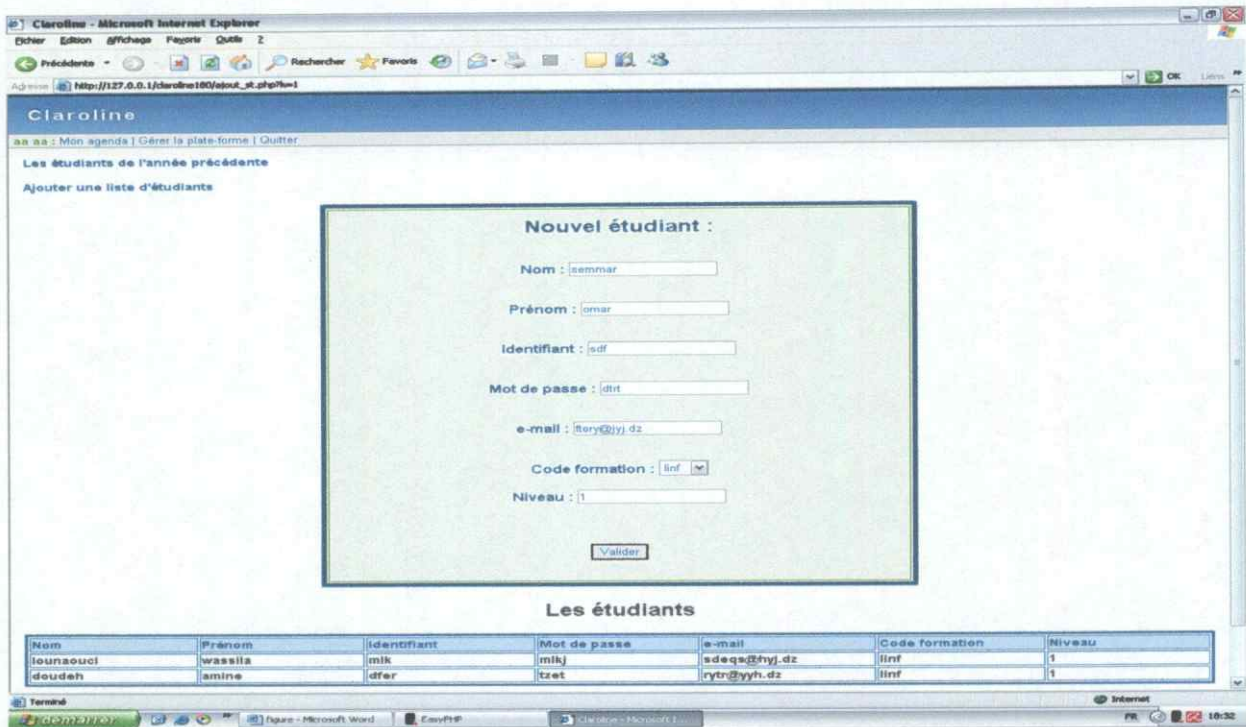


Figure15 : ajouter un étudiant.

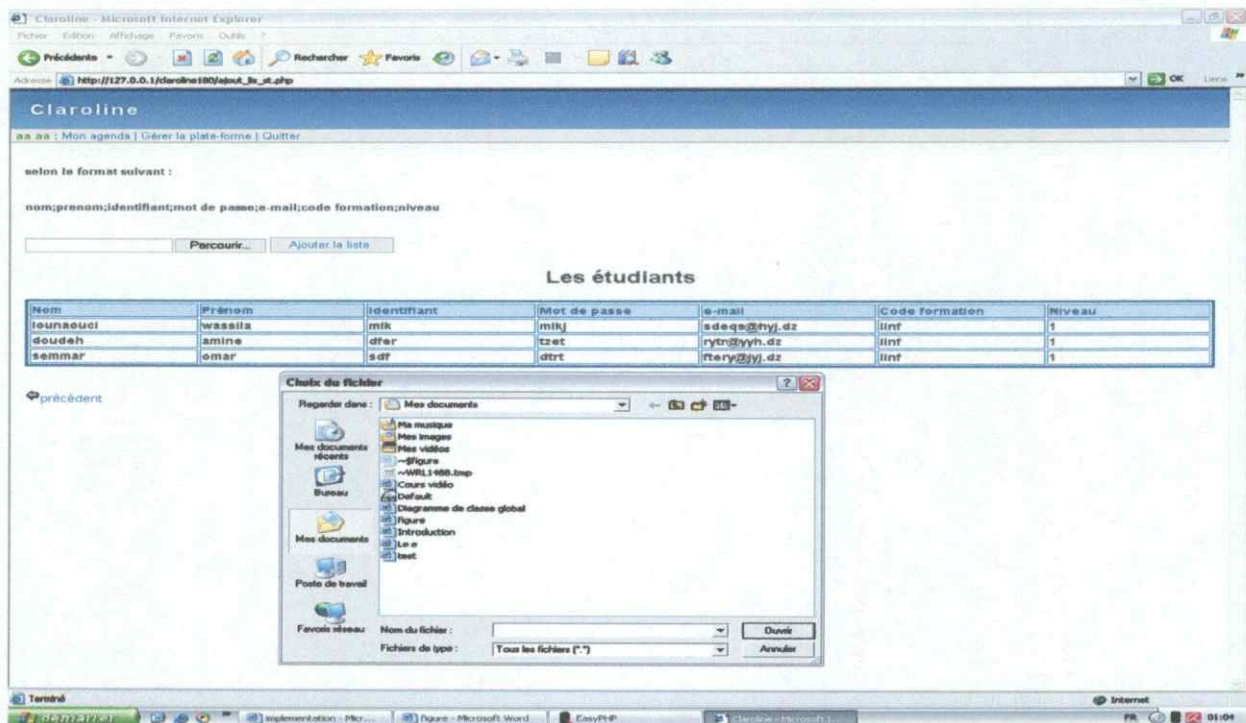


Figure16 : ajouter une liste d'étudiants.

Sélectionner le fichier CSV et valider.

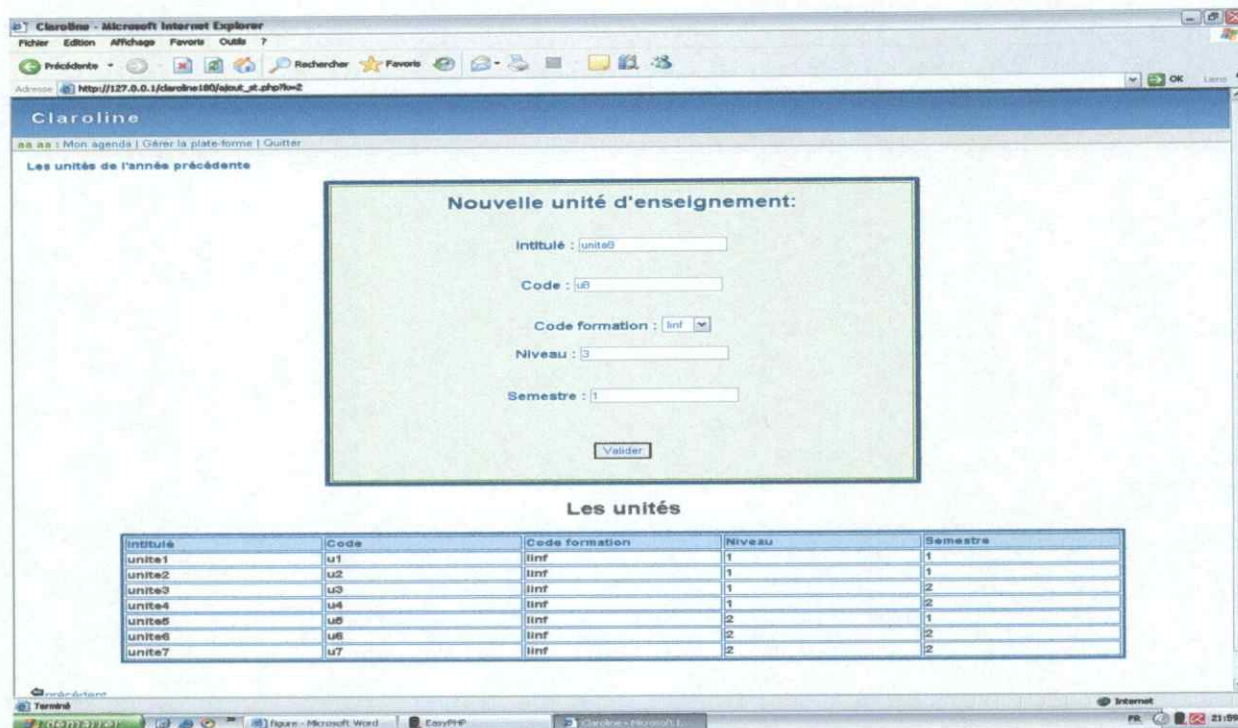


Figure17 : ajouter une unité d'enseignement.

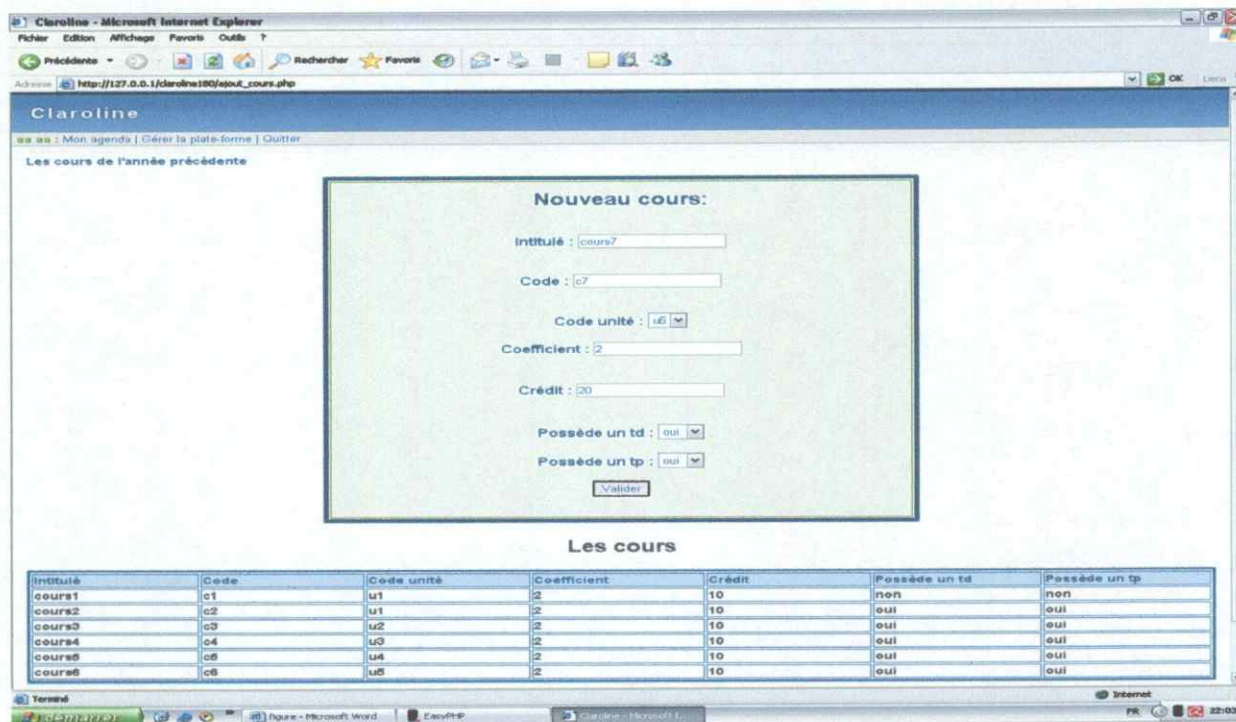


Figure18: ajouter un cours.

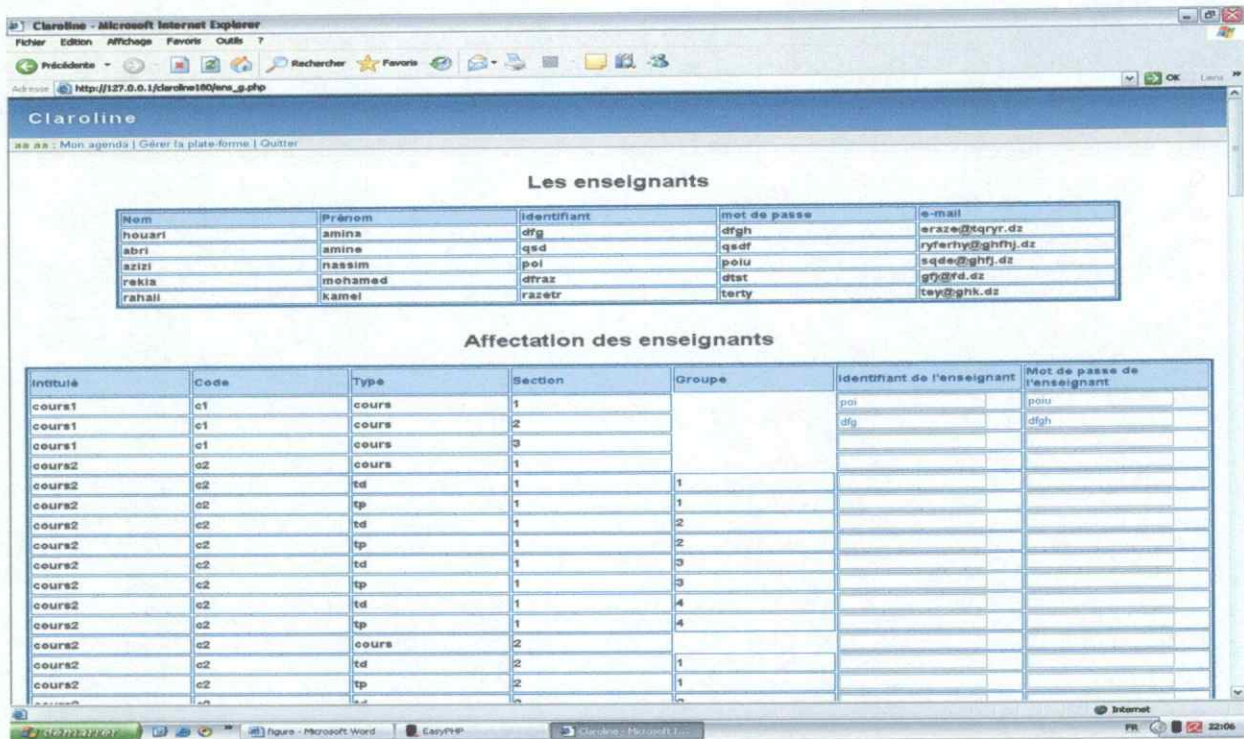


Figure19: affectation des enseignants à leurs cours.

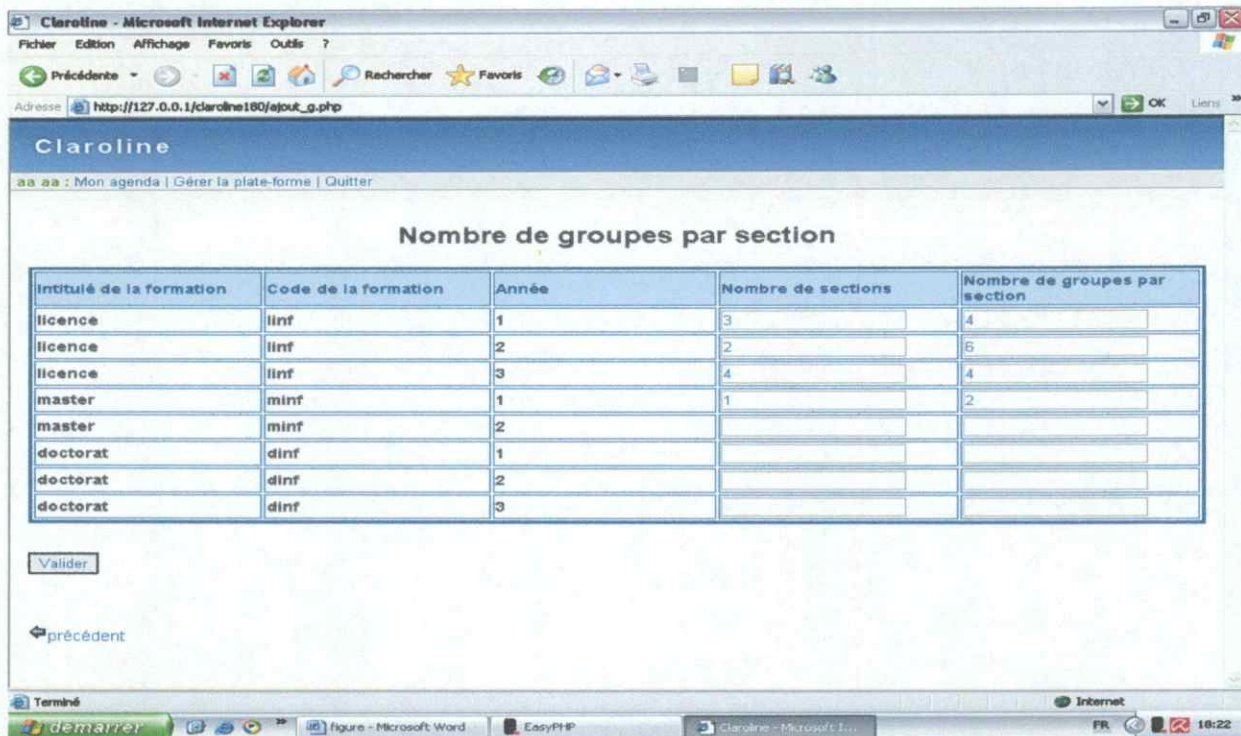


Figure20: détermination du nombre de sections et de groupes par niveau.

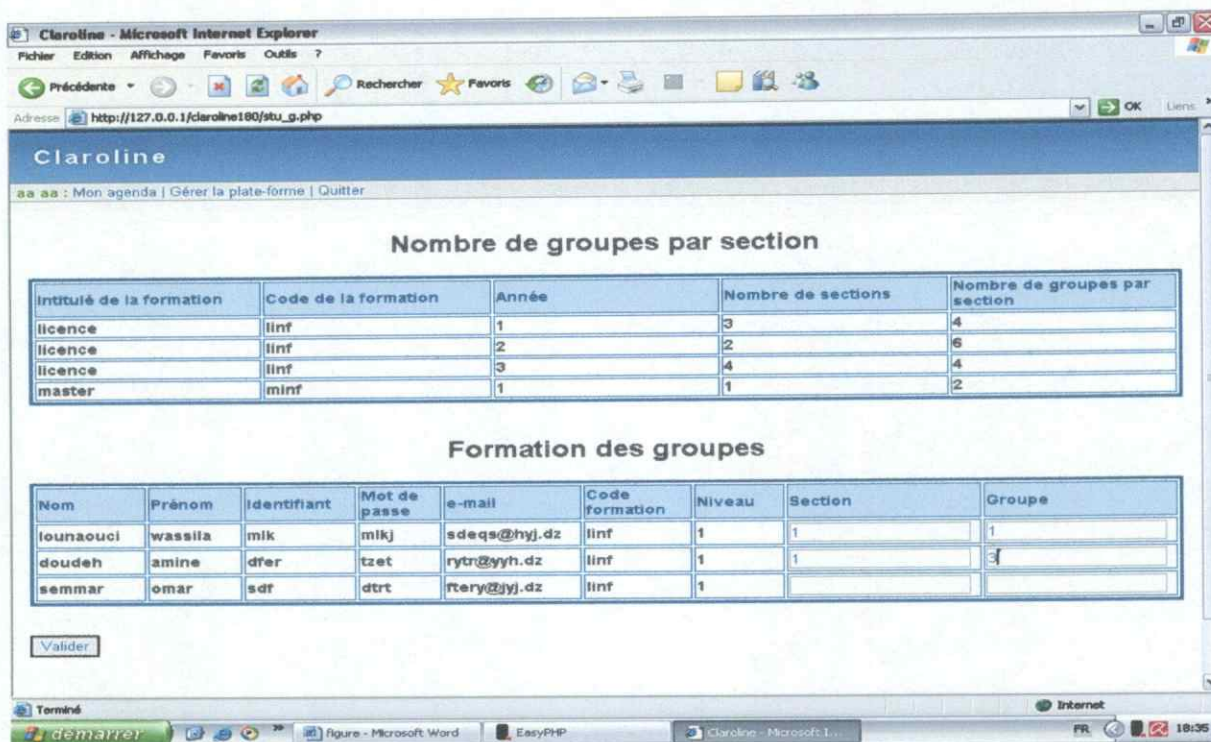


Figure21: affectation des étudiants à leur section et groupe.

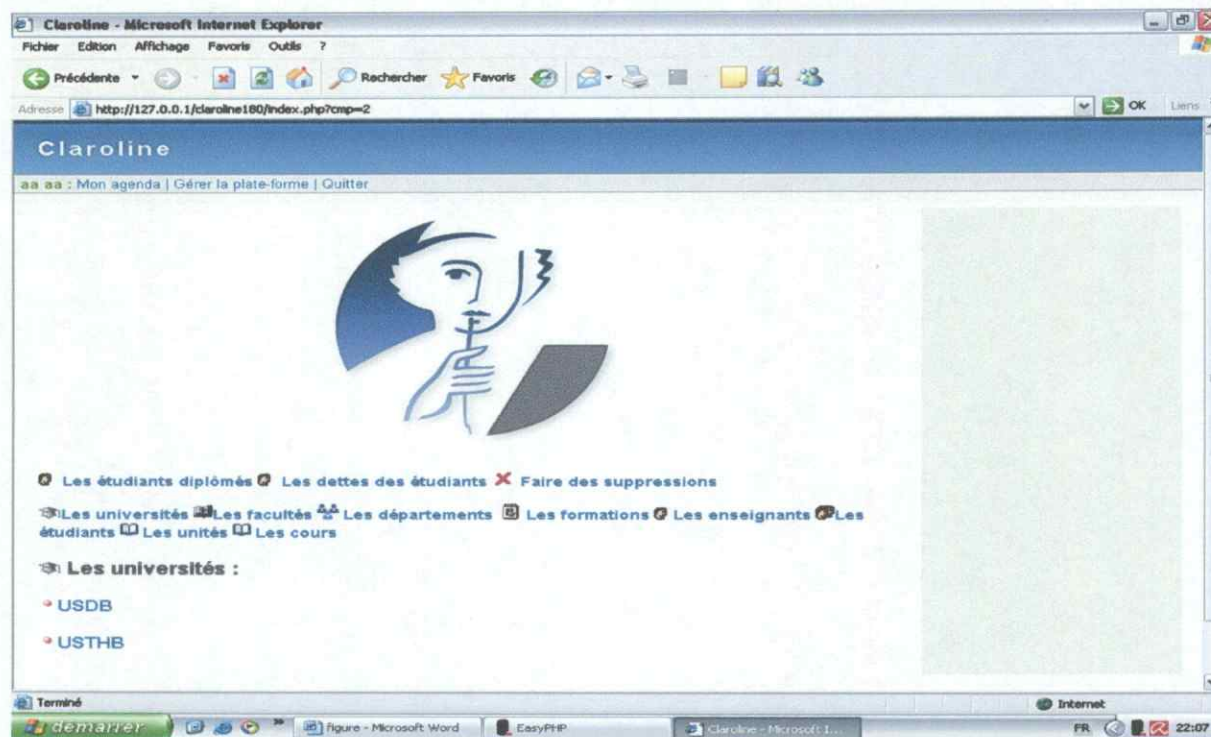


Figure22: les suppressions.

L'administrateur peut supprimer : une université, une faculté, un département, une formation, un enseignant, un étudiant, une unité d'enseignement, un cours.

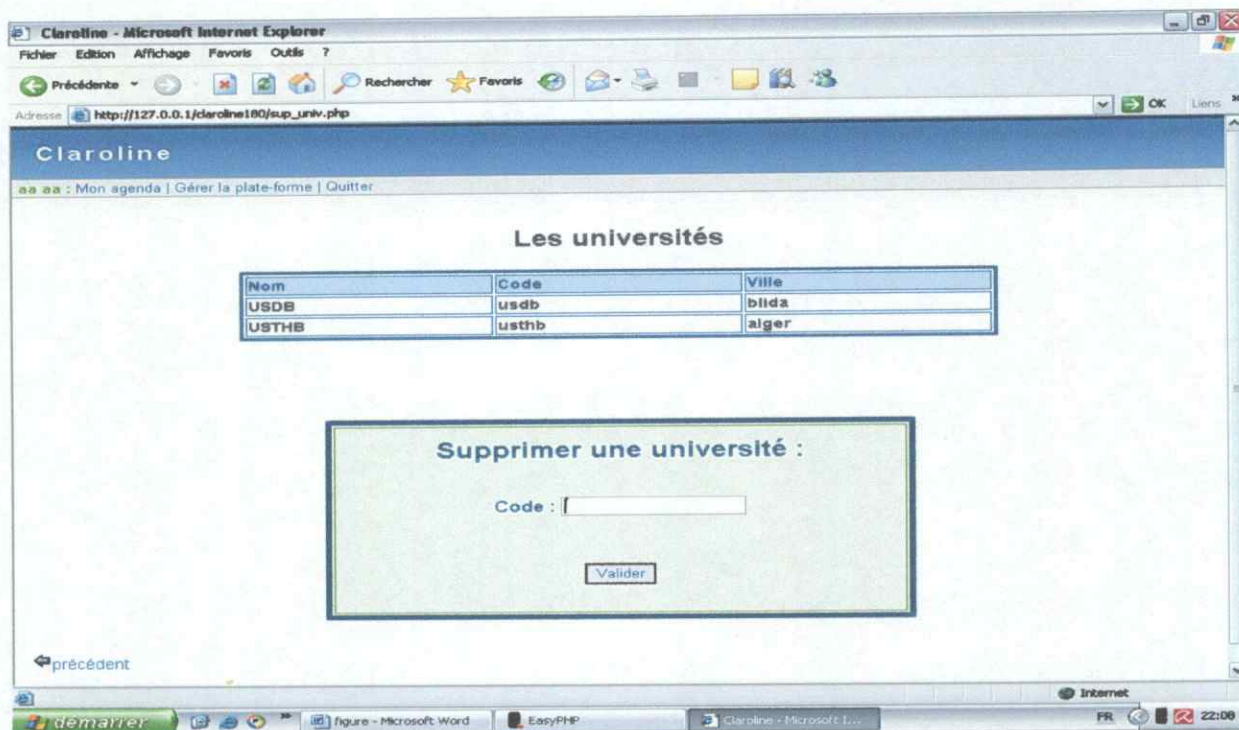


Figure23:supprimer une université.

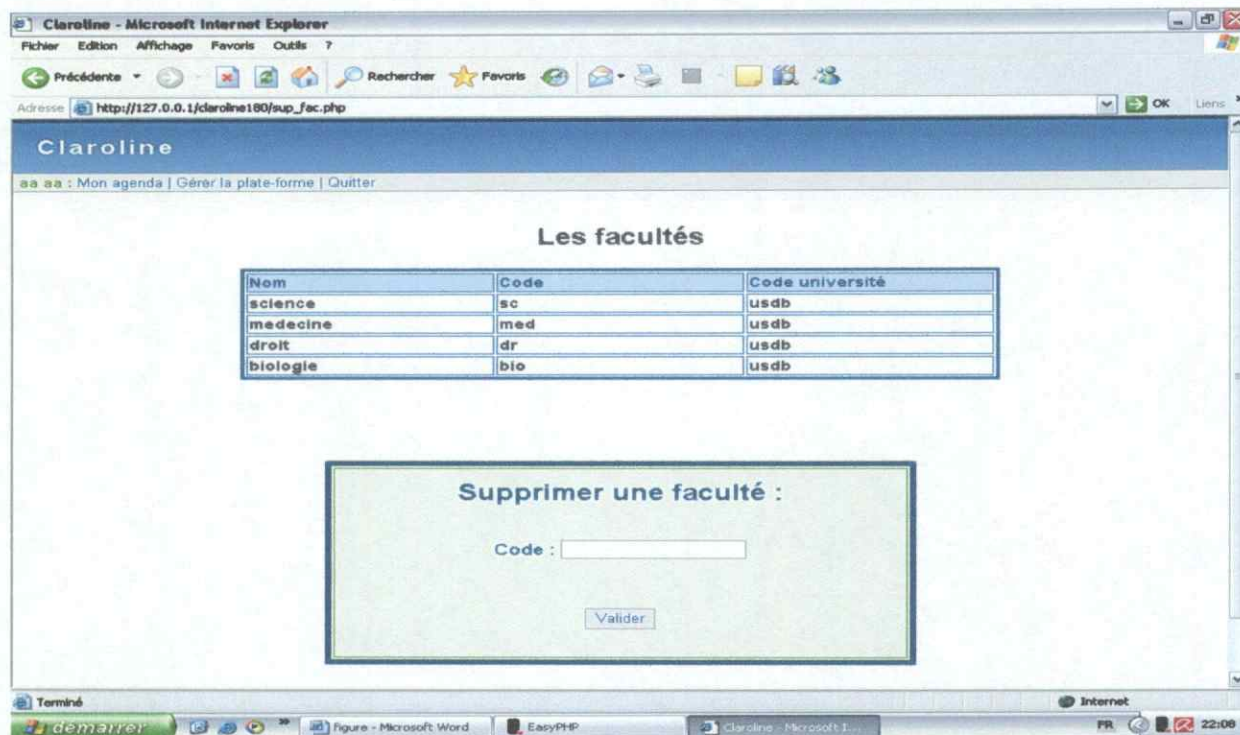


Figure24:supprimer une faculté.

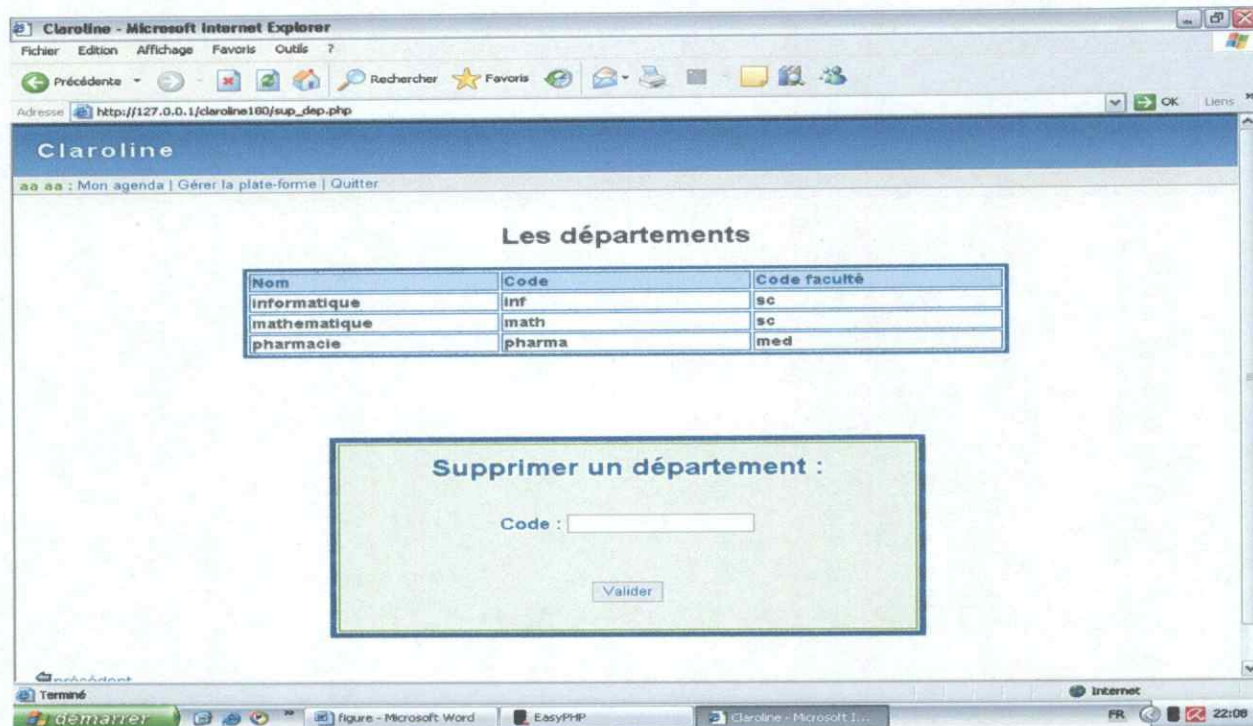


Figure25:supprimer un département.

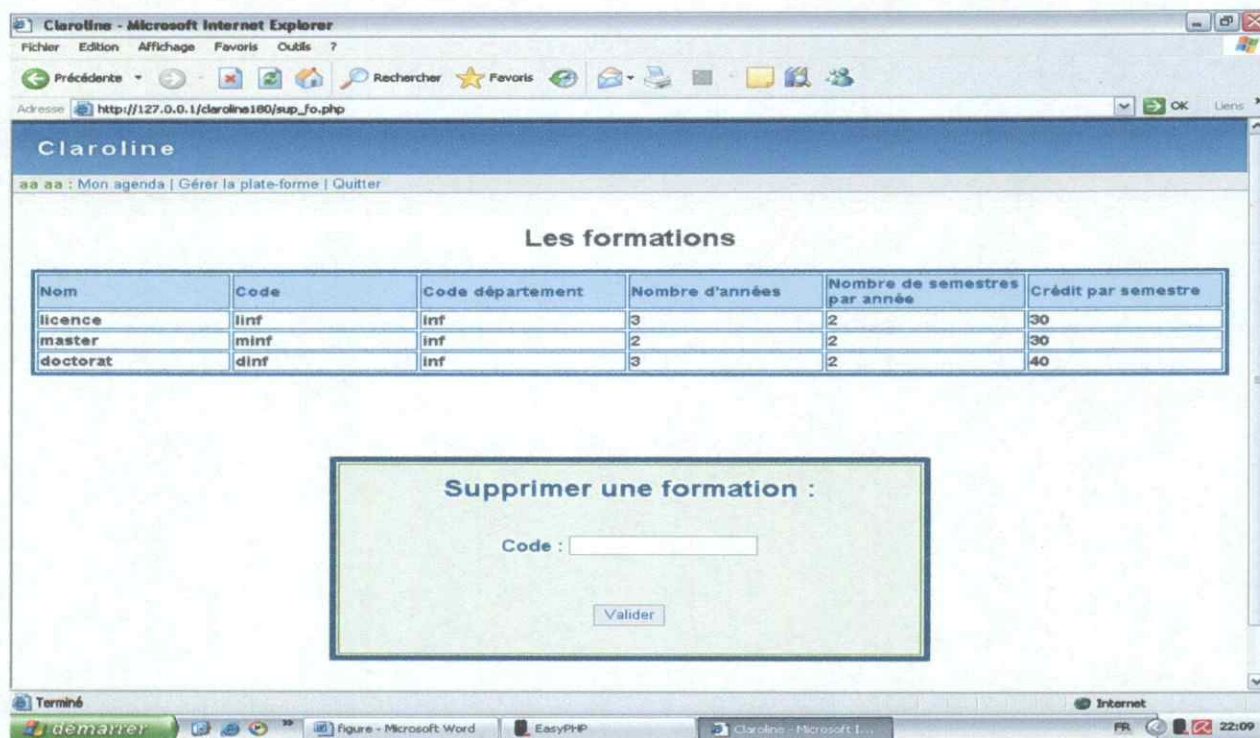


Figure26:supprimer une formation.



Figure27:supprimer un enseignant.

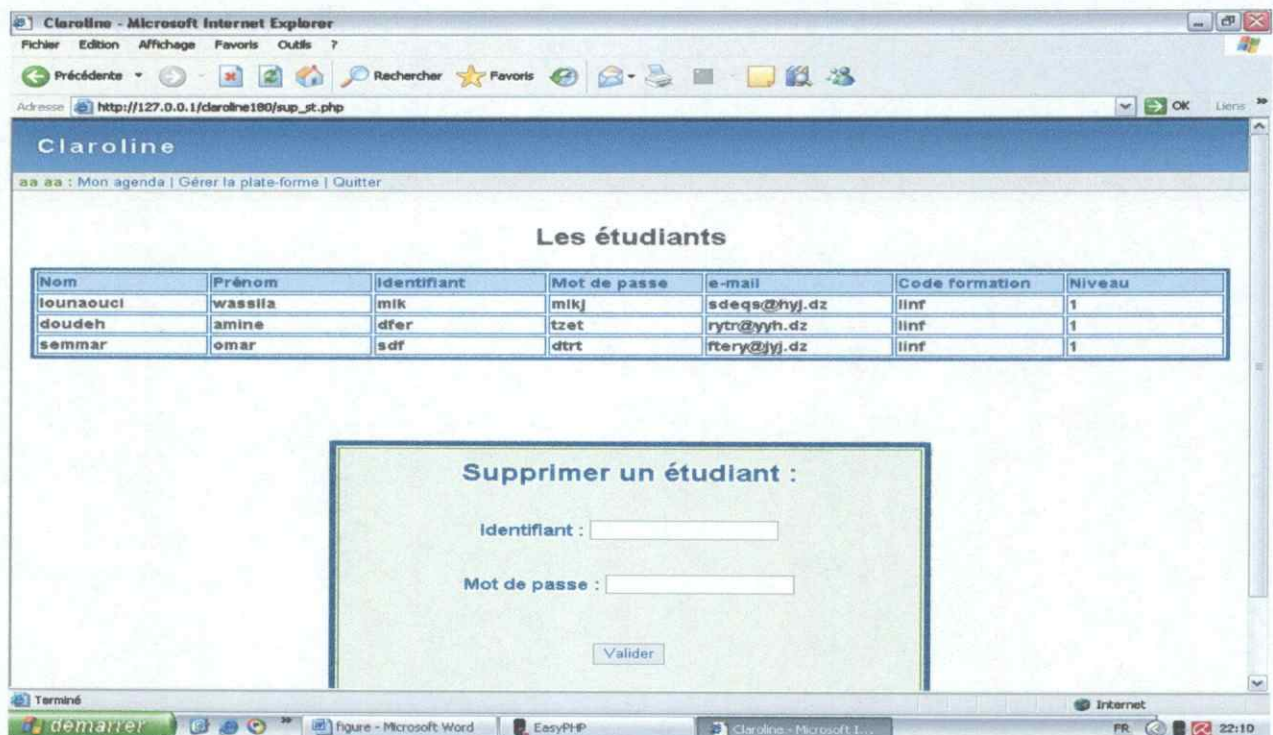


Figure28:supprimer un étudiant.

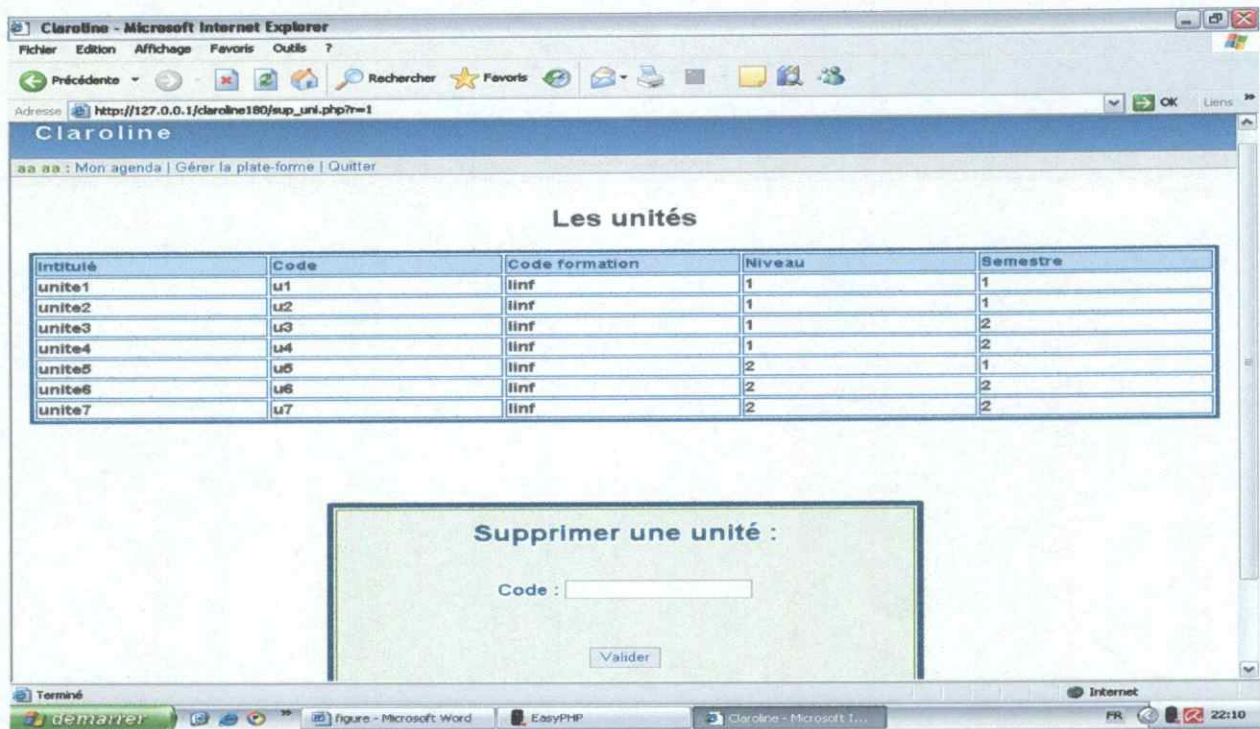


Figure29:supprimer une unité.

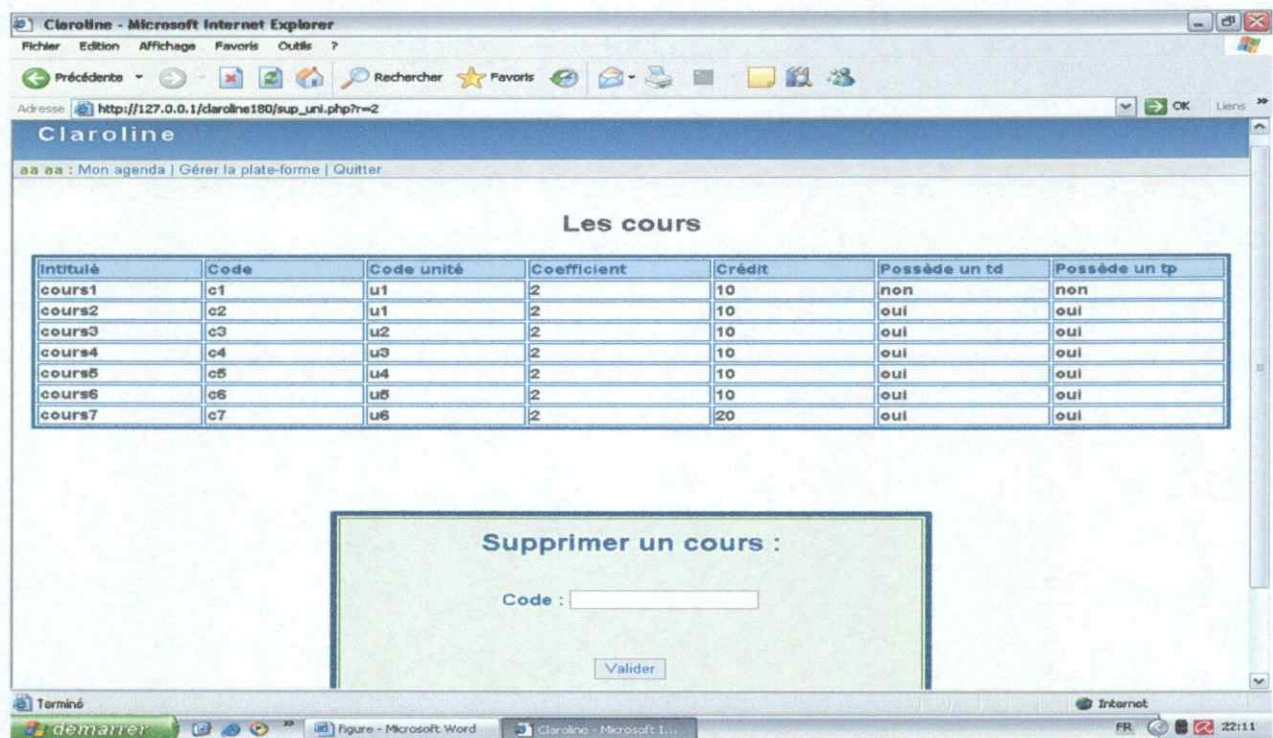


Figure30:supprimer un cours.

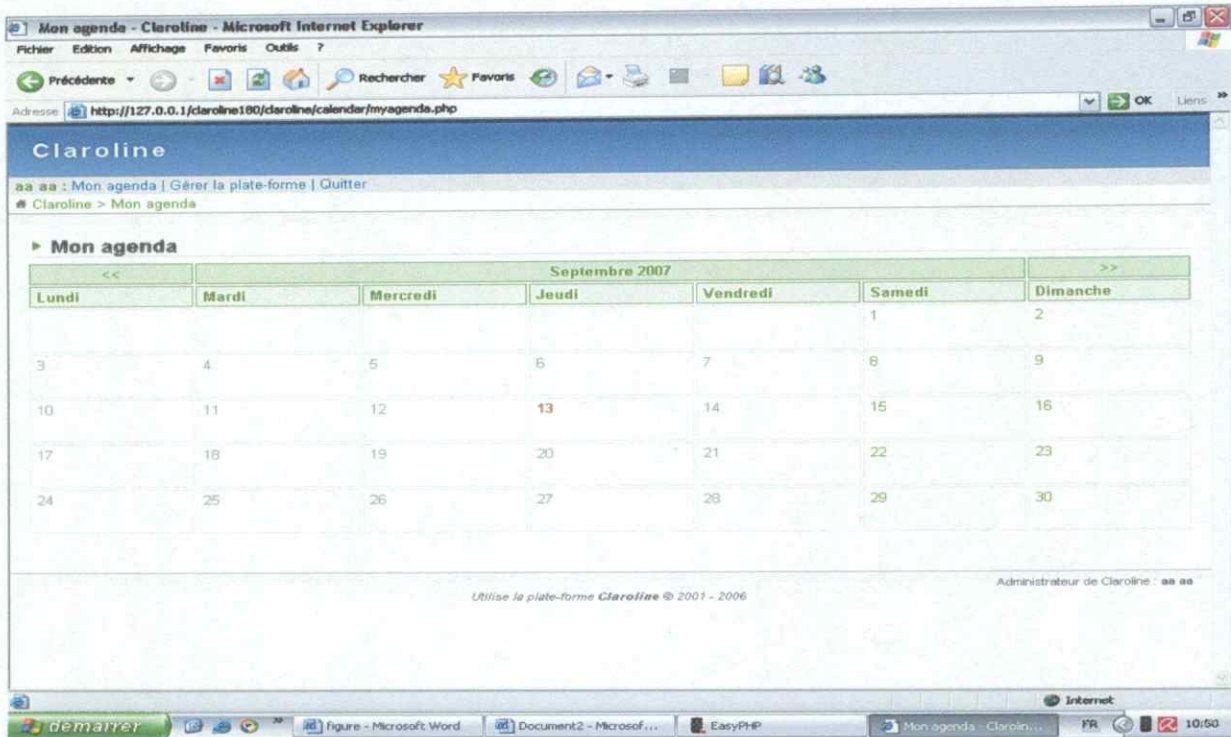


Figure31: consulter l'agenda.

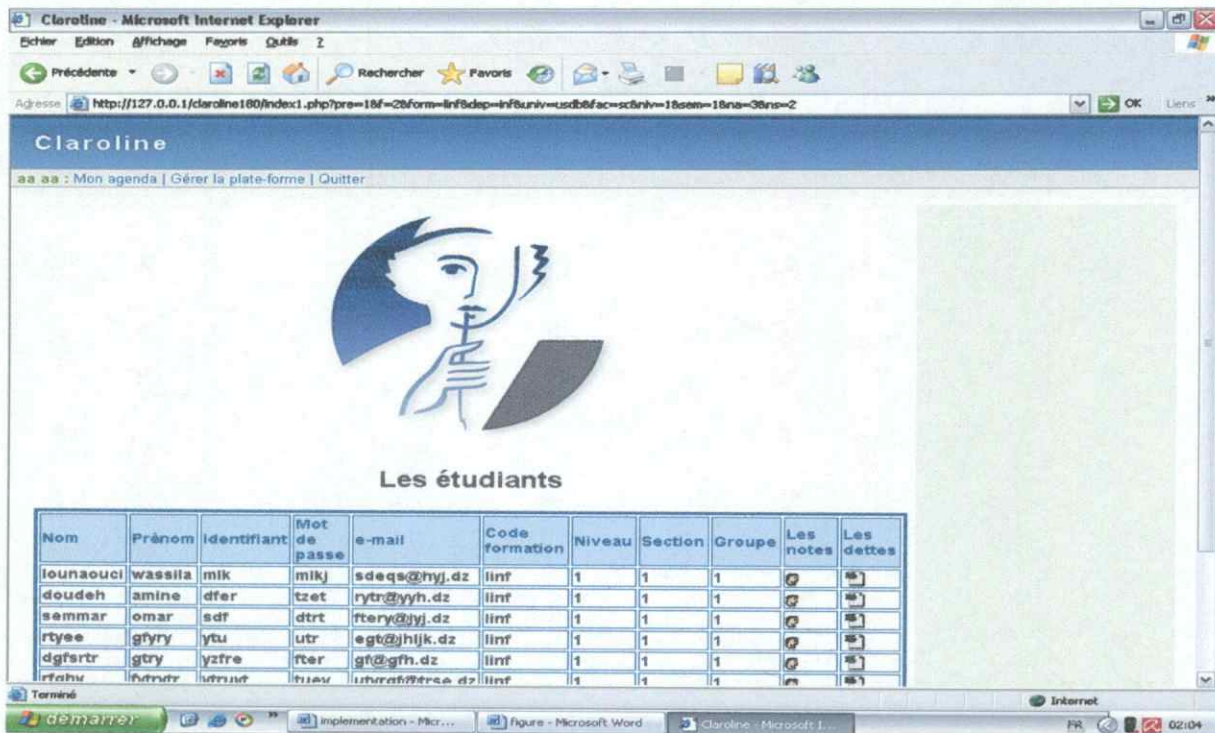


Figure32: consulter la liste des étudiants d'un niveau.

Il y a un lien qui mène vers les notes de l'étudiant et un autre lien pour les dettes.

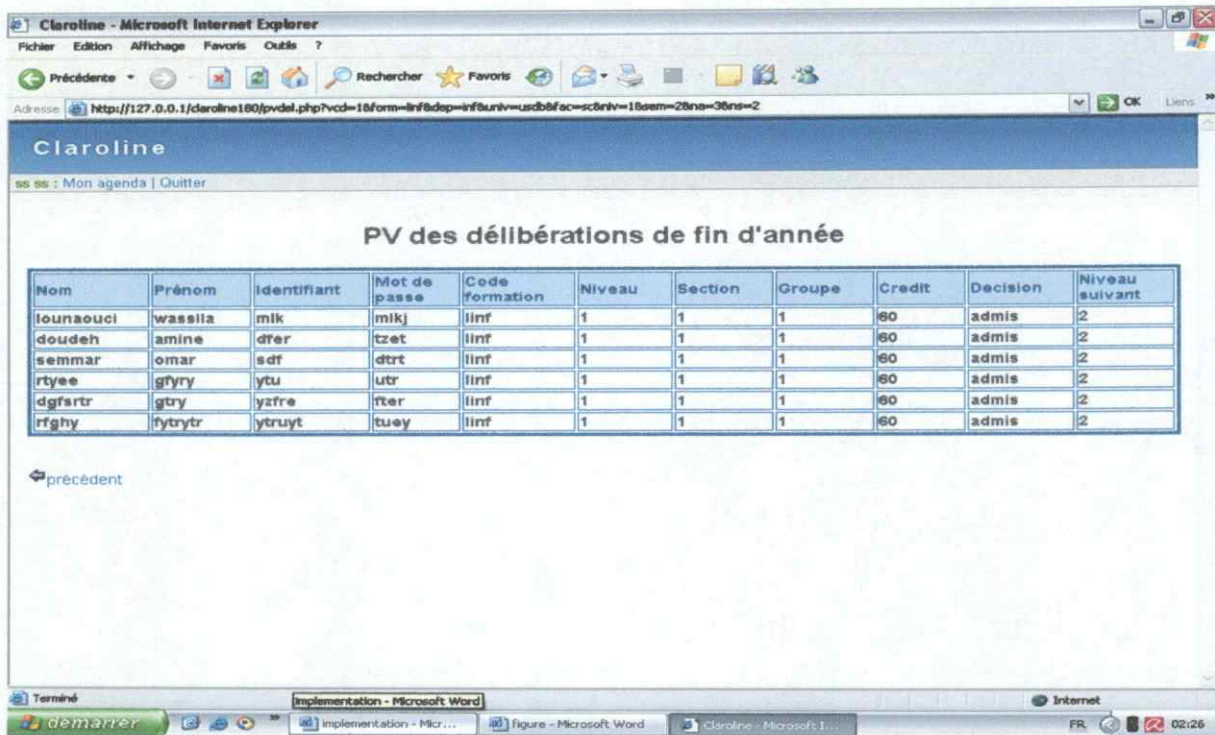


Figure33:consulter le PV des délibérations.

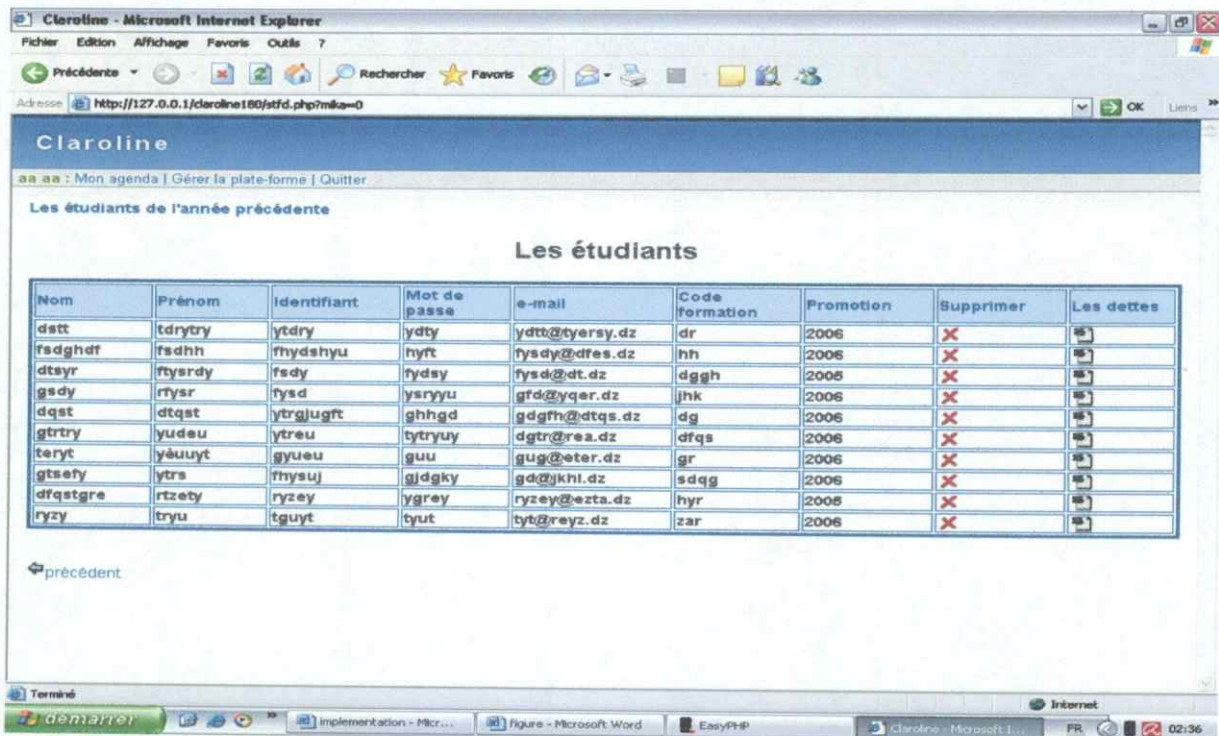


Figure34:les étudiants diplômés et qui ont des dettes.

Il y a un lien qui mène vers les dettes.

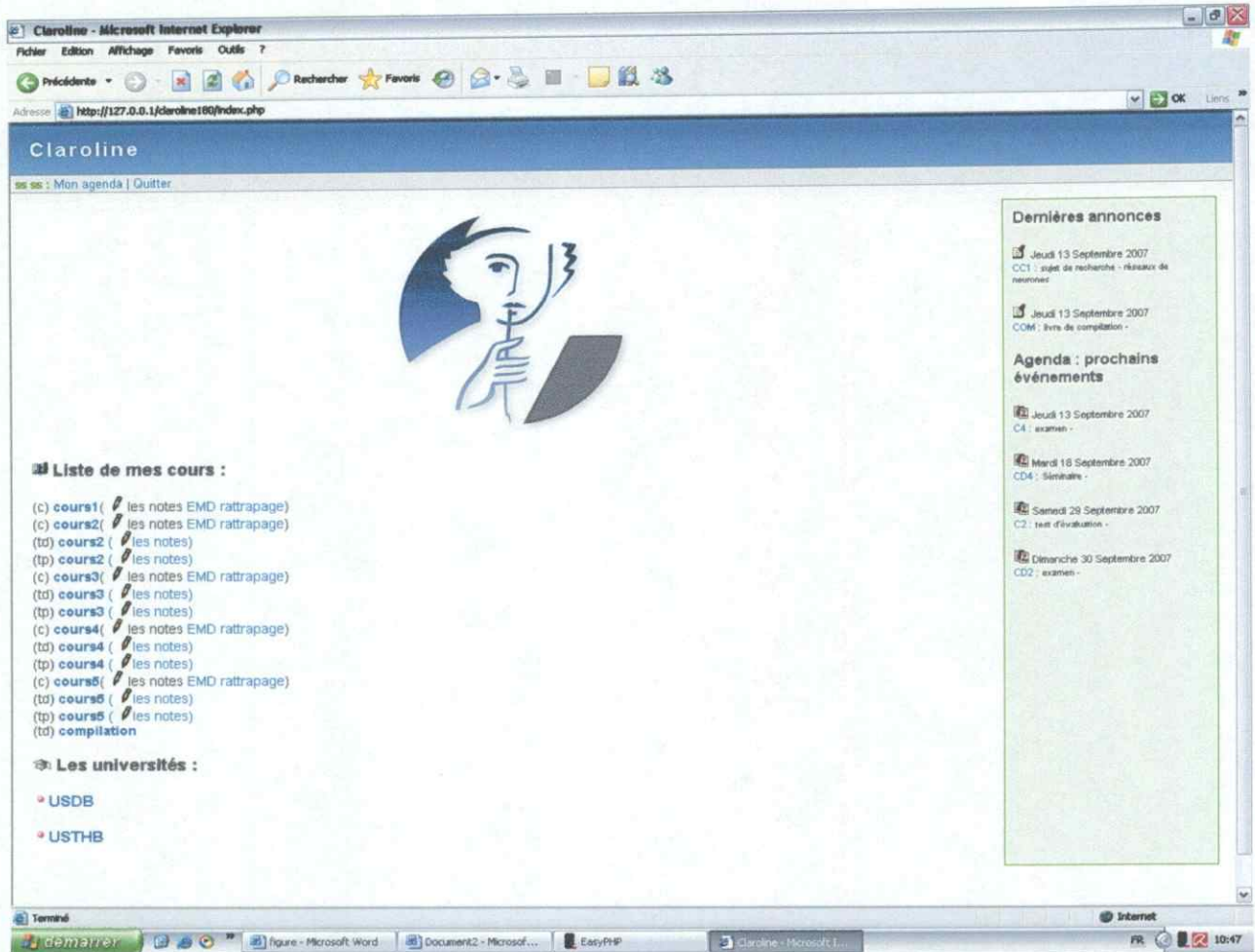


Figure35:page d'accueil de l'enseignant.

En introduisant l'identifiant et le mot de passe, l'enseignant accède a son espace privé.

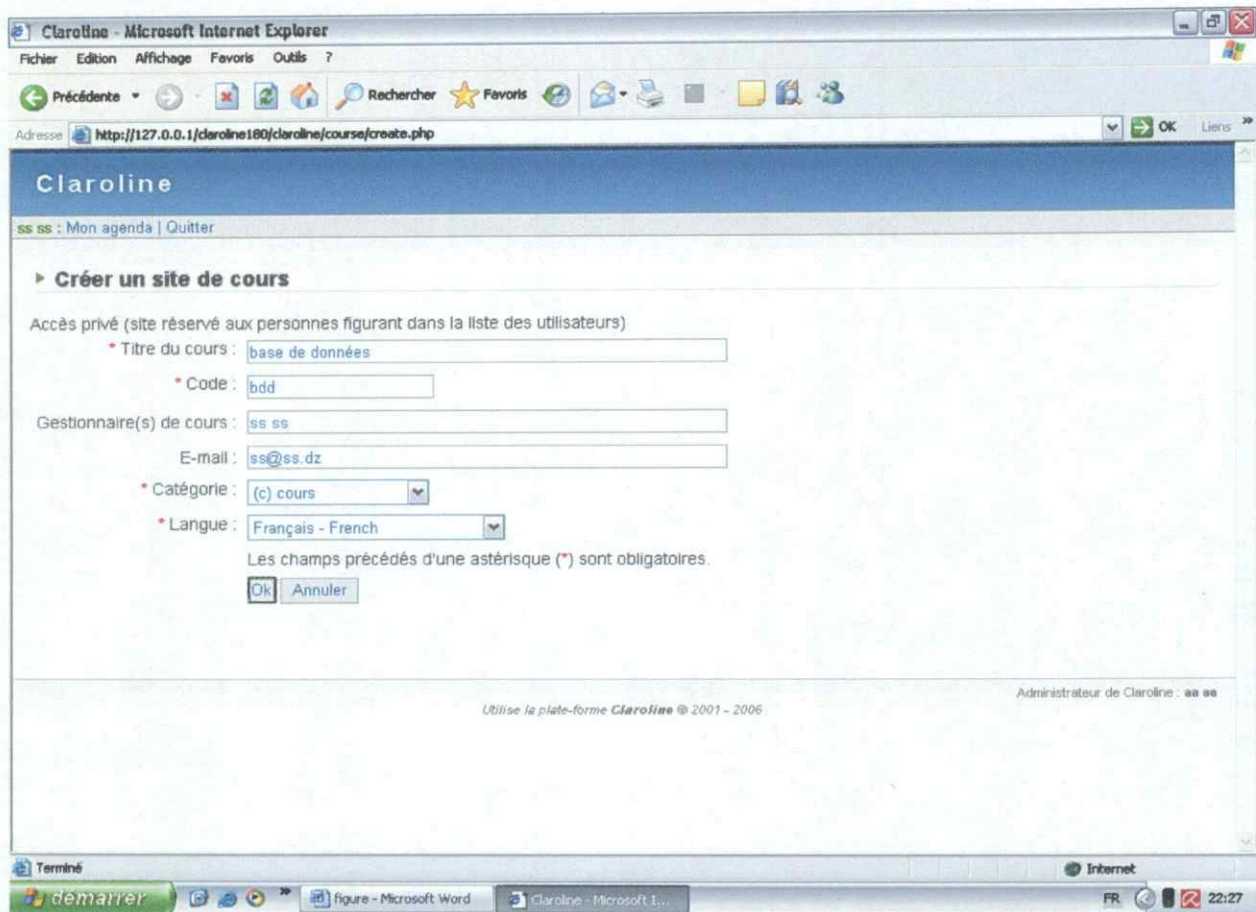


Figure36:création d'un cours.

Pour créer un cours l'enseignant remplit le formulaire puis valide.

Après le contrôle des données saisies, et dans le cas où elles sont conformes, le cours est créé.

Remarque:

L'enseignant crée les cours qui lui sont affectés par l'administrateur mais il peut aussi créer d'autre cours et pour l'évaluation il ne pourra éditer les notes que sur les cours qui lui sont affectés.

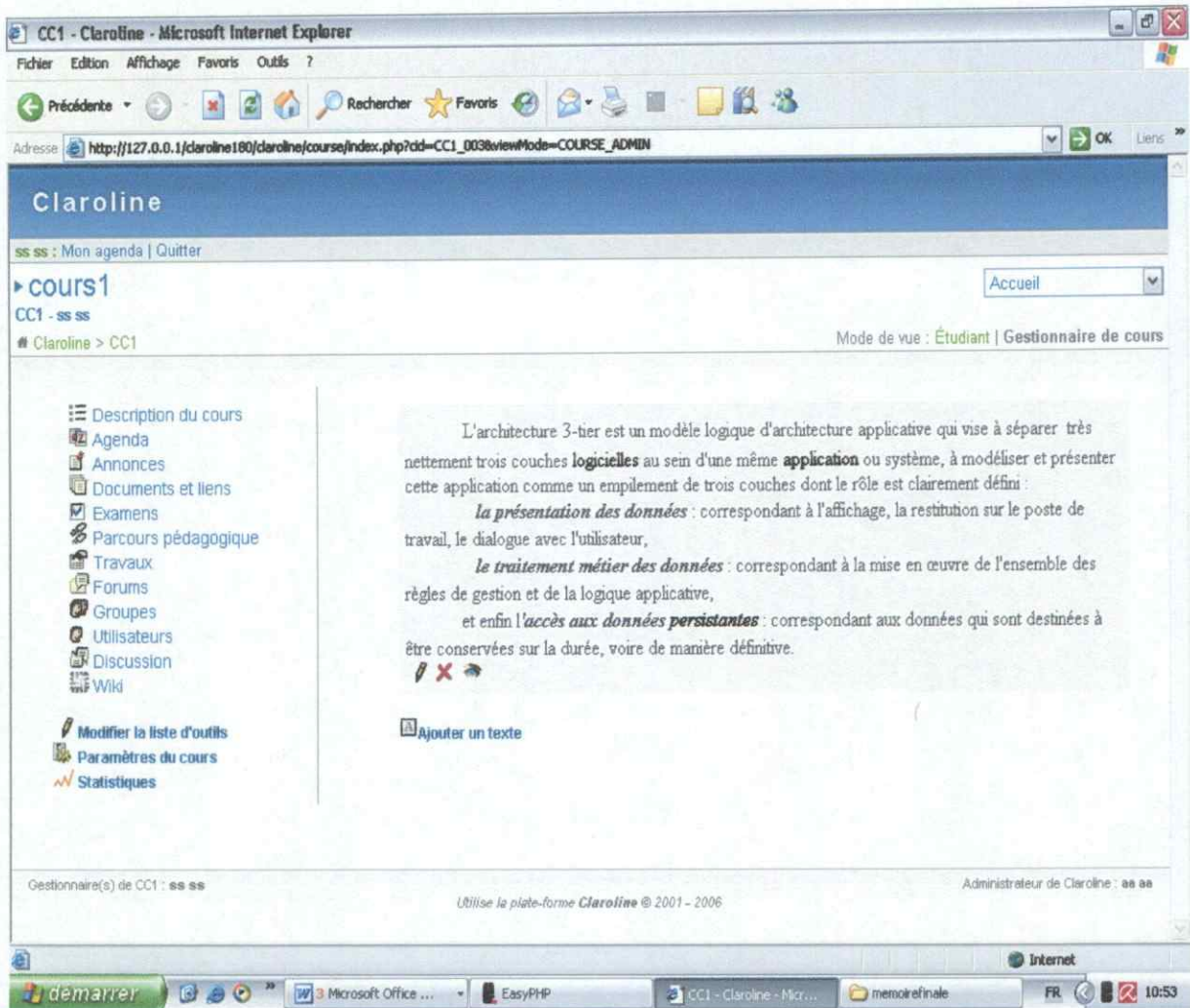


Figure37:page d'accueil du cours pour l'enseignant.

L'enseignant peut :

- Ajouter un texte.
- Faire la description du cours.
- Ajouter des documents et des liens.
- Préparer les examens.
- Préparer les travaux.

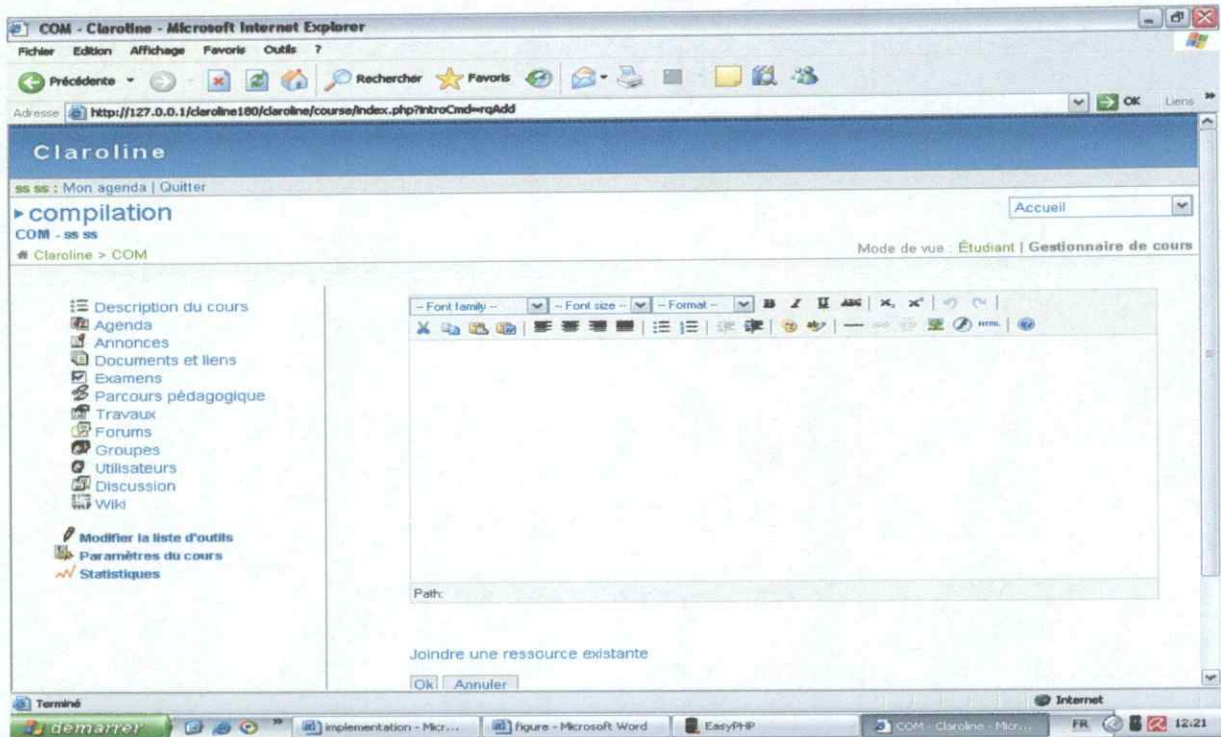


Figure38:ajouter un texte à un cours.

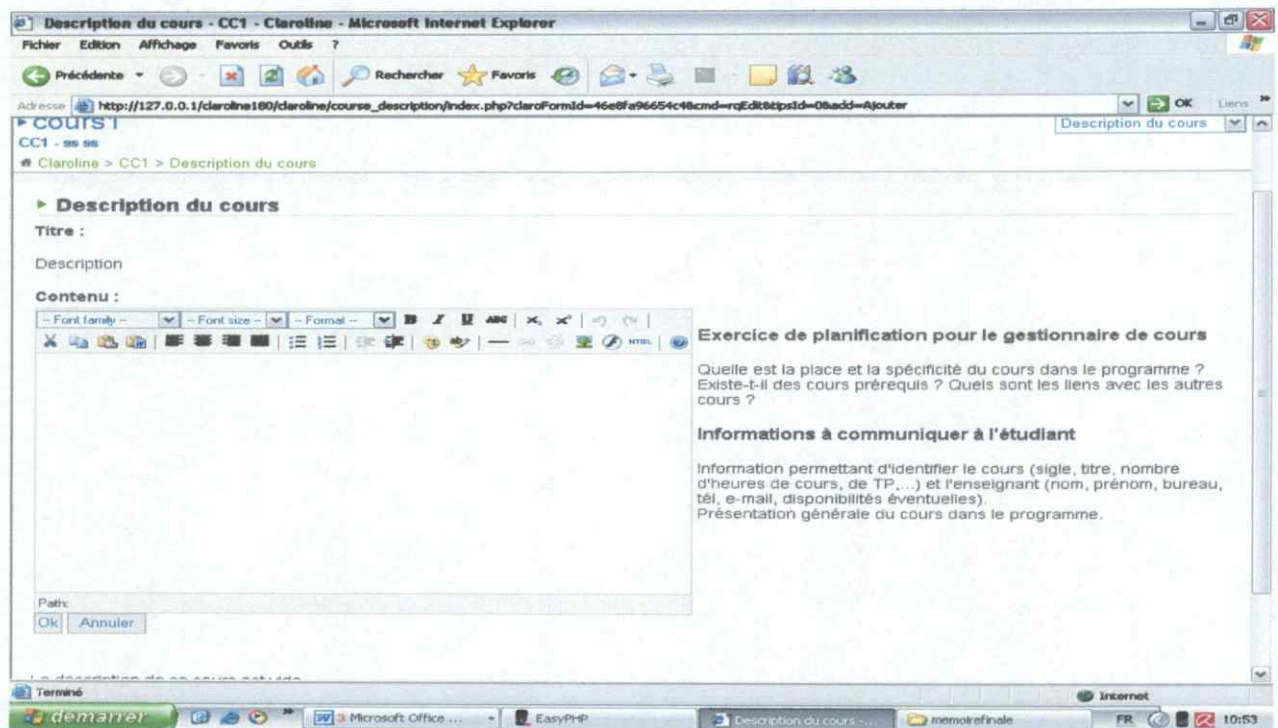


Figure39:faire la description d'un cours.

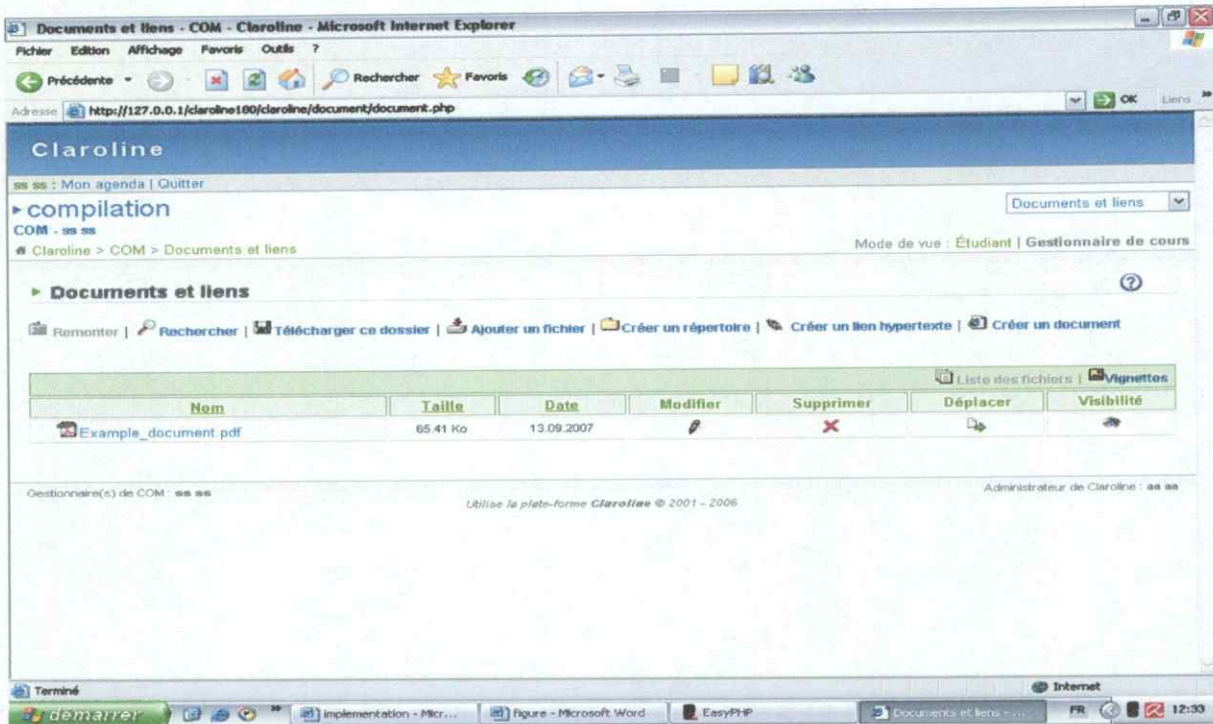


Figure40:ajouter un document ou un lien.

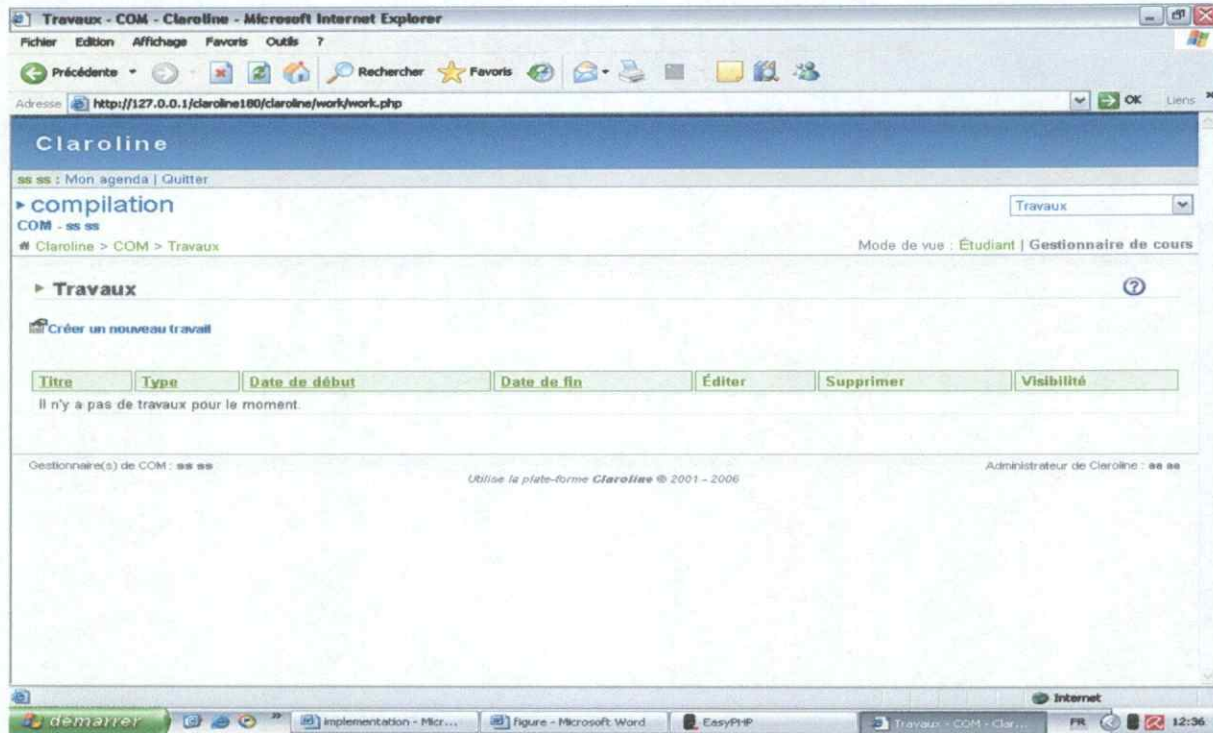


Figure41:préparer un travail.

The screenshot shows a web browser window titled "Nouvel examen - COM - Claroline - Microsoft Internet Explorer". The address bar shows the URL: `http://127.0.0.1/claroline180/claroline/exercice/admin/edit_exercice.php?cmd=rdEdit`. The page content is as follows:

- Navigation:** "ss ss : Mon agenda | Quitter" and a dropdown menu for "Examens".
- Breadcrumbs:** "# Claroline > COM > Examens > Nouvel examen".
- Section:** "► Nouvel examen".
- Form Fields:**
 - "Titre *": A text input field.
 - "Description": A rich text editor with a toolbar and a large text area.
 - "Type d'examen":
 - Exercices sur une seule page
 - Un exercice par page (séquentiel)
 - "Avancé (Facultatif)":
 - "Date de début": 14 / Septembre / 2007 - 12 / 39 (j/m/a hh:mm)
 - "Date de fin": Oui, 14 / Septembre / 2007 - 12 / 39 (j/m/a hh:mm)
 - "Temps maximum autorisé": Oui, 0 min. 0 sec.
 - "essais autorisés": Essais illimités
 - "Essais anonymes":
 - Autoriser (les noms des utilisateurs ne sont pas enregistrés dans les statistiques et les utilisateurs anonymes peuvent faire l'examen)
 - Interdire (les noms des utilisateurs sont enregistrés dans les statistiques et les utilisateurs anonymes ne peuvent pas faire l'examen)

The taskbar at the bottom shows the "démarrer" button, several open applications, and the system clock showing 12:40.

Figure42:préparer un examen.

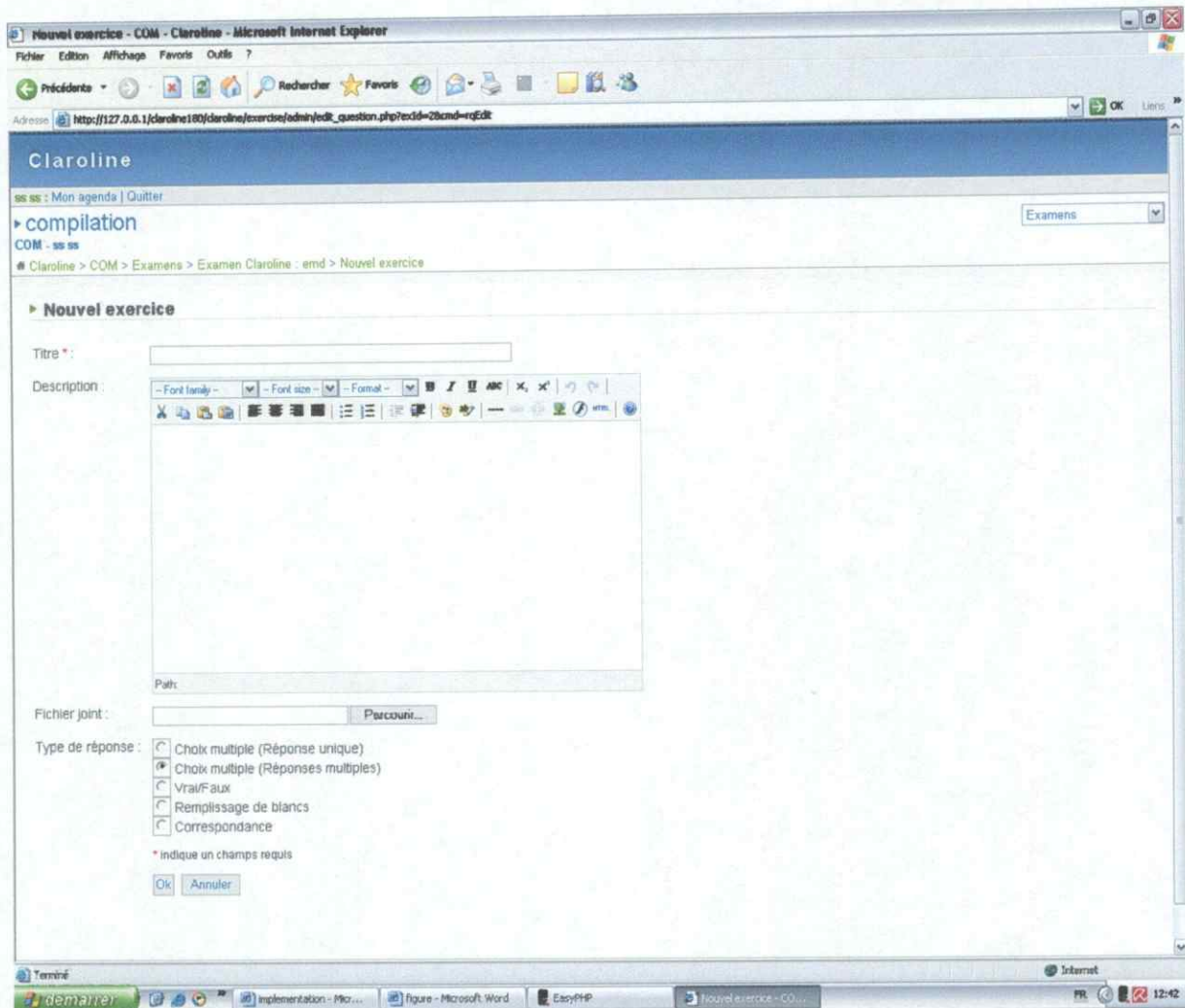


Figure43:ajouter un exercice à un examen.

L'enseignant choisi un type d'exercice :

- Choix multiple (Réponse unique).
- Choix multiple (Réponses multiples).
- Vrai/Faux.
- Remplissage de blancs.
- Correspondance

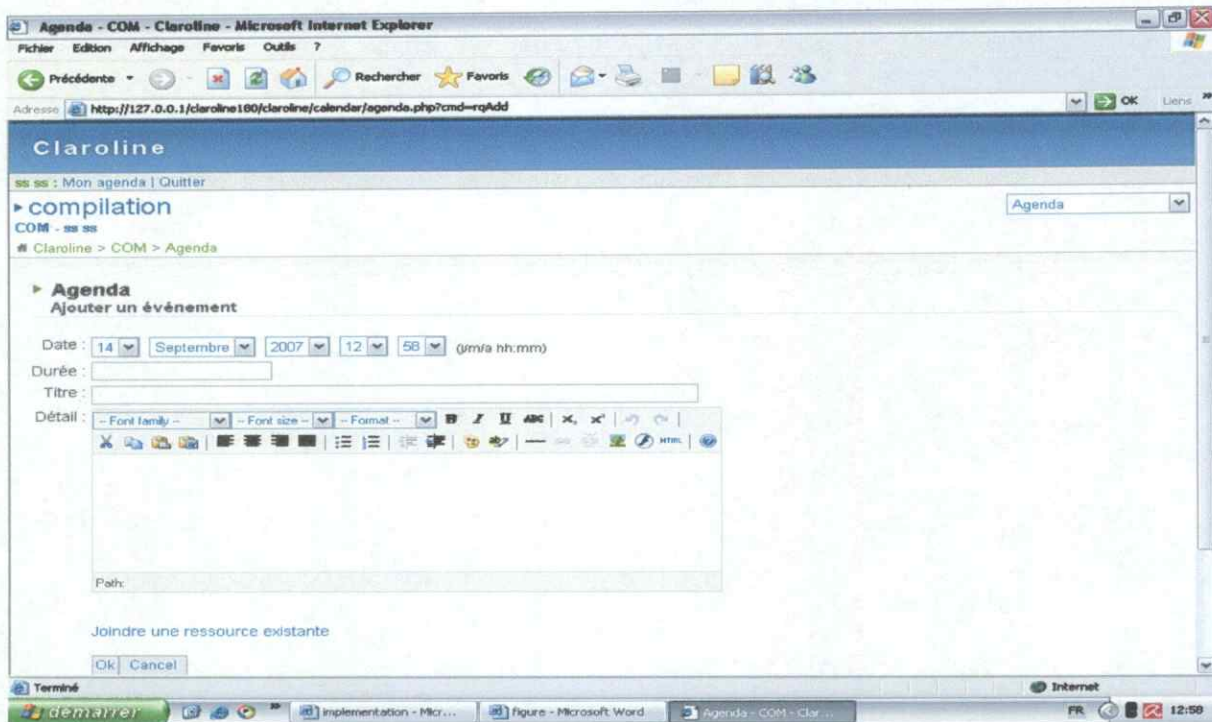


Figure44:ajouter un événement à l'agenda.

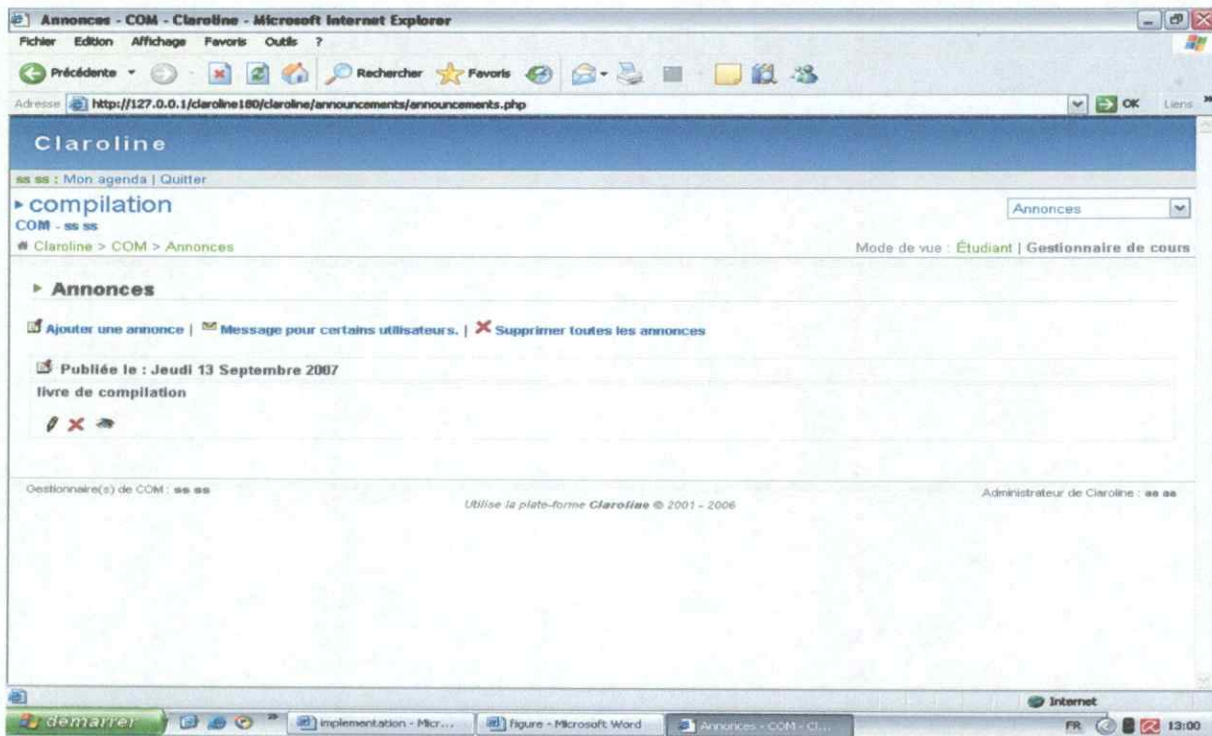


Figure45:les annonces.

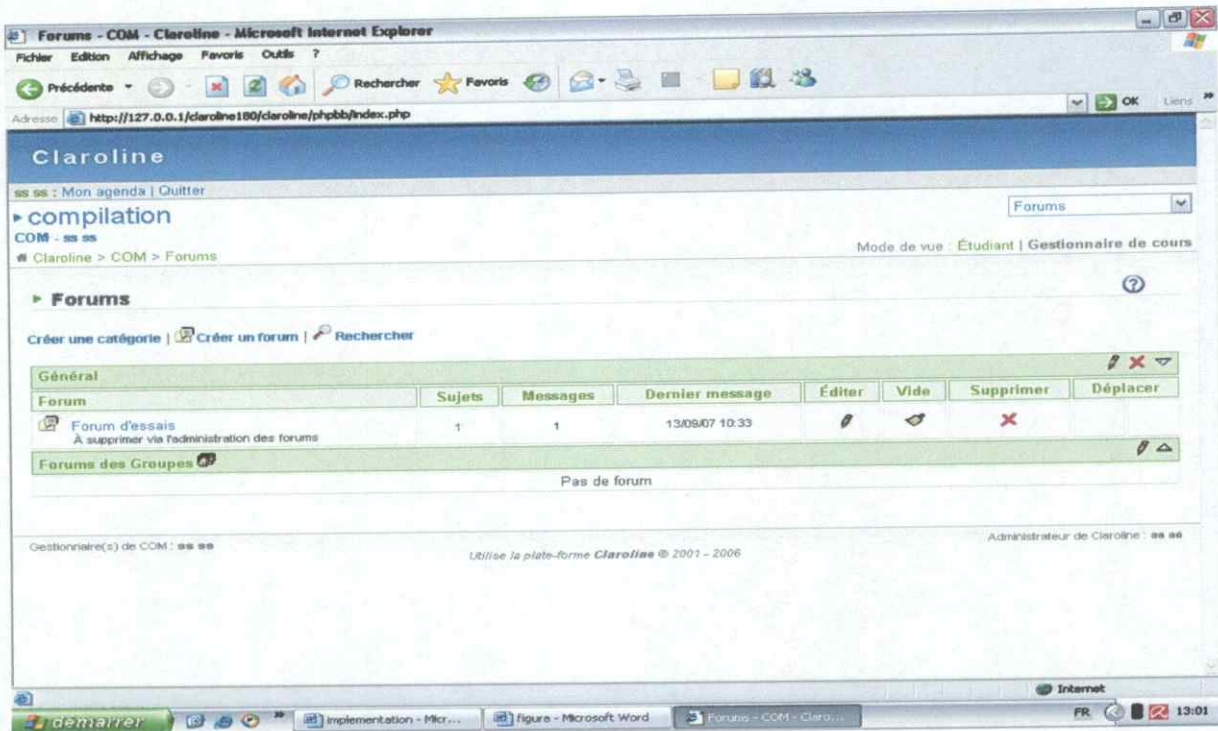


Figure46:forums.

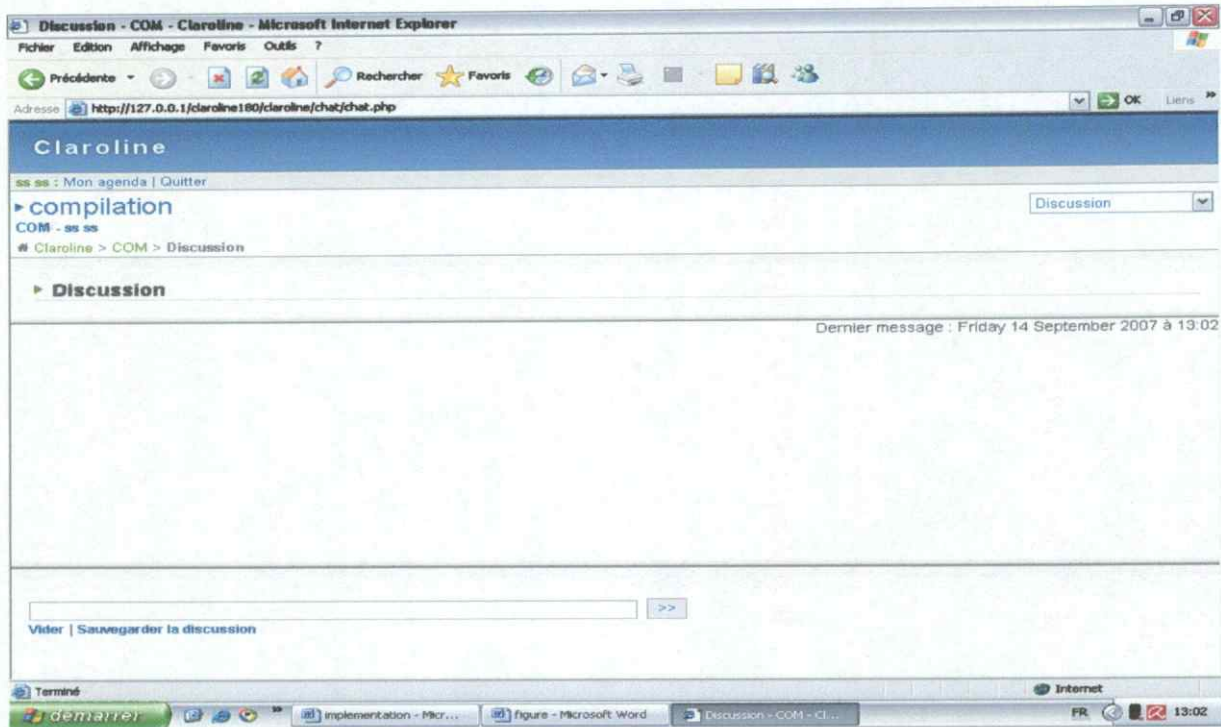


Figure47:discussion.

Claroline - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente Recherche Favoris

Adresse http://127.0.0.1/claroline180/ajout_note.php?ca=1&ex=em&code=c1&typ=c&uens=ss&pens=ss

Claroline

ss ss : Mon agenda | Quitter

Editer les notes

Nom	Prénom	EMD (on line)	EMD
lounaouci	wassila	10	<input type="text"/>
doudeh	amine	15	<input type="text"/>
semmar	omar	0	<input type="text"/>
rtyee	gfyry	20	<input type="text"/>
dgrtr	gtry	5	<input type="text"/>
rfgby	fytrytr	10	<input type="text"/>

Valider

précédent

Terminé Internet

demarrer Implementation - Mic... Figure - Microsoft Word Claroline - Microsoft I...

FR 13:14

Figure48:éditer les notes des examens.

L'enseignant a le choix, il peut conserver les notes de l'examen (on line) et qui sont restituées automatiquement comme il peut éditer d'autre notes.

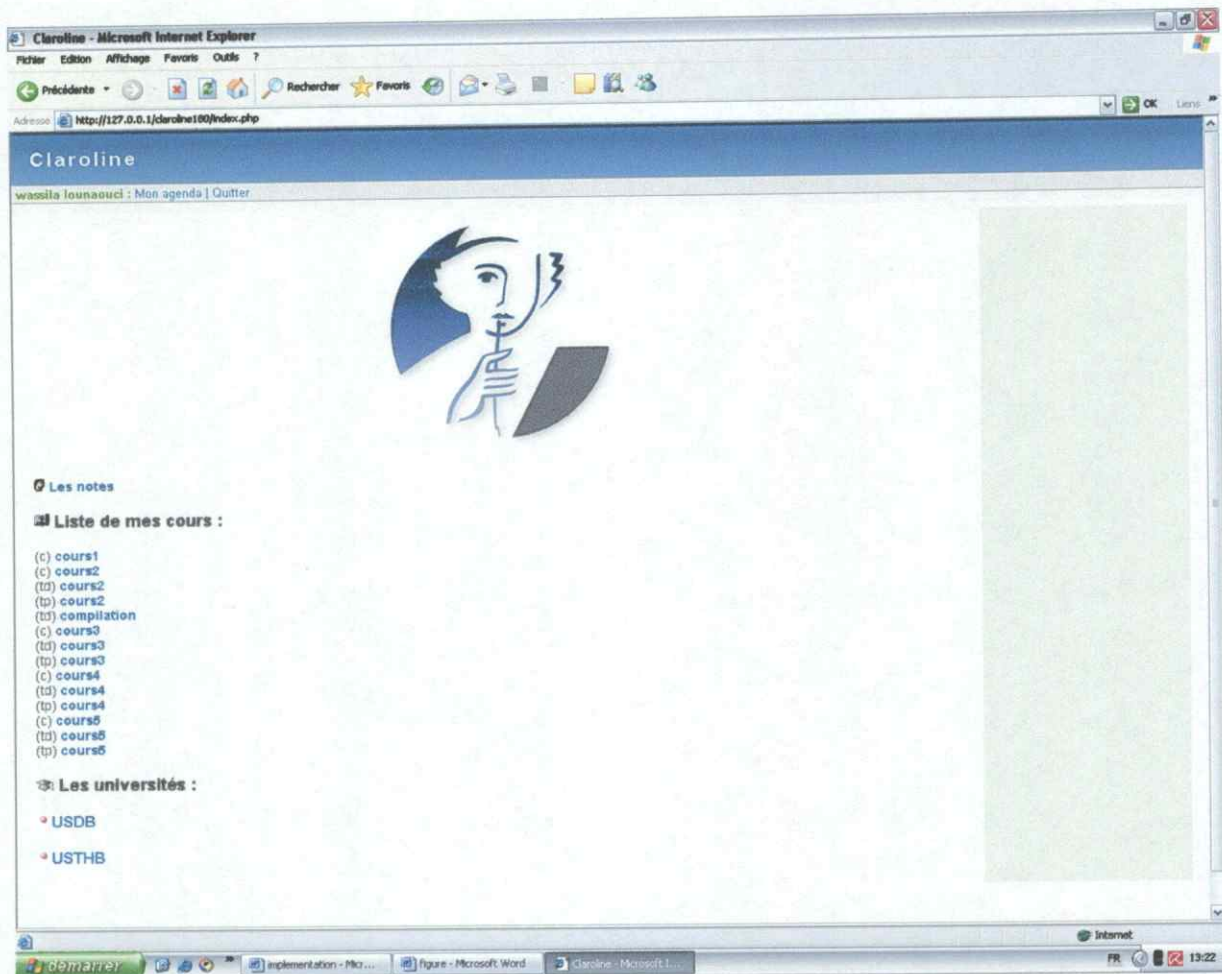


Figure49:page d'accueil de l'étudiant.

En introduisant l'identifiant et le mot de passe, l'étudiant accède a son espace privé.



ClaroLine - Microsoft Internet Explorer
 Adresse http://127.0.0.1/claroline180/notes_st.php?user=ml@pass=mlk

ClaroLine
 wasilla lounaoui : Mon agenda | Quitter

Les notes

Semestre 1
 crédit demandé : 30
 crédit acquis : 30

Unité d'enseignement	cours	crédit du cours	coefficient du cours	EMD	TD	TP	Rattrapage	moyenne du cours
unite1	cours1	10	2	12				12
unite1	cours2	10	2	11	11	12		11.25
unite2	cours3	10	2	14	14	14		14

Unité d'enseignement	moyenne	crédit
unite1	11.625	20
unite2	14	10

Semestre 2
 crédit demandé : 30
 crédit acquis : 30

Unité d'enseignement	cours	crédit du cours	coefficient du cours	EMD	TD	TP	Rattrapage	moyenne du cours
unite3	cours4	20	2	10	11	11		10.5
unite4	cours5	10	2	10	12	11		10.75

Unité d'enseignement	moyenne	crédit
unite3	10.5	20
unite4	10.75	10

Crédit total	Décision
90	admis

Terminé

Figure50:les notes de l'étudiant.

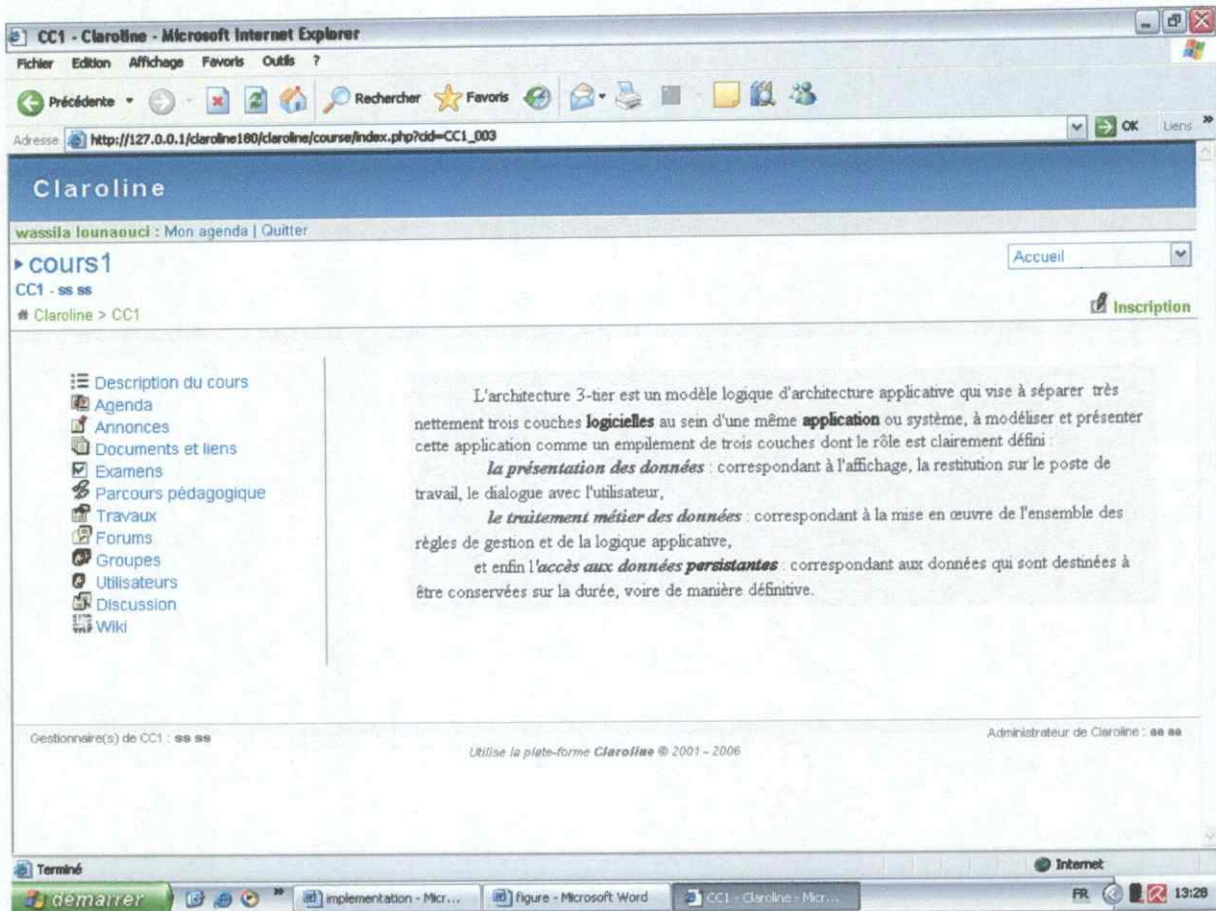


Figure51:page d'accueil du cours pour de l'étudiant.

Conclusion générale

Le travail que nous avons présenté constitue notre premier contact avec le monde du travail, il nous a offert la possibilité d'approfondir nos connaissances et de les concrétiser à l'aide d'un travail pratique.

Grâce à cette expérience nous avons pu comprendre les différents aspects et fonctionnement de la gestion des plates formes de E-learning ainsi que les problèmes reliés à ce domaine.

Nous avons appris l'analyse et la conception à l'aide de la description UML, en implémentant cette conception on a élargie notre apprentissage par un environnement de programmation puissant (php, html, mysql, java).

C'est grâce à ces différentes connaissances qu'on a pu réaliser ce modeste travail.

A l'issue de ce travail, la perspective suivante peut être proposée pour poursuivre les réalisations dans ce domaine :

- Concevoir une interface homme machine en 3D pour les universités

Enfin, nous espérons avoir atteint l'objectif qui nous a été fixé , et que ce travail aussi modeste soit-il, contribuera à aider toute personne désirant travailler sur les plates formes et constituera un point d'appui pour d'autres projets.

Bibliographie

[L-01] Modélisation objet avec UML

Pierre-Alain Muller

Nathalie Gaertner

Edition Eyrolles

[L-02] Le guide de l'utilisateur UML

Grady Booch

James Rumbaugh

Ivar Jacobson

Edition Eyrolles

Webographie

[w-01] <http://www.learnperfect.fr/elearning/index.htm>

[w-02] <http://icb.u-bourgogne.fr/universitysurf/plateformes.html>

[w-03] http://www.refer.ga/IMG/pdf/Etude_comparative.pdf

[w-04] <http://www.claroline.net>

[w-05] <http://cedric.babault.free.fr>

[w-06] <http://www.easyphp.org>

[w-07] <http://www.phpzero.fr>

[w-08] <http://www.lephpfacile.com>

[w-09] <http://www.php.net>

[w-10] <http://uml.org>