

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'Etat en informatique**

Option : Intelligence Artificielle

Sujet :

Conception et Réalisation d'un outil pour le
 contrôle de déroulement d'une séance de
 TP en informatique.

Présenté par : ARBAOUI Abdessamed
 MAZOUNI Hamza

Promoteur : M. Abdelkrim Chebieb

Organismes d'accueil :

Soutenu le date, devant le jury composé de :

Président :

Examineur :

Examineur :

2007- 2008

MIG-004-216-1

REMERCIEMENT

Avant tout nous remercions dieu de tout puissant qui nous a donné
la force du parvenir au bout de nos peines.

Nous à adresser nos sincères remerciements à notre promoteur
M.ABDELKRIM CHEBIEB

qui a su nous conseiller par ses critiques constructives
et nous guider dans notre travail.

Nous remercions aussi les membres de département informatique,
surtout les membres de jury.

Nous remercions tous les travailleur et les enseignants de l'institut
national d'informatique (INI).

Nos sentiments de profonde gratitude vont à nos professeurs qui
ont nous transmit leur savoir sans réserve
pendant tous les années des études.

Nos remerciements vont aussi à tous ceux et celles qui ont participé
de prés ou de loin à l'élaboration de présent travail.

Enfin, nous tenons à remercier tous nos amis et collègues
pour leur soutien moral tous au long de la préparation
de ce mémoire.

DEDICACES

Je dédie ce modeste travail, tout particulièrement, à mes très chers parents pour leur inestimable soutien moral et ce tout au long de mon parcours.

Je tiens également à dédier ce mémoire à mes frères, à mes sœurs ainsi qu'à tous mes amis et collègues pour les encouragements et conseils qu'ils m'ont prodigués.

Biloo

DEDICACES

Je dédie ce modeste travail

A mes très chers parents, pour leur soutien ; ma mère qui n'a pas cessé de prier pour moi et de m'encourager dans les moments difficiles. Mon père, qui m'a toujours soutenu et aidé à affronter les difficultés.

A mes frères : Mustapha, Hicham et le petit Fouad.

A mes sœurs : Samia et Khadîdja.

A toute ma famille.

A tous mes amis

A tous ceux et celles dont les noms n'ont pu être cités.

Hamza

Résumé

Le présent travail a pour objectif la réalisation d'un outil pour le contrôle d'un atelier d'apprentissage, d'outils informatiques tels que les travaux pratiques d'ingénieurs informaticiens sur le C. L'outil doit permettre au maître de la séance un contrôle total des éléments du groupe à partir de sa machine : assister les étudiants, vérifier leur travail, arrêter la machine d'un étudiant ou le remettre à l'ordre s'il sort du cadre du TP, diffuser le travail aux étudiants du groupe, permettre à un étudiant de communiquer avec le maître pour poser sa question ou répondre à une question qui lui est posée...

Ce système va être développé pour servir de plate forme pour l'utilisation des techniques de développement des interfaces home machine qui séparent le noyau de l'interface, le modèle d'architecture MVC qui va être utilisé dans cette application.

Mots clés : IHM, Systèmes collaboratifs, Modèle d'architecture d'IHM, modèle centralisé.

Abstract

This work aims at the realization of a tool for the control of a workshop of training, computer tools such as practical works of computer science engineers on C. the tool must allow to the Master of the meeting a total control of the elements of the group starting from its machine: to assist the students, to check their work, to stop the machine of a student or to give it to the order if it leaves the framework of the TP, to diffuse work with the students of the group, to allow a student to communicate with the Master to put his question or to answer a question which is asked to him...

This system will be developed to be used as platform for the use of the techniques of development of the interfaces home machine which separate the core from the interface, the model of architecture MVC which will be used in this application.

Key words: IHM, Systems collaboratifs, Model of architecture of IHM, model centralized.

المخلص

الهدف من هذا العمل هو انجاز برنامج للتحكم في ورشة للأعمال التطبيقية مثل لأعمال الموجهة لطلبة الإعلام الآلي هذا البرنامج يسمح للأستاذ الحصة في التحكم الكامل في أفراد الفوج من خلال الجهاز الذي بحوزته و بالمقابل يسمح للطلاب التجاوب مع الأستاذ من خلال طرحه للأسئلة أو الإجابة على الأسئلة المطروحة من طرف الأستاذ، هذا النظام سوف ينجز في شكل نواة وواجهة باستعمال أساليب

IHM

مفاتيح: IHM, أنظمة التنسيق الجماعي, المخطط المركزي

SOMMAIRE

Introduction Générale

1. Contexte de l'étude	1
2. Objectif du mémoire.....	2
3. Méthodologie	2

CHAPITRE I : Généralités sur Les Systèmes Collaboratifs

I. Introduction	3
II. Pourquoi étudier l'approche collaborative.....	3
III. Le Travail Collaboratif.....	4
III.1. Travail collaboratif asynchrone	5
III.2. Travail collaboratif synchrone	5
2. Travail collaboratif dans le cadre d'apprentissage.....	6
IV. Groupware ou Collecticiel	8
V.1. Dimensions et Classification des collecticiels.....	8
V.1.1 Les Types de collecticiels	9
V.1.2 Classification Fonctionnelle des Collecticiels	10
V.2. Les Modes de Collaboration	11
V.2.1. La collaboration asynchrone.....	11
V.2.2. La collaboration en session.....	12
V.2.3. La collaboration en réunion.....	12
V.2.4. La collaboration étroite.....	12
V.3. Eléments d'implémentation des collecticiels	12
V.4. Les Applications des Collecticiels.....	14
V.5. Les avantages des collecticiels.....	16
V.6. Les contraintes d'utilisation des collecticiels.....	17
V.7. Architecture logicielle des collecticiels	18
IV. Conclusion	20

CHAPITRE II : Etude des quelques applications

Introduction.....	21
I. iTALC.....	21
1. Présentation de iTALC.....	21
2. Les fonctions proposées par iTALC.....	22
2.1. La barre d'outil.....	22
2.2. Ecran des élèves	23
2.3. Les inconvénients de iTALC	24
II. LanSchool	25
1. Présentation de LanSchool.....	25
2. Les Fonctions proposées par LanSchool	25
3. Les inconvénients de LanSchool.....	27
Conclusion	27

CHAPITRE III : Les architecture Client/Server & Peer To Peer

I. Introduction	28
II. Les réseaux client/serveur.....	28
Définition 1.....	29
Définition 2	30
1. Evolution de l'architecture client/serveur.....	30
3. Fonctionnement d'un système Client/Serveur.....	31
2. Avantages et Inconvénients.....	32
III. Les Réseaux Peer To Peer.....	33
1. Caractéristiques des systèmes P2P	34
2. Architectures de réseaux Peer to Peer.....	34
2.1. Architecture décentralisée (pur)	34
2.2. Architectures hybrides.....	35
IV. Conclusion.....	36



CHAPITRE IV : Analyse et Conception

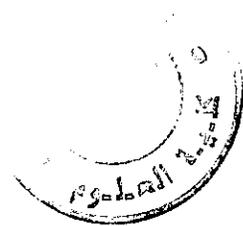
I. Introduction	37
II. Un bref sur la modélisation unifié (UML)	37
III. Détermination des cas d'utilisation	38
IV. Description détaillée des cas d'utilisation	43
V. Diagramme de classe global	105
VI. Architecture du système.....	108

CHAPITRE V : Implémentation et Réalisation

I. Environnement de Développement	110
I.1. Le Langage de programmation	110
I.2. Modèle MVC.....	112
II. Présentation de l'application	115
III. Conclusion	112

Conclusion Générale

Bibliographie



Liste des Figures :

Figures	Description	page
Chapitre I		
Figure01	<i>Exemple de travail collaboratif</i>	4
Figure02	<i>les types des collecticiels</i>	10
Figure03	<i>Trèfle fonctionnel</i>	10
Figure04	<i>le modèle centralisé</i>	18
Figure05	<i>le modèle Répliqué</i>	19
Figure06	<i>le modèle Hybride</i>	19
Chapitre II		
Figure07	<i>La barre d'outil de iTALC</i>	22
Figure08	<i>Interface de l'enseignant pour iTALC</i>	23
Chapitre III		
Figure09	<i>Classification des systèmes informatiques</i>	28
Figure10	<i>architecture client/Server</i>	29
Figure11	<i>Exemple d'architecture Client-Serveur</i>	31
Figure12	<i>L'architecture décentralisée</i>	34
Figure13	<i>L'architecture hybride</i>	35
Chapitre IV		
Figure14	<i>Diagramme de cas d'utilisation- vue de l'administrateur</i>	40
Figure15	<i>Diagramme de cas d'utilisation- vue de l'enseignant</i>	41
Figure16	<i>Diagramme de cas d'utilisation- vue de l'étudiant</i>	42
Figure17	<i>Diagramme de classe global</i>	109
Figure18	<i>Architecture du système</i>	110
Chapitre V		
Figure19	<i>Schéma du MVC</i>	113
Figure20	<i>Interface principale de l'application (Serveur)</i>	115
Figure21	<i>La barre de travail de serveur</i>	115
Figure22	<i>Affichage des étudiants sur serveur</i>	116
Figure23	<i>Créer une nouvelle Séance</i>	117
Figure24	<i>Attribuer des étudiants à une séance</i>	117
Figure25	<i>Attribuer un travail à une séance</i>	118
Figure26	<i>Définir les exercices</i>	118
Figure27	<i>Réponse d'un exercice</i>	118
Figure28	<i>Ajouter étudiant</i>	119
Figure29	<i>Ajouter enseignant</i>	119
Figure30	<i>Envoyer et recevoir des messages</i>	120
Figure31	<i>Gestion des documents</i>	120
Figure32	<i>Interface de l'étudiant avant la connexion</i>	121
Figure33	<i>la barre de travail de l'étudiant</i>	121

Introduction Générale

Introduction

Après une phase d'utilisation « individualiste » de la micro-informatique, l'adoption généralisée des réseaux informatiques a permis de mieux organiser les informations dispersées, d'améliorer la communication entre les utilisateurs et de réduire les dépenses par le partage des périphériques et des applications.

Plus les réseaux sont utilisés dans les écoles, les instituts et les universités plus il devient nécessaire de gérer le processus d'acquisition de connaissances dans les salles de classe en réseau. Le problème est comment trouver un moyen pour une meilleure gestion de ces salles c'est-à-dire un outil qui permet aux enseignants et formateurs d'améliorer l'efficacité des cours en les délivrant sur les ordinateurs des étudiants depuis un poste central, de mieux suivre leurs étudiants en surveillant l'utilisation des applications et d'Internet, et de gagner du temps grâce aux sondages de classe, dont les résultats peuvent être communiqués immédiatement. Enseignants et formateurs peuvent aussi enregistrer toute l'activité écran, clavier et souris d'un poste étudiant pour l'examiner plus tard ou la montrer à la classe.

Contexte de l'étude :

Ce travail s'inscrit dans le cadre de la mise en oeuvre d'une technique de développement séparer des interfaces et du noyau des systèmes interactifs dans le domaine de travail collaboratif et l'utilisation expérimentale des modèles d'IHM tel que MVC ou ARCH, ou d'autres.

Objectif du mémoire :

Concevoir un outil pour le contrôle d'un atelier d'apprentissage d'outils informatiques tels que les travaux pratiques d'ingénieurs informaticien sur le C. L'outil doit permettre au maître de la séance un contrôle total des éléments du groupe à partir de sa machine :

- Assister les étudiants,
- Vérifier leur travail,
- Arrêter la machine d'un étudiant ou le remettre à l'ordre s'il sort du cadre du TP,
- Diffuser le travail aux étudiants du groupe et récupérer leurs travaux,
- Permettre à un étudiant de communiquer avec le maître pour poser sa question ou répondre à une question qui lui est posée...

Ce système va être développé pour servir de plate forme pour l'utilisation des techniques de développement des interfaces home machine séparément du noyau notamment le modèle d'architecture MVC qui va être utilisé dans cette application

Méthodologie :

Pour bien mener notre travail, nous avons structuré notre mémoire comme suit :

La première partie est consacrée à l'état de l'art :

- Généralités sur les systèmes collaboratifs.
- Etude des quelques applications rentrant dans ce cadre.
- Les architectures Client/Server et PeerToPeer : pour faire une comparaison entre les deux afin de choisir laquelle qui est adéquate pour notre application.

La deuxième partie est pour :

- L'analyse et Conception : analyse des besoins globale (ensemble cas d'utilisation), la description des cas d'utilisation, les diagrammes de classes...
- Implémentation et Réalisation : Le modèle MVC pour séparer l'interface de système de son noyau, présentation de l'application.

Et nous terminons par une conclusion générale.

Partie I

Etat de l'art

Chapitre I

Généralités sur les Systèmes collaboratifs

Dans ce Chapitre :

- Introduction.
- Pourquoi étudier l'approche collaborative
- Le travail collaboratif.
- Groupware ou Collecticiel.
- Conclusion.

I. Introduction

Aujourd'hui, un individu ne peut assurer à lui seul la conception, la réalisation, la fabrication et la diffusion d'un produit. Quel que soit le domaine, un produit ou un service est le fruit du travail d'une équipe d'individus souvent géographiquement distribués.

C'est le cas, par exemple, dans les domaines de la conception aéronautique et de la construction automobile. C'est également le cas dans le domaine de la conception de supports de cours, et dans l'utilisation de ces mêmes supports lors de sessions de formation (voir la figure 01). Cette constatation est à l'origine de nombreux efforts dans le domaine de l'ingénierie concourante dans plusieurs secteurs économiques, et a conduit à la mise en place de réseaux intranet et extranet. Ainsi, les premiers travaux ont porté sur la télé-ingénierie asynchrone (ou en temps différé), basée sur le courrier électronique, les transferts de fichiers, etc.

Nous allons dans ce chapitre aborder la notion de travail collaboratif en général et dans le cadre de l'apprentissage en particulier, ensuite nous parlerons des collecticiels ou groupware qui sont des outils dédiés au travail collaboratif.

II. Pourquoi étudier l'approche collaborative ?

L'approche collaborative est le domaine qui va très certainement bouleverser profondément à la fois le fonctionnement des entreprises, les modes de travail et le comportement des personnes au cours des années à venir dans le secteur administratif. Le développement de ce secteur est en fait dû à la conjonction de plusieurs facteurs : le développement des réseaux de télécommunications de plus en plus étendus, l'émergence de nouveaux outils de groupware (messagerie, workflow, ...) et la nécessité de rationaliser la production.

Dans le contexte plus global de la compétitivité, les entreprises développent de plus en plus la collaboration en réseau, que ce soit dans leurs relations avec d'autres via des partenariats ou dans leur gestion interne en valorisant les compétences et la créativité.

De l'automatisation industrielle et des applications de gestion traditionnelles (comptabilité), on passe :

- A la modernisation des outils administratifs par l'automatisation des procédures (workflow),
- Au travail en réseau par la mise en place d'outils facilitant le travail de groupe, l'échange.

III. Le Travail collaboratif : [1186]

Le travail collaboratif est un nouveau modèle économique de production dans lesquels se joint un grand nombre de personnes au moyen de nouveaux outils de communication en vue de maximiser l'énergie créative lors de projet d'envergure (comme une encyclopédie), Le travail collaboratif résulte des deux modalités suivantes :

- organiser le travail en séquences de tâches parallèles,
- fournir aux acteurs de chacune des tâches une information utile et facilement exploitable sur les autres tâches parallèles et sur l'environnement de la réalisation.

La figure01 montre un exemple de travail collaboratif

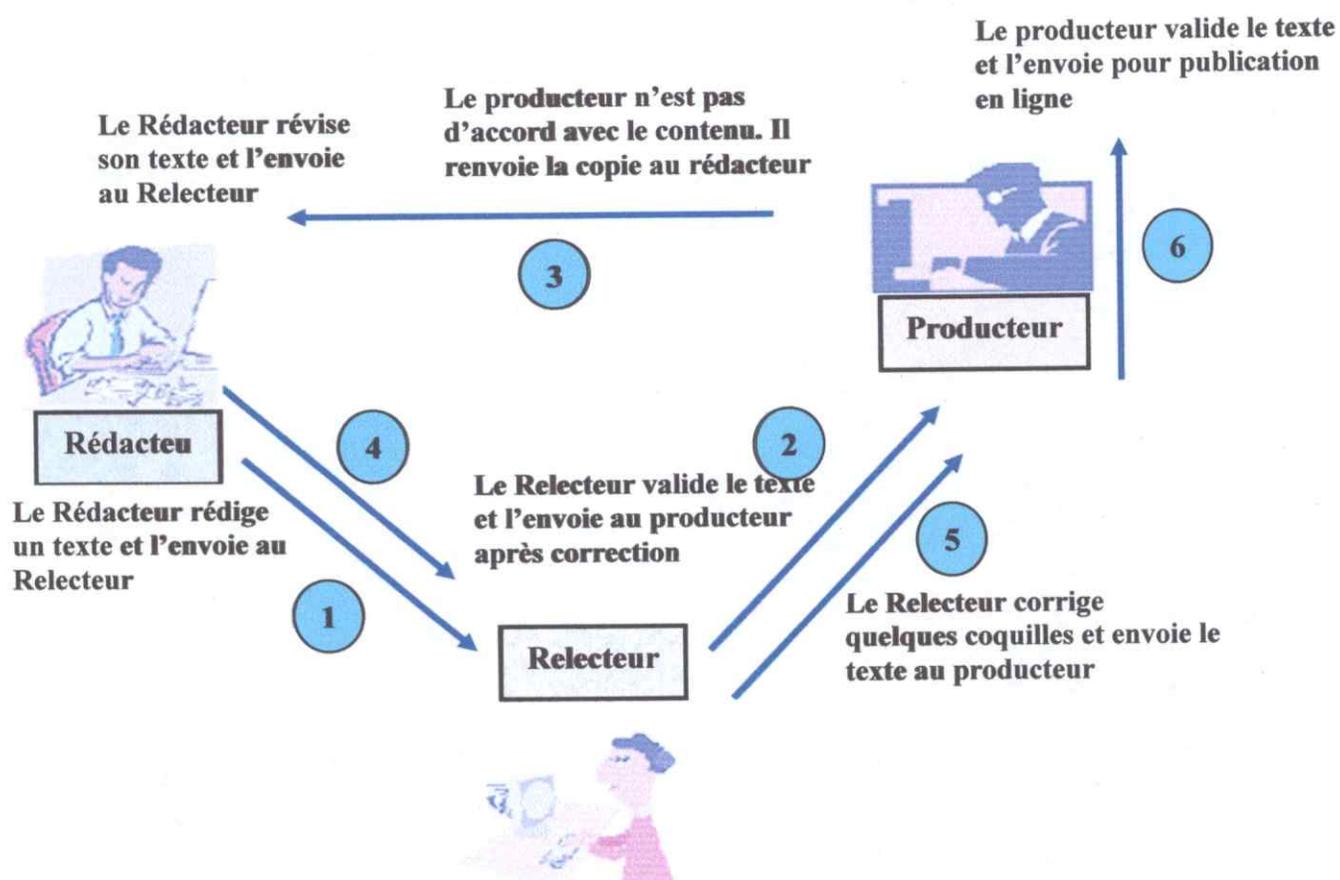


Figure 01 : Exemple de travail collaboratif

III.1. Travail collaboratif asynchrone:

On parle de travail collaboratif asynchrone dans les cas où plusieurs utilisateurs travaillent sur des documents communs, mais pas simultanément. Les premiers environnements de travail coopératif qui sont apparus sont les environnements pour le travail distribué asynchrone qui intègrent sous une interface unifiée des outils de courrier électronique textuels et/ou multimédias, de transfert de fichiers, de consultation de bases de données distribuées, etc.

L'environnement de ce type le plus connu et utilisé aujourd'hui est certainement Lotus Notes ; dans une catégorie de logiciels rendant des services similaires, on assiste également à l'explosion de l'utilisation d'Internet Explorer, de Netscape ou de FireFox. Ces environnements de travail coopératif asynchrone sont apparus avec la libéralisation du réseau Internet et offrent des services dont il semble difficile de se passer dans notre travail quotidien. Cependant, même s'ils offrent aux utilisateurs des moyens de communication performants, ils présentent des lacunes car ils ne permettent que des interactions en temps différé et de ce fait les utilisateurs n'interagissent pas directement sur la conception du produit, ou en liaison synchrone avec leurs collègues et leurs formateurs : ils agissent à tour de rôle, ralentissant ainsi les interactions par rapport à ce qu'elles seraient dans un face à face direct, et surtout privant ces utilisateurs de tous les supports de communication informels, comme le dialogue par exemple, riche d'information. D'où l'intérêt de disposer d'un environnement intégré.

III.2. Travail collaboratif synchrone:

En plus des environnements de travail asynchrone, il faut pouvoir utiliser des services synchrones, si possibles intégrés à l'environnement de travail général précédemment présenté. Les outils de communication multimédias synchrones les plus couramment utilisés et qui composent les principaux environnements de travail coopératif sont :

- Les outils de visioconférence qui permettent des communications par la voix et l'image entre deux utilisateurs ou dans un groupe de travail.
- Les tableaux blancs distribués qui émulent le traditionnel tableau blanc des salles de réunion, permettant aux utilisateurs de bénéficier d'un support (pour dessiner ou écrire) nécessaire à leur exposé ou à leur réflexion. Ces tableaux peuvent

également accepter un « fond d'écran » servant d'état initial aux annotations effectuées par le groupe de travail.

- Les outils de partage d'application qui sont généralement les outils incontournables pour ces environnements. Ces outils permettent aux utilisateurs d'un domaine de partager les logiciels qu'ils utilisent avec les autres participants de la session de travail coopératif. Le partage d'application permet ainsi à un utilisateur de montrer en temps réel l'image de l'application partagée à tous ses interlocuteurs, avec également toutes les évolutions en temps réel, les mouvements des pointeurs, etc. Ces outils permettent souvent aux membres du groupe de prendre la main sur l'application partagée: tous les participants à la session de travail assistent alors aux évolutions de l'application et aux actions de l'utilisateur qui la contrôle.
- Les espaces de travail partagé qui permettent à deux utilisateurs ou à un groupe de travail d'éditer des documents (graphiques, textuels, formels, etc.) coopérativement.

IV. Travail collaboratif dans le cadre d'apprentissage: [DIL 99]

Dans le séminaire "Travail collaboratif assisté par ordinateur" la notion de groupe d'apprentissage est au centre des préoccupations de la communauté enseignant - étudiants et les objectifs déclarés font explicitement mention de la réalisation de tels groupes. L'enseignant, membre à part entière du groupe, y favorise l'interdépendance, s'assure du partage des objectifs, suggère une répartition des tâches ou des fonctions, s'assure des responsabilités de chacun. Les interactions au sein du groupe prennent corps par l'entre aide ou l'apprentissage réciproque ainsi que la mutualisation des informations.

Si on examine l'investissement en temps d'un enseignant ou d'un tuteur qui aurait à s'adresser à chaque apprenant individuellement, on peut affirmer que l'introduction de groupes d'apprentissage, en plus des compétences sociales qu'elle induit, amène une plus value économique. Le dialogue avec un groupe demande moins de temps d'intervention de la part des formateurs que les échanges particuliers et répétés avec les apprenants pris isolément.

Par exemple Kochmann [KOC 01] décrit la discussion d'un cas clinique en deuxième année d'études médicales, comme un cas d'apprentissage collaborative. l'objectif pédagogique n'est pas seulement le diagnostic mais la qualité des méthodes employées pour y parvenir.

Le CSCL¹ s'inspire beaucoup des travaux issus du CSCW², qui nourrissent la réflexion des chercheurs du domaine. Des technologies développées pour soutenir les réflexions et production de groupe, dans des contextes de travail, sont importées, détournées, pour l'apprentissage coopératif. Des idées sont également reprises, mais on constate une certaine complexité des problématiques (l'apprentissage est toujours difficile à mesurer et à observer, les résultats attendus ne sont pas toujours claire et on observe surtout des effets indirects) avec des contraintes souvent opposées (par exemple le processus de collaboration et de travail est plus important que le résultat). Les réflexions liées à une meilleure instrumentation de la communication à distance peuvent déboucher sur des utilisations dans certains contextes de formation en présence. Ainsi, plusieurs systèmes sont conçus afin d'enrichir les interactions dans les cours en amphithéâtre avec l'usage de technologies.

Des travaux antérieurs ont été menés, notamment autour du théâtre électronique à l'université de Liège [LEC 99]. Des questions à choix multiple étaient proposées aux apprenants qui y fournissaient une réponse. Les statistiques des choix retenus étaient affichées en temps réel puis discutées avec le groupe, ce qui permettait un approfondissement du sujet présenté et une régulation de l'action éducative.

D'autres projets existe tel que le projet *activclass* [RAT 03], chaque étudiant dispose d'un PDA et peut poser des questions qui sont regroupées en directe pour l'enseignant durant son cours. Il semble que cela enrichit les questions posées. De même, [WES 03] proposent un dispositif nommé *ConcerStudeo* combinant des PDA, un tableau électronique et un collecticiel. L'enseignant peut poser une question (QCM) et voir les réponses. Notons que les PDA utilisés ne donnent pas de représentation visuelle commune et sont limités à l'envoi de messages.

¹ CSCL=computer supported collaborative learning (ACAO Apprentissage Collaborative Assisté par Ordinateur)

² CSCW=computer supported collaborative work (TCAO Travail Collaboratif Assisté par Ordinateur)

Un groupe R, T, S, K¹ décrit un système avec lequel le professeur navigue et écrit sur une présentation à base de diapositives à partir d'une tablette connectée sans fil. Les étudiants voient les transparents sur un affichage collectif ainsi que sur leur ordinateur (portable ou PDA) à connexion également sans fil. Ils peuvent réagir en cliquant à un endroit des diapositives en sélectionnant dans un menu une option d'annotation, la vue pour le professeur agrège les avis des différents auditeurs, avec un codage de couleur, ce qui lui permet de modifier le fil de son discours. D'après les auteurs, les premières expérimentations apparaissent encourageantes, dans le sens où ce système permet aux étudiants de dépasser certaines inhibitions et aux enseignants d'avoir un retour en temps réel des étudiants.

V. Groupware ou collecticiel : [EGR91]

Le terme « collecticiel » ou « groupware » est connu dans l'industrie comme étant très ambigu. Il ne semble pas y avoir une définition unique de ce qu'est un collecticiel, néanmoins, de façon générale, ce terme réfère aux logiciels dont le but premier est d'améliorer le partage d'information et la prise de décision au sein d'un groupe de personnes engagées dans l'exécution d'une tâche ou d'un but commun.

Les collecticiels offrent un mécanisme en vue de partager des opinions et de l'information rapidement dans une ère où l'information n'est stratégique et nouvelle que pendant quelques heures et où le courrier électronique traditionnel ne répond plus aux besoins des entreprises d'aujourd'hui.

V.1. Dimensions et Classification des collecticiels:

Il existe deux dimensions pour caractériser un collecticiel :

Dimension technologique : consiste à Développer des systèmes informatiques qui Assistent un groupe de personnes engagées dans une tâche commune et qui Fournissent une interface à un environnement partagé.

Dimension sciences humaines : consiste à Essayer de comprendre et étudier La nature et les caractéristiques du travail collaboratif, Les facteurs sociologiques, psychologiques, économiques, etc.

¹ RTSK= Anderson R, Vandegrift T, Wolfman S, Yasuhara K

Le collecticiel emprunte donc à de nombreuses disciplines ou sous disciplines, telles que :

- le multimédia,
- le réseau,
- les systèmes distribués, et les applications réparties en général,
- les interfaces homme-machine,
- la sociologie, la psychologie cognitive, etc.

V.1.2 Les Types de collecticiels :

Le schéma suivant de classification des collecticiels suggère quatre scénarios établis selon deux dimensions : l'endroit et le moment choisis pour le déroulement des réunions.

Salle de Décision (même moment, même endroit) : C'est l'équivalent électronique d'une réunion traditionnelle. Le groupe est assis en demi cercle et face à un écran public (ex : Vidéo-projecteur, Tableau blanc, Mémo)

Réseau local de Décision (moment différent, même endroit) : Ce type de réseau permet aux membres du groupe de travailler chacun de son côté à différents moments. Ex (*Mémoire électronique* : Intranet, Kiosque électronique).

Téléconférence (même moment, endroit différent) : Ce type de collecticiel est requis pour des groupes dont les membres sont dispersés géographiquement mais qui doivent «se rencontrer virtuellement » pour prendre une décision. Ex (*Réunion Virtuelle* : Vidéo/Visio conférences, Partage d'Application, Editeur Synchrones, Shared offices (MediaSpace)).

Groupe de décision Eloigné (moment différent, endroit différent) : C'est le cas d'un groupe de décideurs dispersés géographiquement et qui doivent se rencontrer sur une base régulière. Ex (Courrier électronique, Internet, Forum, Workflow, Outils d'éditions en groupe).

Types de collecticiels

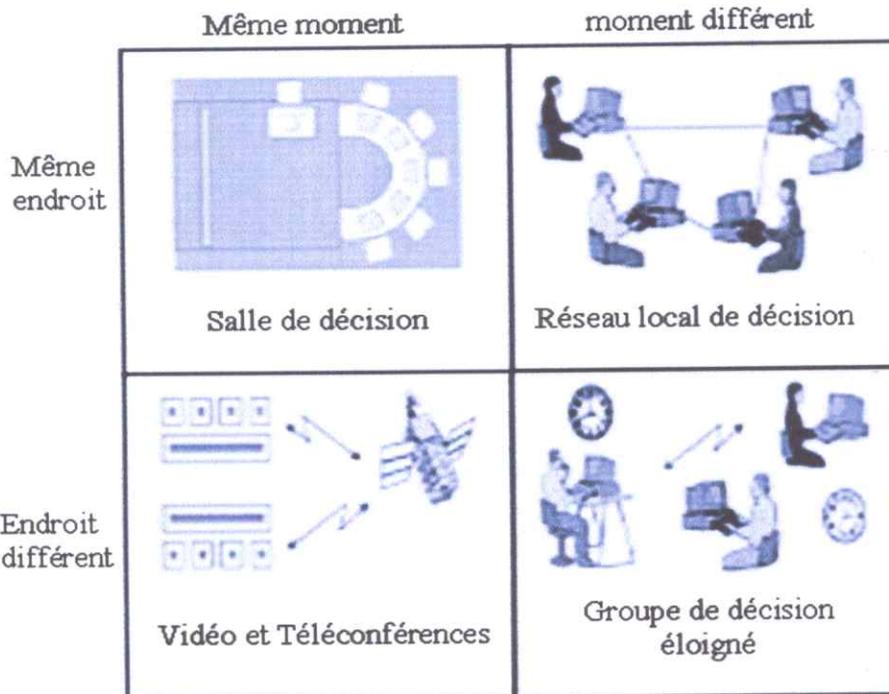


Figure 02 : les types des collecticiels

V.1.3 Classification Fonctionnelle des Collecticiels :

Les outils de collecticiels sont généralement classifiés selon le trèfle fonctionnel [EGR91].

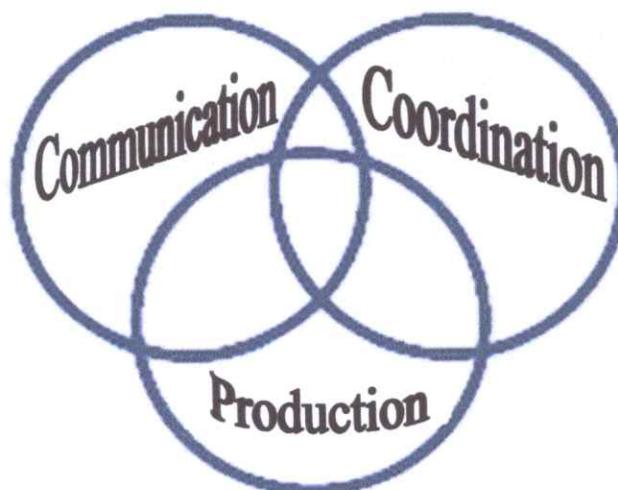


Figure 03 : Trèfle fonctionnel

Production/coopération : Les environnements de coopération sont en fait des environnements de production dans lesquels c'est un groupe qui produit.

Coordination: Produire collectivement nécessite la coordination des acteurs, de leurs actions et de leur production:

- Meilleure gestion des conflits ou maintien de la cohérence des actions
- Séquencement des tâches (+ validation = Workflow) gestion du temps

Communication : Communication Homme Homme Médiatisée (CHHM). Le contenu sémantique de la communication est étranger au système.

Le modèle du trèfle est un modèle fonctionnel du système :

- Décrit les classes de fonction qu'un collecticiel doit implémenter au niveau logiciel
- Décrit les liens entre les fonctionnalités d'une application et l'architecture logicielle.
- Intervient après la description de l'activité collaborative.
- En pratique il oriente l'analyse du travail coopératif
→ Pour adapter les services logiciels aux besoins réels de la collaboration.

V.2. Les Modes de Collaboration:

Quatre modes de collaboration peuvent être utilisés en conception et peuvent être généralisés dans d'autres domaines d'application:

- Collaboration **asynchrone**
- Collaboration **en session**
- Collaboration **en réunion**
- Collaboration **étroite**

→ La réussite d'un travail collaboratif peut se mesurer à la difficulté de supporter l'ensemble des quatre modes et également la trans-modalité (passage d'une modalité à une autre)

V.2.1. La *collaboration asynchrone*:

Les participants interagissent en échangeant les données et en travaillant **quand ils le peuvent**.

→ Dans le contexte de l'entreprise, la collaboration asynchrone correspond au mode de travail autonome.

V.2.2. La collaboration en session:

- Les participants travaillent en même temps sur des données qui leur sont propres.
- Ils peuvent communiquer en même temps, mais sans partager de façon visuelle les objets de leurs discussions.

→ L'objectif principal est de réduire les délais d'interaction entre les membres du projet.

V.2.3. La collaboration en réunion:

• Les participants travaillent et communiquent en même temps tout en partageant les objets de leurs travaux et discussions.

• Ils se voient attribués des rôles en relation avec le but de la réunion. Leurs interventions sont régies par un mécanisme de type «tour de parole».

• La collaboration en réunion impose la participation active (par exemple réponse immédiate aux questions).

→ L'objectif principal est d'accroître la coordination entre les membres du projet.

V.2.4. La collaboration étroite:

• Les participants peuvent travailler, communiquer et interagir en temps réel sur tous les objets partagés du projet.

• Les conséquences de leurs interventions sont directement gérées au niveau des objets manipulés.

• La collaboration **étroite** permet une interaction maximale entre les individus dans un monde cohérent simulant la réalité.

V.3. Eléments d'implémentation des collecticiels :

L'environnement technologique requis pour l'implantation des collecticiels se compose des éléments suivants :

V.3.1. Le matériel

Le minimum de matériel requis pour les collecticiels est un réseau d'ordinateurs personnels ou de stations de travail. En plus dans le cas « salle de décision », on a besoin d'un grand écran public pour projeter l'information aux participants. De même dans le « vidéo et téléconférence », un matériel spécifique est nécessaire pour permettre une communication avec des sons et des images pour des personnes éloignées

géographiquement. Ces équipements comprennent des caméras, des vidéos, des microphones et des haut-parleurs. Enfin, pour le « groupe de décision éloigné », un réseau étendu est nécessaire.

V.3.2. Les utilisateurs

La composante « utilisateurs » d'un collecticiel inclut les membres du groupe et un facilitateur (présent surtout dans les réunions même endroit/même moment) qui est responsable du bon déroulement de la réunion surtout en ce qui concerne l'utilisation de la technologie.

V.3.3. Les procédures

Il s'agit des procédures qui permettent l'usage efficace de la technologie par les membres du groupe. Ces procédures peuvent s'appliquer au fonctionnement du matériel et du logiciel, ou elles peuvent inclure des règles concernant les discussions verbales entre les membres du groupe pendant une réunion.

V.3.4. Interface Homme-Machine:

•L'interaction homme-machine dans les collecticiels est plus complexes que celle dans les applications mono-utilisateur.

→Enrichir les outils de *production* classiques avec des outils de *conversation*, de *communication* et de *coordination*).

→Prendre en compte les surcharges cognitives qui peuvent survenir (suite à la collaboration à distance).

→Prendre en compte certaines adaptations individuelles

→Permettre aux utilisateurs de maîtriser les informations transmises les concernant.

•Il s'agit de permettre d'agir sur toutes les activités dans les quatre espaces du modèle3C

→Conversation, coordination, production et communication.

•Donner une priorité à l'interface Multi-Utilisateurs qui doit rendre compte de l'activité du groupe.

→Faire évoluer les IHMs vers des interfaces homme-machine-homme en respectant certains critères.

V.4. Les Applications des Collecticiels :

Les applications des collecticiels peuvent se regrouper dans les grandes catégories suivantes :

1. Le partage et la diffusion d'information

Un usage possible des collecticiels consiste à partager et à diffuser les documents internes à l'organisation. Le courrier interne devient ainsi accessible par le moyen de collecticiels tel Lotus Notes par exemple, qui permet de créer une base de données sur le serveur de l'entreprise et d'accéder aux documents de référence par le biais des postes clients. Il devient possible aux employés d'accéder aux informations internes en procédant à des requêtes par mots clés en vue de ne lire que le courrier qui les intéresse.

2. Le courrier électronique

Les collecticiels permettent de faire du courrier électronique de façon plus sophistiquée. Le principe est le même que le courrier électronique classique avec envoi et réception de messages mais les fonctionnalités sont bien plus riches en ce qu'il est possible de tirer profit du multimédia. Un exemple de collecticiel qui offre ce type de messagerie est cc : Mail qui est une famille de plus de vingt produits destinés à fournir des fonctionnalités de messagerie électronique multimédia aux utilisateurs des cinq principaux systèmes d'exploitation d'ordinateurs personnels (Macintosh, MS-DOS, OS/ 2, Windows et Unix). Ainsi, cc : Mail permet la connectivité avec les systèmes de messagerie électronique des mini et grands systèmes, possède des passerelles vers les messageries publiques et permet la transmission de télécopies directes de et vers des télécopieurs partout dans le monde. Beaucoup de sociétés utilisent cc : Mail comme messagerie pour faire du transfert de fichiers ainsi que pour du « workflow », la gestion de formulaires électroniques et l'intégration d'applications.

3. La gestion des agendas des membres d'un groupe

La gestion des agendas est une propriété importante des collecticiels. Ainsi, il devient possible de consulter les disponibilités des collègues de travail en vue de fixer un rendez-vous que chaque participant pourra confirmer ou non selon l'évolution de son agenda.

4. La gestion de projet

Les collecticiels permettent à tous les participants, et non plus uniquement aux gestionnaires de projet, de faire le suivi de projet. Ceci est possible puisque tous les membres du groupe peuvent accéder aux données et connaître à tout moment l'état

d'avancement du projet et les rapports d'activité de chaque participant. La bonne conduite du projet devient la responsabilité de tout le groupe et chacun doit y veiller.

5. La génération et l'évaluation d'idées

Les collecticiels offrent un support technologique aux activités de remue-méninges afin de générer des idées et des commentaires de façon non structurée. Les participants répondent à une question ou aux commentaires des autres. La participation est simultanée et anonyme. Les idées générées pourront être structurées selon des catégories, commentées et évaluées par les participants. Cette application des collecticiels a été précieuse pour la conception de nouveaux processus d'affaires dans plusieurs entreprises dans le cadre d'une Ré ingénierie des Processus d'Affaires.

6. Le choix d'alternative et le vote en groupe

Plusieurs collecticiels permettent aux membres d'un groupe d'effectuer un choix entre des alternatives proposées en donnant des poids à une liste d'alternatives selon une liste de critères. Les résultats peuvent être visualisés à l'aide de graphiques afin de faire des simulations en ajustant les poids. Le vote est un outil pour l'évaluation d'options à choisir. Les participants ont différents outils pour voter, soit le oui/non ou la priorisation. Cette application des collecticiels est intéressante surtout que l'anonymat est préservé et donc le vote ne sera pas biaisé par des considérations personnelles

7. La rédaction d'un document en groupe

Les collecticiels permettent aux membres d'un groupe de créer un document qui peut être un plan stratégique à adopter, un nouveau projet, un dictionnaire commun, etc. À titre d'exemple, le collecticiel GroupSystemV de l'entreprise Ventana offre ces fonctionnalités et permet aux membres du groupe de créer, d'éditer et d'annoter le même document. Il est aussi possible d'offrir un support à l'information en permettant au groupe de bâtir un dictionnaire constitué d'une liste de termes qui ont la même signification pour tous les participants.

8. Le sondage d'opinions par questionnaire

Les collecticiels permettent de distribuer un questionnaire préparé sur les postes des participants. Les données obtenues des participants peuvent être compilées dans un même rapport. Cette application est intéressante dans plusieurs contextes dont le lancement d'un nouveau produit, le choix de campagnes de publicité, l'évaluation d'un programme de formation proposé, le choix du traiteur pour le party de Noël, etc.

9. Le développement d'applications

Les collecticiels peuvent être utilisés par un groupe d'analystes concepteurs de systèmes d'information et d'utilisateurs pendant toutes les phases du cycle de développement d'un système d'information. En effet, plusieurs entreprises utilisent ces outils pour cerner les besoins en information des utilisateurs.

V.5. Les avantages des collecticiels :

Elimination des contraintes de temps et d'espace

Les membres du groupe peuvent travailler sur un projet ensemble sans être physiquement dans la même pièce au même moment. Ils peuvent écrire des documents en commun, générer des idées, discuter de problèmes, voter l'importance des préoccupations, etc. en travaillant de leur bureau, de chez eux ou même d'ailleurs. Cela signifie que plus de personnes peuvent participer dans un projet et qu'il y a une grande flexibilité pour organiser le travail grâce aux « réunions virtuelles ».

Communication en parallèle et réduction du temps de prise de décision

Dans une réunion traditionnelle, si 10 membres du groupe participent de façon égale à la réunion, ils auront un temps de parole de 6 minutes chacun et 54 minutes d'écoute des autres (ou du moins 54 minutes de non-participation); ce qui représente un moyen peu efficace de mener des réunions. De plus, nous savons que dans la plupart des réunions, le temps de parole est inégal en ce que 20 % des participants monopolisent 80 % du temps de parole. Les opinions sont partagées bien plus rapidement puisque la communication se fait en parallèle. IBM a réalisé une économie sur les temps de réunions de 56 % grâce aux collecticiels et Boeing a gagné par ce même biais 11 478 heures sur 64 réunions avec 1 000 participants.

Amélioration de la participation

Les collecticiels encouragent les membres à participer aux activités de façon plus libre et plus active. L'anonymat est permis du fait de la séparation entre l'identité et les idées des participants. Dans les réunions traditionnelles, plusieurs bonnes idées ne sont pas exprimées par crainte des participants d'être mal vus par le supérieur hiérarchique ou par timidité. De plus, les participants peuvent sentir une pression volontaire ou involontaire du groupe en vue de se conformer aux idées dominantes. L'anonymat,

permis par les collecticiels, conduit à plus d'honnêteté dans les commentaires et donc, donne lieu à une réunion plus productive.

Structuration des activités de groupe

L'usage d'un collecticiel rend possible l'établissement d'un guide structuré du travail et il devient plus difficile aux participants de ne pas le suivre.

Mémoire organisationnelle

La contrainte d'établir après chaque réunion un procès-verbal n'a plus lieu d'être. Avec les collecticiels, toutes les informations sont enregistrées et chacun peut aller consulter les informations entrées par les autres. Cela donne une flexibilité telle que les participants peuvent attendre le moment opportun pour entrer dans la discussion et ne risquent pas de « perdre le fil » s'ils ont fait une pause pendant la réunion. La synergie de groupe n'en est que plus forte puisque les participants prennent le temps de penser, de s'imprégner des commentaires des autres. La mémoire organisationnelle est précieuse en cas d'oubli d'éléments établis pendant la réunion parce qu'il est toujours possible de consulter les documents qui s'y rattachent.

V.6. Les contraintes d'utilisation des collecticiels:

Le style de gestion des dirigeants de l'entreprise est important et constitue un terrain favorable ou pas à l'introduction de collecticiel. Les collecticiels ont besoin d'une érosion du modèle top-down de la gestion d'entreprise en vue de passer à une prise de décision selon un mode plus collégial et participatif. Si le style de gestion est trop centralisé, il peut entraver la réussite de l'implantation du collecticiel. Ainsi, la réussite ou l'échec de l'implantation de ces systèmes dépend de facteurs qui se chevauchent dont les plus importants sont le style de gestion, l'intégration ou pas avec les processus d'affaires et la formation des utilisateurs.

Il va sans dire que l'investissement dans les collecticiels est assez coûteux et que toutes les entreprises ne peuvent pas se le permettre. De plus, l'introduction d'un collecticiel dans une organisation devra être bien préparée en matière de réflexion sur le contenu de formation des utilisateurs, des coûts correspondants et des mesures à prendre pour éviter ou réduire la résistance au changement. De plus, la maintenance de tels équipements et logiciels peut s'avérer coûteuse et le coût d'une panne éventuelle

pourrait être élevé à cause de la dépendance qui ce sera constituée par rapport à cette technologie; ainsi, il est possible qu'une simple panne puisse entraver le bon fonctionnement d'une entreprise qui se serait réorganisée autour de son collecticiel.

V.7. Architecture logicielle des collecticiels :

Les architectures des collecticiels séparent le noyau fonctionnel qui Implémente les fonctionnalités et les concepts du domaine indépendamment de leur présentation de l'interface utilisateur pour faciliter une conception itérative de l'interface, pour favoriser la réutilisation et pour favoriser la portabilité du logiciel.

On distingue trois (3) modèles d'architectures :

Le Modèle centralisé : (figure 04) :

Les premiers collecticiels reposaient sur une architecture *centralisée*. Dans ce type d'architecture, non seulement l'état global de l'application se trouve à un endroit bien défini (par exemple, au sein d'un seul processus), mais également la vue qu'a chaque utilisateur.

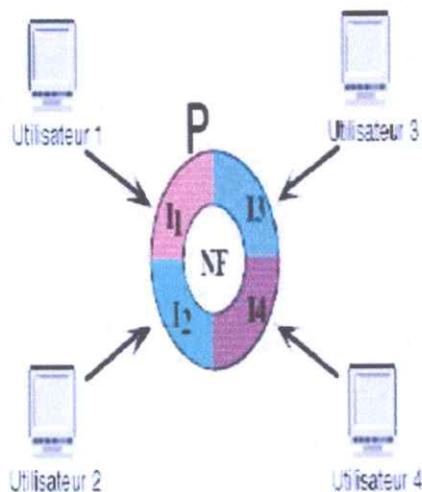


Figure 04 : le modèle centralisé

Avantage : simple à implémenter.

Inconvénients : plus de temps de réponse moins fiable.

Le Modèle Répliqué : (figure 05) :

En général, les collecticiels temps-réel sont trop demandeurs en ressources machine pour qu'il soit raisonnable de les implémenter de façon centralisée. C'est certainement le cas si de la vidéoconférence est en jeu. Pour disposer des performances voulues, il est donc nécessaire de distribuer les applications.

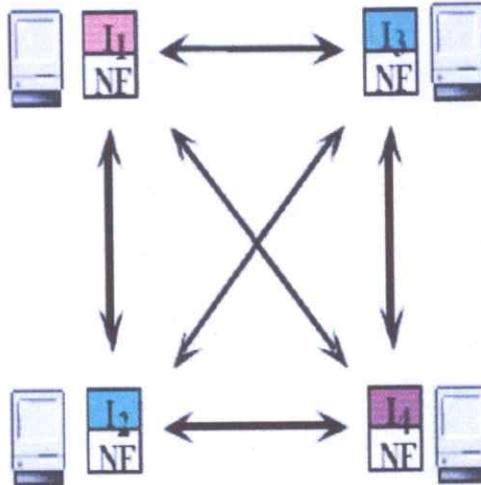


Figure 05 : le modèle Répliqué

Avantage : temps de réponse rapide.

Inconvénient: difficile à implémenter et à gérer la cohérence.

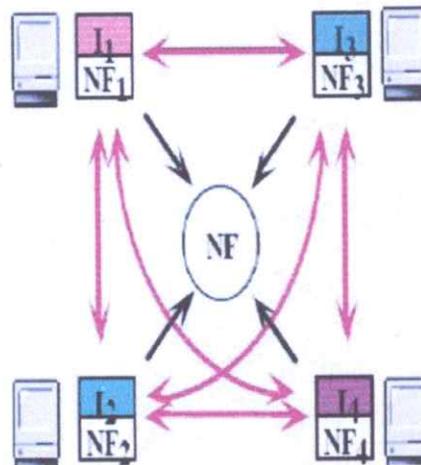
Le Modèle Hybride : (figure 06) : regroupe les deux architectures

Figure 06 : le modèle Hybride

Avantage : temps de réponse plus ou moins rapide

Inconvénients : difficile à implémenter et à gérer la cohérence.

VI. Conclusion :

Accompagnant la pression institutionnelle autour de la nécessité d'intégration des technologies, à tous les niveaux de l'éducation, de nombreux discours voient dans les activités collaboratives une évolution incontournable, notamment dans les formations universitaires. On en arrive à croire que les apprenants sont automatiquement intéressés à coopérer. Il n'y a qu'à leur fournir des moyens techniques appropriés, un peu de formation... et le tour est joué. En particulier, dans le cadre de la formation à distance, c'est un bon moyen de favoriser la participation.

Il y a beaucoup d'obstacles montrant la nécessité de mise en place des technologies pour l'apprentissage, de reproblématiser dans des situations particulières les modalités de leur utilisation, sans croire à des effets de transfert de bonnes pratiques dont les déterminants échappent souvent. Vaste chantier pour la recherche dans le domaine des technologies issues de l'informatique à l'éducation et à la formation. Vaste chantier pour la diffusion d'une culture de ces technologies permettant aux différents acteurs de la formation d'inventer de nouvelles pratiques.

Nous avons présenté dans ce chapitre le travail collaboratif et nous avons vu qu'il peut être utilisé pour améliorer le rendement de l'apprentissage et de l'enseignement en général. Dans le chapitre suivant nous allons présenter quelques applications qui utilisent le travail collaboratif dans l'apprentissage.

Chapitre II

Etude des quelques Applications

Dans ce Chapitre :

- iTALC.
- LanSchool.

Introduction :

Après avoir dans le chapitre précédant que le travail collaboratif peut être utilisé pour l'enseignement notamment l'apprentissage, nous présentons dans ce chapitre quelques applications les plus connues sur le marché tel que iTALC, LanSchool afin de comprendre mieux les fonctions d'un collecticiel dédié à l'enseignement et nous en servir pour l'étude et la définition des grandes lignes l'outil objet de ce mémoire.

I. iTALC :

I.1. Présentation de iTALC :

iTALC¹ est une interface de contrôle se basant sur le protocole VNC. Il permet le monitoring d'une salle informatique d'un coup d'oeil, prendre le contrôle d'un ordinateur, faire une démonstration sur tous les ordinateurs ou partie, et bloquer l'interface si besoin.

Le logiciel comprend deux parties : une partie cliente (à installer sur tous les postes) et une partie maître (à n'installer que sur le poste du professeur). Afin d'augmenter la sécurité du logiciel, le poste maître crée une clé publique qui doit être reprise pour chaque poste élève, afin de pouvoir se connecter. Ceci empêche une prise de contrôle venant d'un ordinateur externe. Une fois le logiciel installé sur tous les postes, il est possible, sur le poste maître, d'ajouter les ordinateurs élèves et de les grouper sous forme de classes (pour créer des groupes dans la salle, ou encore pour gérer plusieurs salles en même temps). Une fois ces classes créées, on peut voir d'un coup d'oeil tous les ordinateurs de cette classe.

Les fonctions utilisables sont simples et efficaces : outre la simple surveillance des différents écrans, on peut prendre le contrôle d'une machine, montrer son écran à tous les élèves, montrer le travail d'un élève en particulier, envoyer des messages au groupe ou à un élève en particulier, bloquer l'accès à l'ordinateur (en cas de bêtise ou le temps d'expliquer quelque chose), redémarrer ou arrêter un ou plusieurs postes (à la fin de la séance, par exemple). Des fonctions manquent encore et sont prévues pour les versions à venir : envoi de fichiers sur les ordinateurs cibles.....

¹ - L'acronyme iTALC signifie Intelligent Teaching And Learning with Computers, soit en français Enseigner et apprendre intelligemment à l'aide d'ordinateurs

I.2. Les fonctions proposées par iTALC :

I.2.1. La Barre d'outil :



Figure 07 : La barre d'outil de iTALC

- **Salle** : Cette icône permet de choisir la salle à contrôler.
- **Aperçu** : Affiche un aperçu de tous les écrans des élèves.
- **Démo (on/off)** : Cette fonction affiche votre écran sur tous les écrans des élèves, afin de réaliser une démonstration. Dans ce mode, les élèves ne peuvent réaliser aucune autre action que regarder votre démonstration, diffusée en plein écran.
- **Démo/Windows (on/off)** : Cette fonction affiche votre écran sur tous les écrans des élèves sous forme de fenêtre, afin de réaliser une démonstration. Dans ce mode, les élèves peuvent réduire ou fermer la fenêtre dans laquelle s'affiche votre démonstration.
- **Lock All** : Cette icône bloque l'utilisation des ordinateurs des élèves en affichant un écran noir avec un cadenas.
- **Message texte** : Permet d'envoyer un message texte aux élèves, auquel ils ne peuvent pas répondre.
- **Allumer** : Permet d'allumer tous les ordinateurs de la salle, à distance. Cela n'est faisable que si les adresses IP et adresses MAC sont enregistrées, si la carte réseau est compatible et si l'option WakeOnLan est activée dans le BIOS de l'ordinateur.
- **Eteindre** : Éteint tous les ordinateurs de la salle.
- **Connexion** : Cette fonction vous propose de se connecter à un ordinateur en utilisant un nom d'utilisateur, en local ou à un domaine. Les nom d'utilisateur, mot de passe et nom du domaine sont nécessaires.
- **Ajuster/aligner** : Cette fonction aligne et ajuste les écrans des élèves à la taille la plus grande possible, dans l'ordre dans lequel vous les avez placés.
- **Auto-vue** : Cette fonction aligne et ajuste les écrans des élèves à la taille la plus grande possible, en les plaçant dans le même ordre que dans la liste de gauche.

1.2.2. Ecran des élèves :

En faisant un double clic gauche sur l'écran d'un élève, vous en prenez immédiatement le contrôle. Si vous faites un clic droit sur l'écran d'un élève, vous obtenez un menu proposant différentes actions.

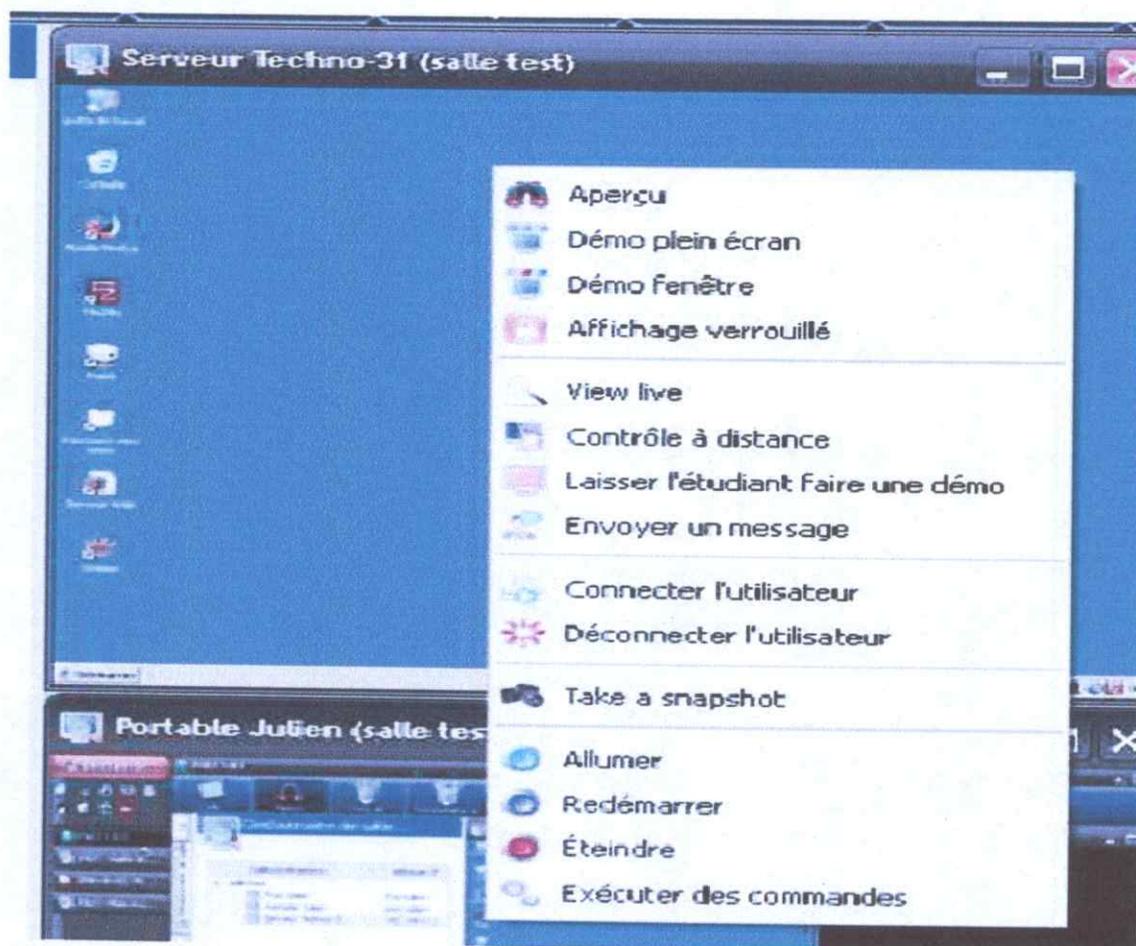


Figure 08 : Interface de l'enseignant pour iTALC

- **Démonstration plein écran :** Cette fonction affiche votre écran sur l'écran de cet élève, en plein écran.
- **Démonstration fenêtre :** Cette fonction affiche votre écran sur l'écran de cet élève, dans une fenêtre qu'il peut réduire ou fermer.
- **Affichage verrouillé :** Cette fonction bloque l'utilisation de cet ordinateur en affichant un écran noir avec un cadenas.
- **View live :** Cela vous permet d'afficher l'écran de l'élève en plein écran sur votre ordinateur, mais ne pouvez pas intervenir.
- **Contrôle à distance :** Cela vous permet de prendre en main la station d'un élève.

- **Laisser l'étudiant faire une démo** : L'écran de cet élève sera affiché sur tous les autres écrans de la salle.
- **Envoyer un message** : Envoyer un message texte à cet ordinateur. L'élève ne peut pas y répondre.
- **Connecter l'utilisateur / Déconnecter l'utilisateur** : Connecter ou déconnecter un utilisateur à cet ordinateur à partir du trio nom d'utilisateur-mot de passe-domaine.
- **Allumer / Redémarrer / Eteindre** : Permet de mettre en marche, redémarrer ou arrêter cet ordinateur à distance. La mise en marche n'est possible que si l'adresse IP et adresse MAC sont enregistrées, si la carte réseau est compatible et si l'option WakeOnLan est activée dans le BIOS de l'ordinateur.
- **Exécuter des commandes** : Cette fonction correspond à la fonction « Exécuter » du menu démarrer de Windows et permet donc de lancer une installation, une application ou un fichier .bat à distance sur ce poste.

I.2.3. Les inconvénients de iTALC :

- Pour que l'application puisse démarre il faut désactiver le pare-feu de Windows.
- Un problème d'affichage des accents.
- Le dossier d'enregistrement par défaut du fichier de configuration « *globalconfig.XML* » est *C:\Document And setting\, donc si un autre utilisateur se connecte sur le poste il lui faudra créer un nouveau fichier de configuration.*
- Le logiciel a de grandes qualités et de nombreuses fonctions, mais il ne possède pas encore de programme de désinstallation.
- Le nombre de postes est limité par 12 postes au maximum.
- L'installation de ce logiciel exige un informaticien puisque il y a la notion du clé privé et clé public qui peut guider à une mal installation et la possibilité d'avoir 2 poste maîtres à la même salle.

II. LanSchool :

II.1. Présentation de LanSchool :

LanSchool est un outil logiciel qui retire des distractions en donnant à des professeurs l'option pour verrouiller à l'extérieur le contrôle les ordinateurs des étudiants. Un module d'instructeur monté sur ordinateur de l'enseignant et le module de stagiaire étant monté sur chaque ordinateur dans le laboratoire(classe), LanSchool emploie le réseau en mode de communication de TCP/IP pour effectuer les ordinateurs de laboratoires et salles en environnement d'enseignement optimal. Son émission, surveillance, et fonctionnements à télécommande amènent les étudiants et l'enseignant plus étroitement ensemble.

LanSchool augmente l'efficacité d'apprendre en éliminant les distractions et permet l'interaction croissante de étudiant avec l'ordinateur..

L'avantage immédiat de LanSchool était de donner les instructeurs le contrôle absolu de chaque ordinateur dans le laboratoire. Donc l'enseignant pendant un cours, contrôle ce qui est sur chaque moniteur. Si il explique une application sur son ordinateur du front du cours, chaque étudiant voit chaque cliquetis de souris et frappe sur son propre ordinateur. Si il a besoin de l'attention sur lui même, il peut noircir à l'extérieur chaque écran. LanSchool est un outil d'enseignement de valeur inestimable.

II.2. Les Fonctions proposées par LanSchool :

Console LanSchool

- Accès facilité à la console grâce au menu barre d'état système.
- Gestion des élèves par un affichage sous forme de liste ou de vignettes.
- Accès aux fonctionnalités par les menus, les icônes ou le clic droit.
- Gestion de tous les élèves ou d'une sélection d'élèves.

Diffuser l'écran de l'enseignant sur les écrans des élèves

- Affiche l'écran de l'enseignant sur les écrans des élèves.
- En mode plein écran, l'écran de l'enseignant s'affiche en plein écran sur celui de l'élève. La souris et le clavier de ce dernier sont verrouillés.
- En mode fenêtre, le contenu de l'écran de l'enseignant s'affiche sur celui de l'élève dans une fenêtre redimensionnable.
- L'enseignant dessine sur l'écran pour souligner les détails importants.

Diffuser l'écran d'un élève sur les écrans des autres élèves

- Permet à un élève de montrer son écran aux autres élèves de la classe.
- L'élève contrôle la diffusion, mais l'arrêt de cette diffusion est contrôlé par l'enseignant.

Réinitialiser les écrans des élèves

- Permet de capter à nouveau l'attention des élèves en réinitialisant leurs écrans et en verrouillant leurs claviers et leurs souris.
- Tout en réinitialisant les écrans, l'enseignant peut diffuser un message personnalisé à la classe.

Commande à distance

- Permet de venir en aide à un élève en prenant le contrôle de son écran, de son clavier et de sa souris.
- Durant une session de commande à distance, une option existe qui permet de désactiver le clavier et la souris de l'élève.

Envoyer un message

- Envoyer un message de texte à tous les élèves ou à une sélection d'élèves.

Limitations concernant l'accès à Internet

- Bloque toutes les fonctions liées à Internet : navigateurs, courrier électronique et messageries instantanées.
- Permet de bloquer ou d'autoriser l'accès à certains sites Internet.
- Permet d'enregistrer puis de charger les paramètres concernant Internet pour les différentes classes.

Limitations concernant les applications

- Permet de n'autoriser que les applications nécessaires aux élèves.
- Permet de bloquer le fonctionnement de certaines applications.
- Permet d'enregistrer puis de charger les paramètres concernant les applications pour les différentes classes.

Vote des élèves

- Permet de poser aux élèves des questions à réponse oui/non ou à choix multiples.
- Permet de voir le nombre d'élèves ayant voté.
- Permet de suivre les résultats en temps réel.

Question émanant d'un élève / chat

- Les élèves souhaitant poser une question n'ont qu'à cliquer sur l'icône LanSchool pour en informer l'enseignant.
- L'objet de la question peut alors être éclairci par l'enseignant.
- L'enseignant peut initier une session de discussion par chat avec un élève.

Envoyer et collecter des fichiers

- Permet d'envoyer des fichiers vers l'ordinateur des élèves ou sur le lecteur réseau.
- Permet de collecter des fichiers depuis les ordinateurs des élèves.

Applications en cours de fonctionnement

- L'enseignant visualise toutes les applications fonctionnant sur l'ordinateur d'un élève ; il peut alors les ajouter à ses listes d'applications autorisées ou bloquées.

Capture d'écran

- Permet de réaliser une sauvegarde de l'écran d'un élève.

II.3. Les inconvénients de Lanschool :

- Que le maître qui peut commencer le chat avec les étudiant et les conversations se réalisent seulement au niveau du professeur.
- L'étudiant ne peut pas poser des questions au professeur via des messages.
- Les deux individus ne peuvent pas communiquer entre eux par l'envoi des fichiers.
- Lanschool voit toutes les salles comme une seule salle, et contrôle tous les ordinateurs même si celles qui font pas partie de la séance.
- L'absence de la prise en charge des groupes séparément et la sauvegarde des travaux pour chaque groupe.
- La seule moyenne pour désinstaller l'application est d'exécuter *setup.exe* (qui contient le module de suppression), si on perd *setup.exe* on pourra pas supprimer l'application.

Conclusion: Les deux applications qui nous avons décrits dans ce chapitre nous ont aider à mieux cerner les besoins dans le cadre d'un système d'apprentissage et de mieux comprendre la logique de leur fonctionnement, et nous avons constater qu'ils utilisent le modèle centralisé vu dans le chapitre précédent. Nous allons dans le chapitre suivant présenter les architectures client/Server et peerTopeer qui servent pour implémenter les collecticiels.

Chapitre III

Les Architectures Client/Server & PeerToPeer

Dans ce Chapitre :

- Introduction.
- Les réseaux Client/Server
- Les réseaux PeerToPeer.
- Conclusion.

I. Introduction :

Les systèmes informatiques peuvent être classés en deux grandes catégories, présentés dans la *figure 10* : les systèmes centralisés reposant sur des mainframes et les systèmes distribués. Ces derniers peuvent être construits selon deux modèles : le modèle client/serveur et le modèle pair à pair qui peut être pur ou hybride.

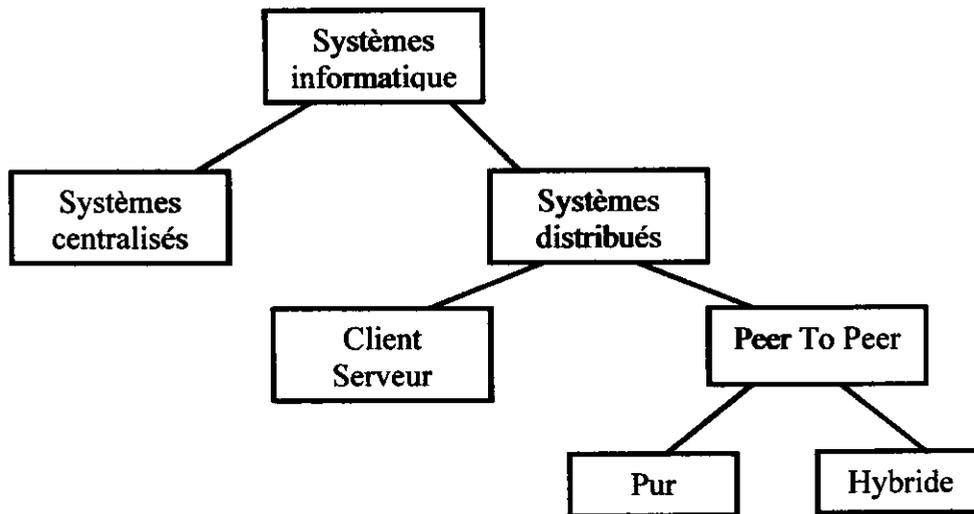


Figure 09 - Classification des systèmes informatiques

II. Les réseaux client-serveur :

Dans l'informatique moderne, de nombreuses applications fonctionnent selon un environnement client serveur; cette dénomination signifie que des machines clientes (faisant partie du réseau) contactent un serveur - une machine généralement très puissante en termes de capacités d'entrées\sorties - qui leur fournit des services. Nous allons voir comment cette technologie permet d'exploiter au mieux les réseaux, et permet un haut niveau de coopération entre différentes machines sans que l'utilisateur se préoccupe des détails de compatibilité.

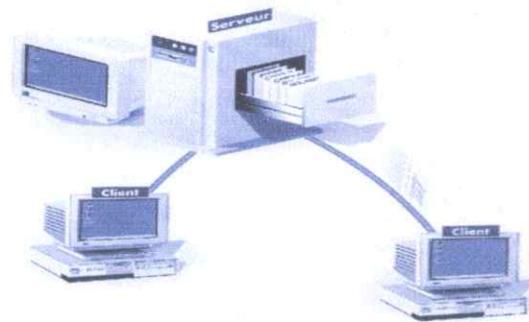


Figure 10 : architecture client/Server

Définition1 :

Les acteurs principaux d'une architecture Client-serveur sont au nombre de trois:

1. **le client** : processus demandant l'exécution d'une opération à un autre processus serveur par l'envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.
2. **le serveur** : processus accomplissant une opération sur demande d'un client et transmettant la réponse à ce client.
3. **le middleware** : ensemble des services logiciels construits au-dessus d'un protocole de transport afin de permettre l'échange de requêtes et des réponses associées entre client et serveur de manière transparente.

Fonctions d'un middleware :

- procédure d'établissement de connexion
- exécution des requêtes
- récupération des résultats
- procédure de fermeture de connexion
- initiation des processus sur différents sites
- services de répertoire (nommage)
- accès aux données à distance
- gestion des accès concurrents
- sécurité et intégrité
- monitoring
- terminaison des processus
- mise en cache des résultats et des requêtes.

Définition2 :

Pour définir la technologie Client-Serveur, il faut ajouter à cela un ensemble de propriétés :

- le **service** : c'est le travail fourni par le serveur suite à la requête du client. Le client est donc consommateur et le serveur fournisseur de services.
- les **ressources partagées** : le serveur est capable de servir de nombreux clients simultanément et réguler leur accès.
- les **protocoles asymétriques** : de nombreux clients demandent un service à un serveur qui attend leurs requêtes.
- la **transparence** : le serveur peut résider ou non sur la même machine que le client, sans différence pour ce dernier.
- la **compatibilité** : l'application Client ou Serveur est indépendante du matériel.
- le **faible couplage** : le client et le serveur communiquent uniquement par envoi de messages
- l'**encapsulation des services** : le serveur choisit la manière dont il réalise le service demandé, le client se borne à définir ce qu'il désire obtenir. Ainsi l'implémentation du serveur peut être changé sans préoccuper le client.
- l'**adaptabilité** : des machines peuvent être ajoutées ou retirées du réseau, les performances des machines peuvent aussi évoluer.

II.1. Evolution de l'architecture client/serveur :

Jusqu'à ce jour, les développements de la technologie Client-Serveur se sont produits en trois grandes étapes:

- **la première vague** : celle de l'apparition du partage des ressources; différents dispositifs sont alors mis en commun tel que des imprimantes et des lecteurs. C'est donc principalement le début des serveurs de fichiers; le client demande des enregistrements de fichiers au serveur.
- **la deuxième vague** : celle des applications centralisées de bases de données. Le client forme des messages de requêtes pour que le serveur sélectionne dans sa base l'information demandée et la lui renvoie via le réseau. Le client reçoit donc juste ce qui l'intéresse et non un fichier complet.

– **la troisième vague** : celle des objets distribués qui regroupent toutes les possibilités de techniques antérieures en leur ajoutant la capacité de répartir au mieux les fonctions entre clients et serveurs, dans chaque cas de figure.

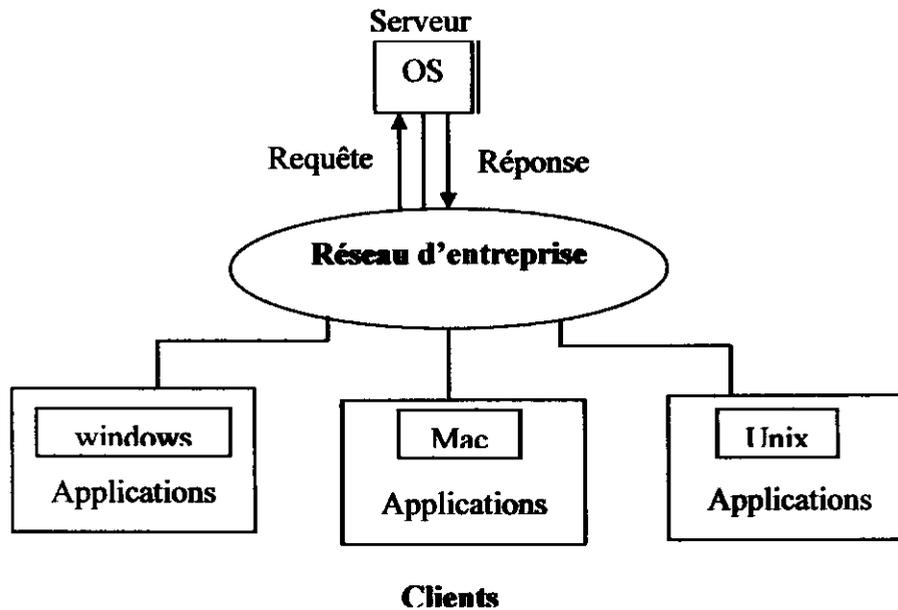


Figure 11 : Exemple d'architecture client-Serveur

II.2. Fonctionnement d'un système Client-Serveur :

Un système Client-Serveur fonctionne selon le schéma suivant:

- le client émet une requête vers le serveur grâce à son adresse et un port, qui désigne un service particulier du serveur.
- le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et son port.

Pour établir une connexion, nous aurons besoin de l'objet suivant:

Nom_de_Socket	
ENTREE	SORTIE
@Local Num_Port_Local	@Destination Num_Port_Destination

ENTREE représente l'emplacement où l'on reçoit les données.

- @locale représente l'adresse de ma machine
- Num_Port_Locale représente le port de ma machine qui est en communication

SORTIE représente l'emplacement où l'on envoie les données.

- *@destination* représente l'adresse de la machine avec laquelle on est en communication.
- *Num_Port_Destination* représente le port de la machine avec laquelle on est en communication.

Un flux est établi entre deux sockets, un chez le client, et un chez le serveur.

II.3. Avantages et Inconvénients :

Avantages:

Le modèle Client-Serveur est particulièrement recommandé pour les réseaux nécessitant un haut niveau de fiabilité, ses principaux atouts sont:

- des ressources centralisées (étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction).
- une meilleure sécurité (le nombre de points d'entrée permettant l'accès aux données est moins important).
- une administration au niveau serveur (les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés).
- un réseau évolutif (grâce à cette architecture on peut supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures).

Inconvénients :

L'architecture Client-Serveur a tout de même quelques lacunes:

- un coût élevé dû à la technicité du serveur (un serveur doit être puissant et rapide, afin de pouvoir gérer le plus rapidement possible les requêtes d'un grand nombre de processus)
- un maillon faible (le serveur est le seul maillon faible du réseau Client-Serveur, étant donné que tout le réseau est architecturé autour de lui, heureusement, il a une grande tolérance aux pannes).

III. Les Réseaux Peer To Peer :

L'expression "réseau peer to peer" (P2P), que l'on traduit généralement par réseau de "poste à poste", "pair à pair" ou encore "d'égal à égal" désigne une architecture de réseau où les postes connectés communiquent directement entre eux et partagent leurs ressources. Tous les postes ont un rôle équivalent, à la fois client et serveur par rapport à ces ressources (espace de stockage, puissance de calcul...), d'où leur appellation de "servent" (contraction de serveur et client). C'est la mise en commun de ces ressources qui fait la force et le succès de ces réseaux.

Ce type d'architecture qui est à la base d'Internet dès l'origine, a été largement médiatisé ces dernières années, suite aux péripéties judiciaires liées aux partages de fichiers protégés par un copyright.

Les applications vont du partage de films ou de fichiers musicaux pour le grand public au travail collaboratif ou au calcul distribué pour l'entreprise. On estime que le trafic p2p représente 60% du trafic des réseaux haut débit le jour et 90% la nuit preuve de sa popularité.

Objectifs :

Le modèle P2P étant très général, des applications de nature très différente peuvent l'utiliser. En effet, les objectifs sont variés:

- Partage et réduction des coûts entre les différents peers.
- Fiabilité et passage à l'échelle, l'absence d'élément centralisé pour l'échange des données permet d'accroître la fiabilité en supprimant tout point central de panne et d'améliorer le passage à l'échelle en évitant les goulots d'étranglement.
- Agrégation des ressources et interopérabilité, en mettant en commun des ressources individuelles comme de la puissance de calcul ou de l'espace de stockage.
- Accroissement de l'autonomie en l'absence d'une autorité centrale, il est de la responsabilité de chacun de partager ou non des fichiers.
- Anonymat pouvant être assuré par certaines applications, en utilisant par exemple des algorithmes de routage qui rendent quasiment impossible le pistage d'une requête.
- Communication AD-HOC et collaborative.

III.1. Caractéristiques des systèmes P2P :

Les principales caractéristiques des différents protocoles sont:

- la localisation des fichiers dans un environnement distribué ;
- des index du réseau P2P ;
- la libre circulation des fichiers entre systèmes ;
- un mode de communication standard (TCP et http) ;
- des capacités de connexion variables suivant les modèles ;
- des peers non sûrs ;
- aucun peer n'a une vue globale du système.

III.2. Architectures de réseaux peer to peer :

III.2.1. Architecture décentralisée (pur) :

Nous avons vu que l'architecture centralisée pose des problèmes de sécurité, robustesse, et de limitation de la bande passante. Les problèmes sont directement issus de l'utilisation de serveurs dont le seul but est de posséder l'annuaire des clients. Si on désire supprimer les serveurs centraux, il faut donc trouver le moyen de constituer un annuaire sur chaque client, puis de les faire communiquer. C'est sur ces mécanismes que sont basés les réseaux Peer to Peer décentralisés. Il n'y a donc plus de serveurs centraux, ce sont tous les éléments du réseau qui vont jouer ce rôle. Chaque machine dans ses rôles est identique à une autre, c'est pour cela que l'on appelle ces types de réseaux **pur peer to peer** voir (figure 12).

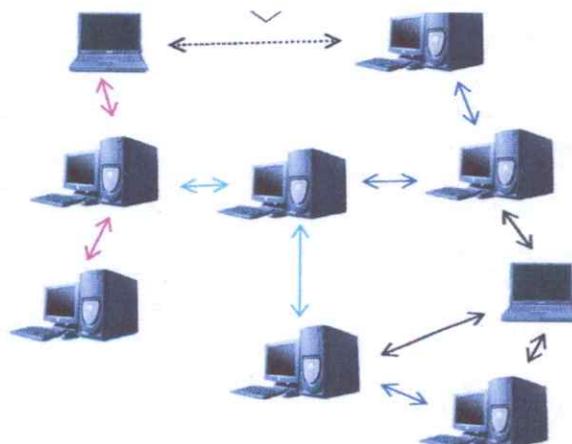


Figure 12 : – L'architecture décentralisée

Avantages :

- Administration simple et mutualisée.
- Topologie évolutive (taille illimitée en théorie).
- Disponibilité du réseau, on ne peut l'arrêter.
- S'adapte bien à la dynamique du réseau (allées et venus des pairs).
- Total anonymat en théorie.

Inconvénients :

- Gros consommateur de bande passante à cause des broadcasts envoyés.
- Pas de garantie de succès, ni d'estimation de la durée des requêtes.

III.2.2. Architectures hybrides :

Il s'agit ici d'un modèle que l'on pourrait qualifier d'hybride entre le modèle centralisé et pur. En effet, dans ce type de réseau il existe des machines particulières jouant le rôle de serveur local appelé "super node" ou "super peer" gérant un groupe de postes.

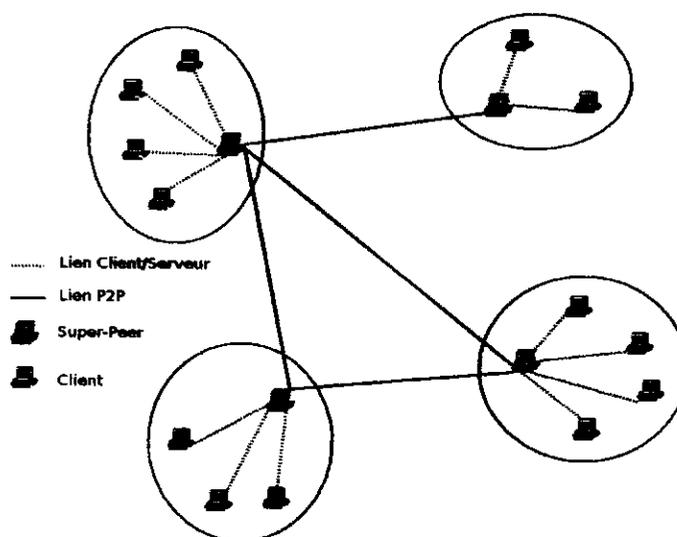


Figure 13 : – L'architecture hybride

1. Avantages :

- Indexation centralisée des ressources au niveau d'un groupe par les super nodes.
- Offre une meilleure bande passante.

2. Inconvénients :

- Complexe à mettre en place.
- La fonction de « super node » consomme environ 10 % des ressources du poste.

VI. Conclusion :

Une différence importante entre le réseau client/serveur et celle de peer-to-peer et que les clients ne partagent aucune de leurs ressources (capacité de stockage, puissance de calcul, connexion au réseau, la largeur de bande, contenu). mais Dans le concept client/serveur existe seulement une entité centrale, qui fournit tout le contenu et services qui sont offerts dans un certain réseau.

De notre point de vue, la différence la plus distinctive entre le réseau client/serveur et Le réseau Peer-to-Peer est le concept d'une entité agissant en tant que *Servent*, qui est employé dans les réseaux p2p. *Servent* est un mot artificiel qui est dérivé de la première syllabe du mot serveur ("serv") et la deuxième syllabe du mot client ("ent").

Ainsi ce terme « *Servent* » représente la capacité des noeuds d'un réseau peer to peer qui fonctionnent en même temps qu'un serveur aussi bien qu'un client. C'est complètement différent au réseau client/serveur, où les noeuds participants peuvent agir en tant que serveur ou en tant que client mais ne peuvent pas embrasser les deux capacités.

Partie II

Conception
&
Réalisation

Chapitre IV

Analyse & Conception

Dans ce Chapitre :

- Introduction.
- Un bref sur la modélisation unifié (UML)
- Détermination des cas d'utilisation
- Description détaillée des cas d'utilisation
- Description de diagramme de classe global
- Diagramme de classe global
- Architecture du système

I. INTRODUCTION :

On est bien d'accord qu'aujourd'hui, pour construire un système informatique, il est impossible d'attaquer directement à l'implantation, mais au contraire, il faut d'abord modéliser ce système. La modélisation consiste tout d'abord à écrire un problème (exemple les besoins à propos d'un système informatique à construire), puis à décrire la solution de ce problème (la conception du système à construire).

Grosso modo, l'approche objet consiste à penser à un problème, à un système en termes des concepts orientés objets, à savoir : objet et encapsulation, héritage et polymorphisme, message et interaction entre les objets. Il faut penser à l'objet non seulement lors de la programmation mais également lors de l'analyse et de la conception des systèmes informatiques, ça ce que nous allons montrer dans ce chapitre à l'appuis d'UML.

II. Un bref sur la modélisation unifié (UML):

UML (Unified Modeling Language), qui peut être traduit en français comme le langage de modélisation objet unifié, est né de la fusion des notations utilisées par une cinquantaine de méthodes objets au milieu des années 90, notamment OMT (Object Modeling Technics), Booch et OOSE (Object Oriented Software Engineering); il s'agit des trois méthodes qui ont plus influencé.

La modélisation objet, UML a été conçu comme un ensemble de notations qui visent à être utilisées dans différentes étapes de la modélisation objet, de l'expression des besoins jusqu'à l'implantation. Il s'agit des notations accessibles par différents intervenants de la modélisation objets - utilisateurs, analystes, concepteurs, développeurs des systèmes:

- Représenter des concepts abstraits (graphiquement par exemple).
- Limiter les ambiguïtés (parler un langage commun, au vocabulaire précis).
- Faciliter l'analyse (simplifier la comparaison et l'évaluation de solutions).

Construire un modèle objet en notations UML consiste à construire neuf types de diagrammes UML:

- **Diagramme de cas d'utilisation** : ils représentent les fonctions du système du point de vue de l'utilisateur.
- **Diagrammes de classes** : ils représentent la structure statique en termes de classes et de relations. Ce sont les diagrammes les plus fréquents dans une modélisation par objets.
- **Diagrammes d'objets** : ils représentent les objets et leurs relations qui sont les instances des éléments qui apparaissent dans les diagrammes de classes.
- **Diagrammes de séquence et diagrammes de collaboration** : ils présentent les objets et leurs interactions. Les diagrammes de séquence mettent l'accent sur le classement chronologique des interactions alors que les diagrammes de collaboration mettent l'accent sur l'organisation structurelle des objets qui s'interagissent.
- **Diagrammes d'état-transition** : ils représentent le comportement des objets d'une classe en terme d'états.
- **Diagrammes d'activités** : ils représentent le comportement d'une opération en termes d'actions.
- **Diagrammes de composants** : ils représentent les composants physiques d'une application.
- **Diagrammes de déploiement** : ils représentent le déploiement des composants sur les dispositifs matériels.

III. Détermination des cas d'utilisation :

Avant de déterminer les cas d'utilisation, il faut tout d'abord déterminer les acteurs qui interviennent dans le système.

Un acteur représente un rôle joué par une personne ou une chose qui interagit avec le système.

Dans notre système il existe trois (03) acteurs :

- Administrateur
- Enseignant.
- Etudiant.

Un cas d'utilisation est un comportement du système en réponse à une interaction d'un acteur, on peut associer à chaque acteur un ou plusieurs cas d'utilisations.

Dans notre étude, nous avons repéré les cas d'utilisations cités dans le tableau suivant :

Acteur	Cas d'utilisation
Administrateur	<ul style="list-style-type: none"> - Gestion des comptes d'enseignants - Gestion des comptes d'étudiants - Gestion des modules
Enseignant	<ul style="list-style-type: none"> - Créer une séance - mettre à jour les séances - lancer une séance - mettre à jour les étudiants de la séance - Préparer un TP - Préparer un Test - Envoyer un Document à l'étudiant - Envoyer un Message à l'étudiant - Voir l'écran d'un étudiant - Faire une Démonstration - limiter des applications pour les étudiants - gérer les notes des étudiants - Etablir la liste de présence - Redémarrer, Arrêter et Verrouiller des postes - contrôler le bureau d'étudiant - Gérer la boîte de réception
Etudiant	<ul style="list-style-type: none"> - Accéder à la séance en cours - voir les notes du testes - Lancer une Application - Gérer la boîte de réception - demande d'intervention - Envoyer un fichier à l'enseignant - Participer au teste

1. Diagramme de cas d'utilisation- vue de l'administrateur:

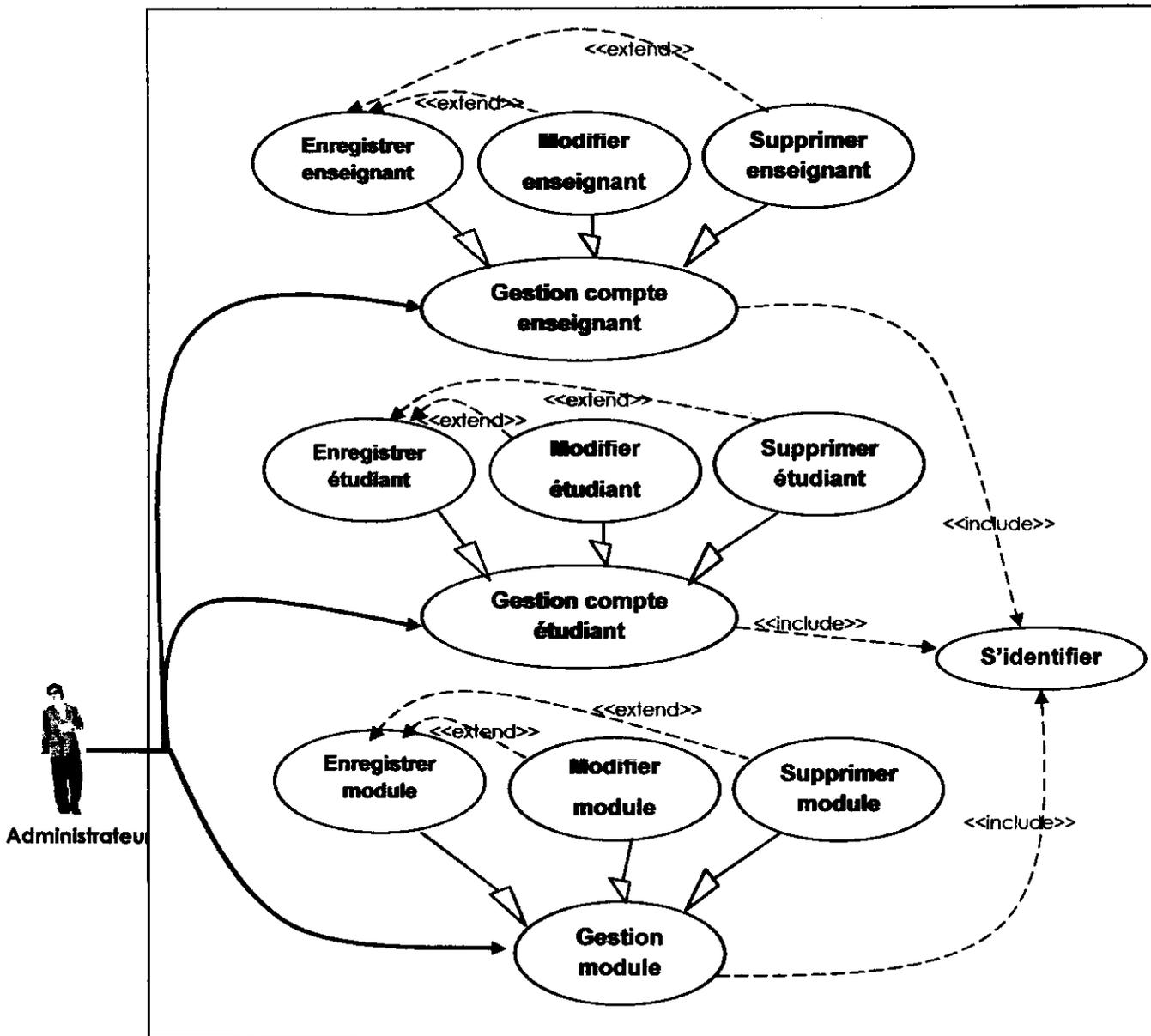


Figure 14: Diagramme de cas d'utilisation- vue de l'administrateur

2. Diagramme de cas d'utilisation- vue de l'enseignant

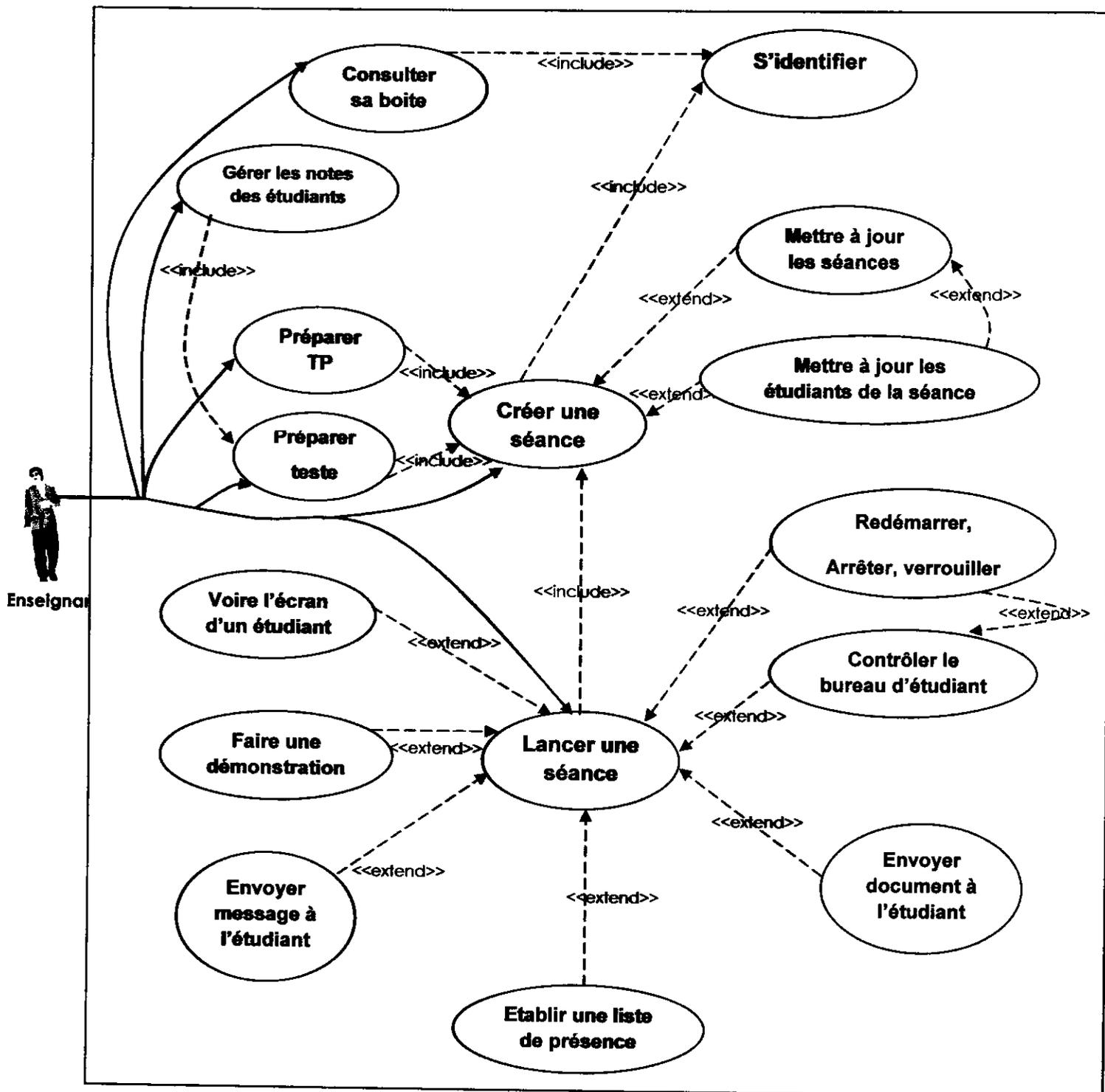


Figure 15: Diagramme de cas d'utilisation- vue de l'enseignant

3. Diagramme de cas d'utilisation- vue de l'enseignant:

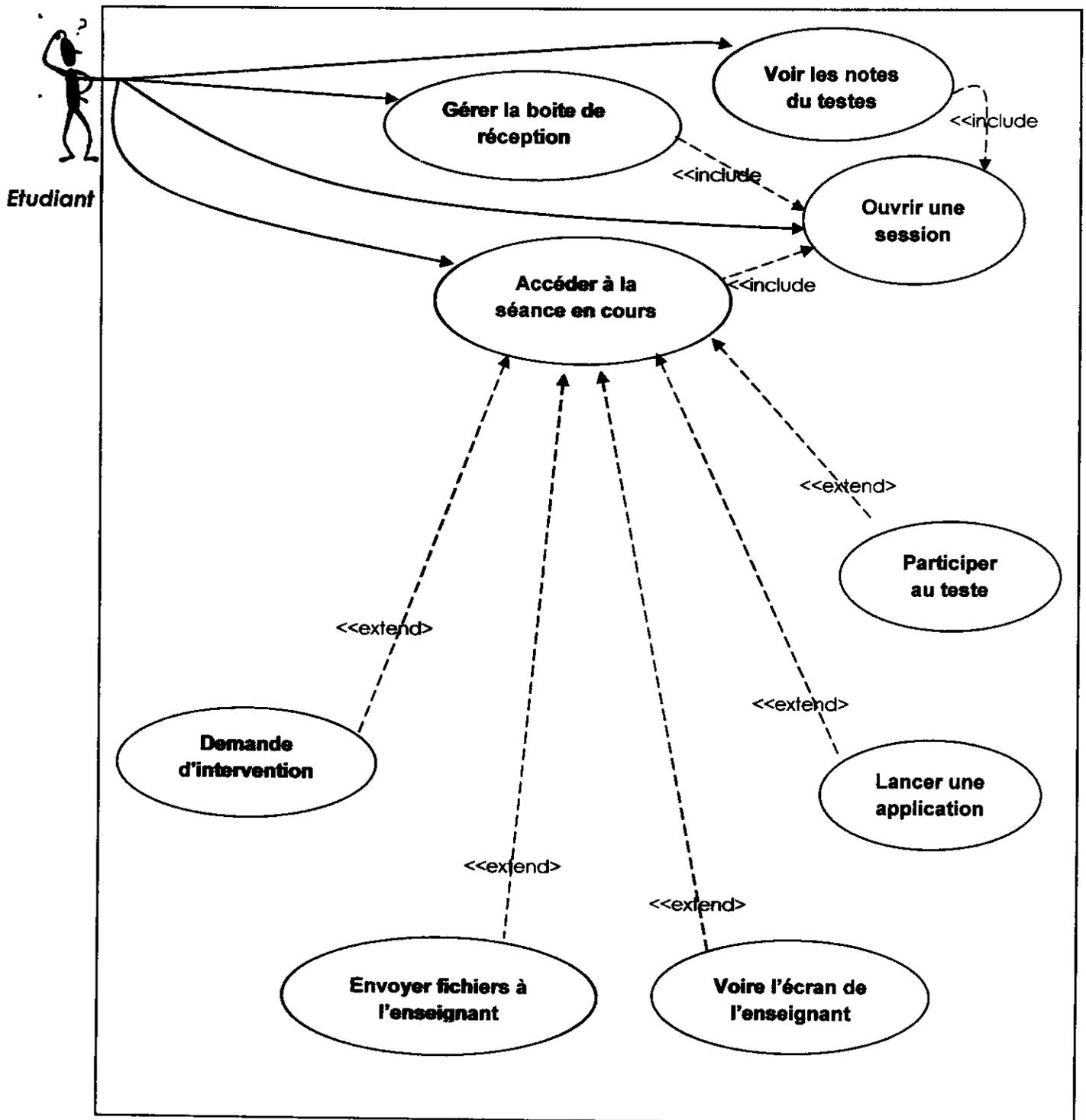


Figure 16: Diagramme de cas d'utilisation- vue de l'étudiant

VI. Description détaillée des cas d'utilisation :

❖ Administrateur :

1. Cas d'utilisation << Gestion des comptes d'enseignants >> :

1.1-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. Enregistrer un enseignant :

4. l'administrateur demande d'enregistrer un enseignant.
5. le système affiche un formulaire enseignant.
6. l'administrateur saisit les données, puis valide.
7. le système contrôle puis enregistre les données.
8. le système affiche une liste des enseignants.

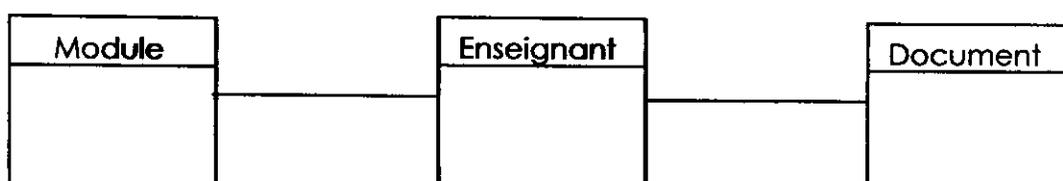
B. Modifier un enseignant :

4. l'administrateur demande de modifier un enseignant.
5. le système affiche une liste des enseignants.
6. l'administrateur sélectionne un enseignant, puis valide.
7. le système affiche un formulaire enseignant.
8. l'administrateur modifie les données, puis valide.
9. le système contrôle puis enregistre les données.

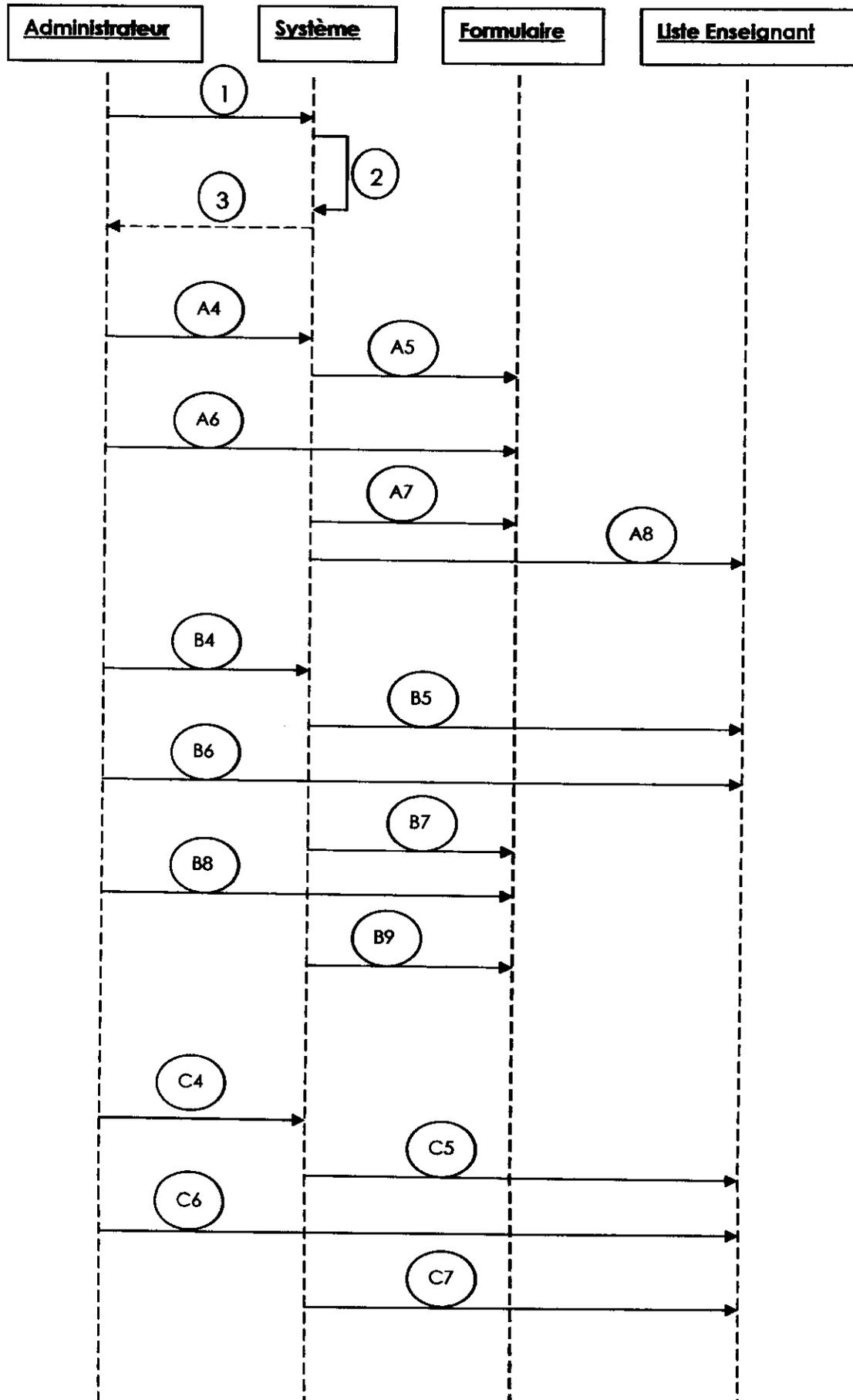
C. Supprimer un enseignant :

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

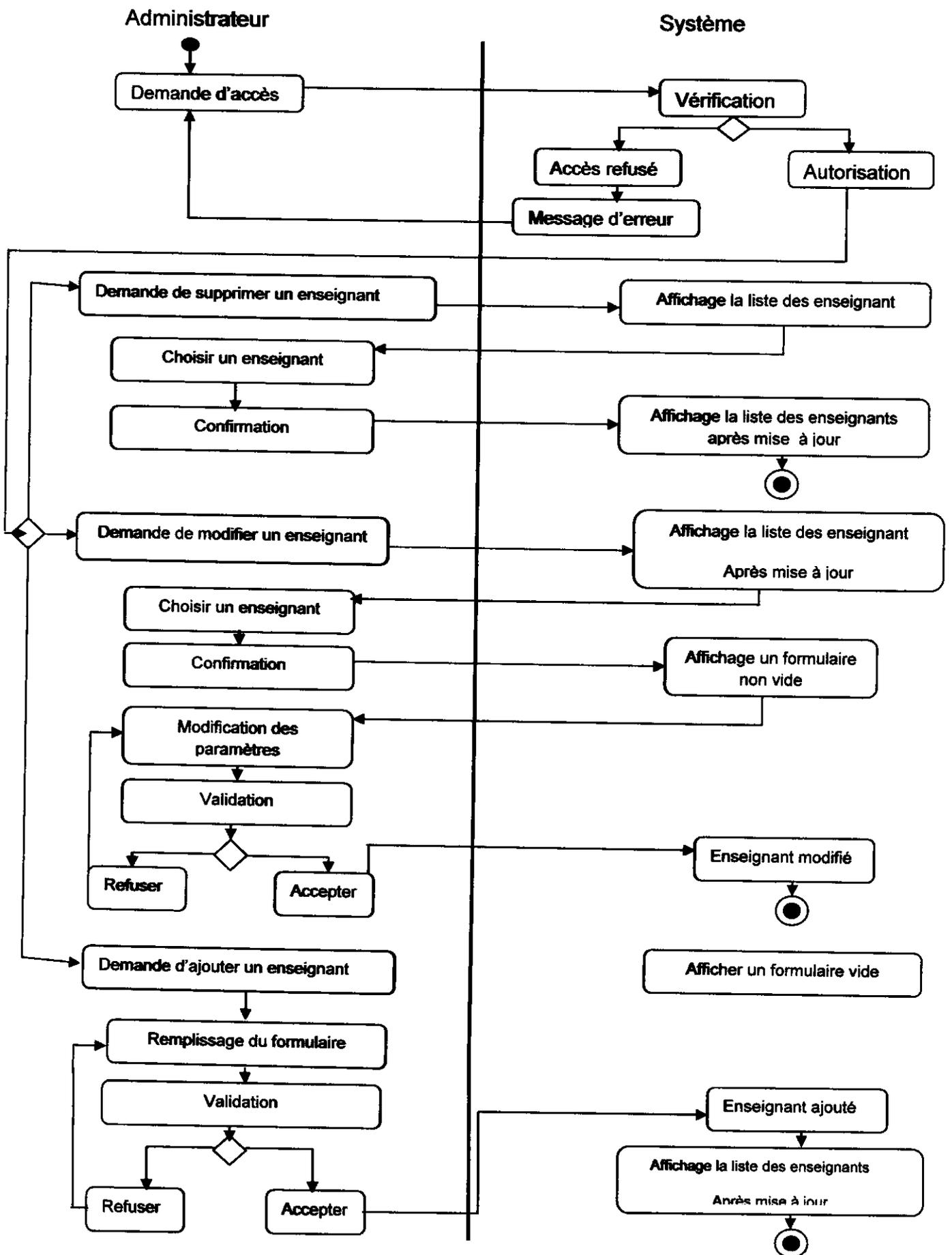
1.2-Diagramme de classe :



1.3-Diagramme de séquence :



1.4-Diagramme d'activité :



2. Cas d'utilisation << Gestion des comptes d'étudiants >> :

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. Enregistrer un étudiant :

4. l'administrateur demande d'enregistrer un étudiant.
5. le système affiche un formulaire étudiant.
6. l'administrateur saisit les données, puis valide.
7. le système contrôle puis enregistre les données.
8. le système affiche une liste des étudiants.

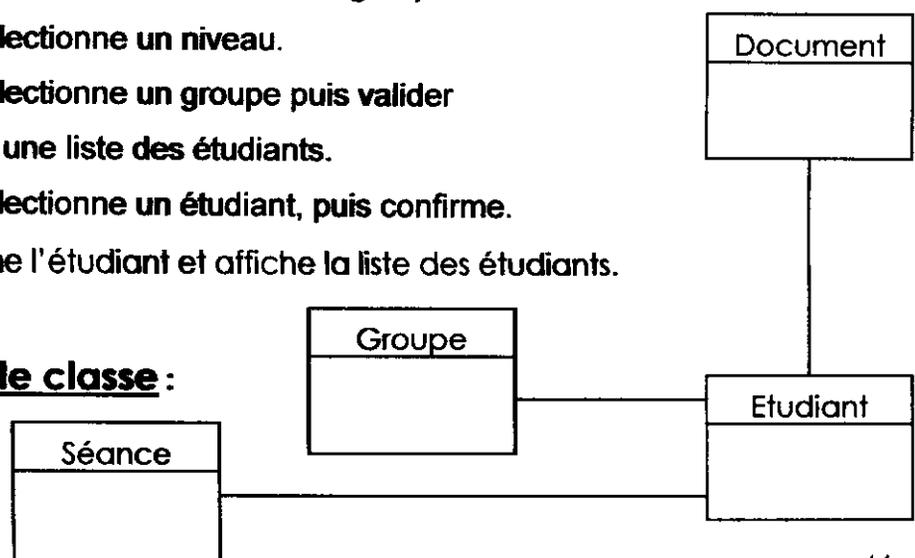
B. Modifier un étudiant :

4. l'administrateur demande de modifier un étudiant.
5. le système affiche une liste des niveaux et groupes.
6. l'administrateur sélectionne un niveau.
7. l'administrateur sélectionne un groupe puis valide.
8. le système affiche une liste des étudiants.
9. l'administrateur sélectionne un étudiant puis valide.
10. le système affiche un formulaire étudiant.
11. l'administrateur modifie les données, puis valide.
12. le système contrôle puis enregistre les données.

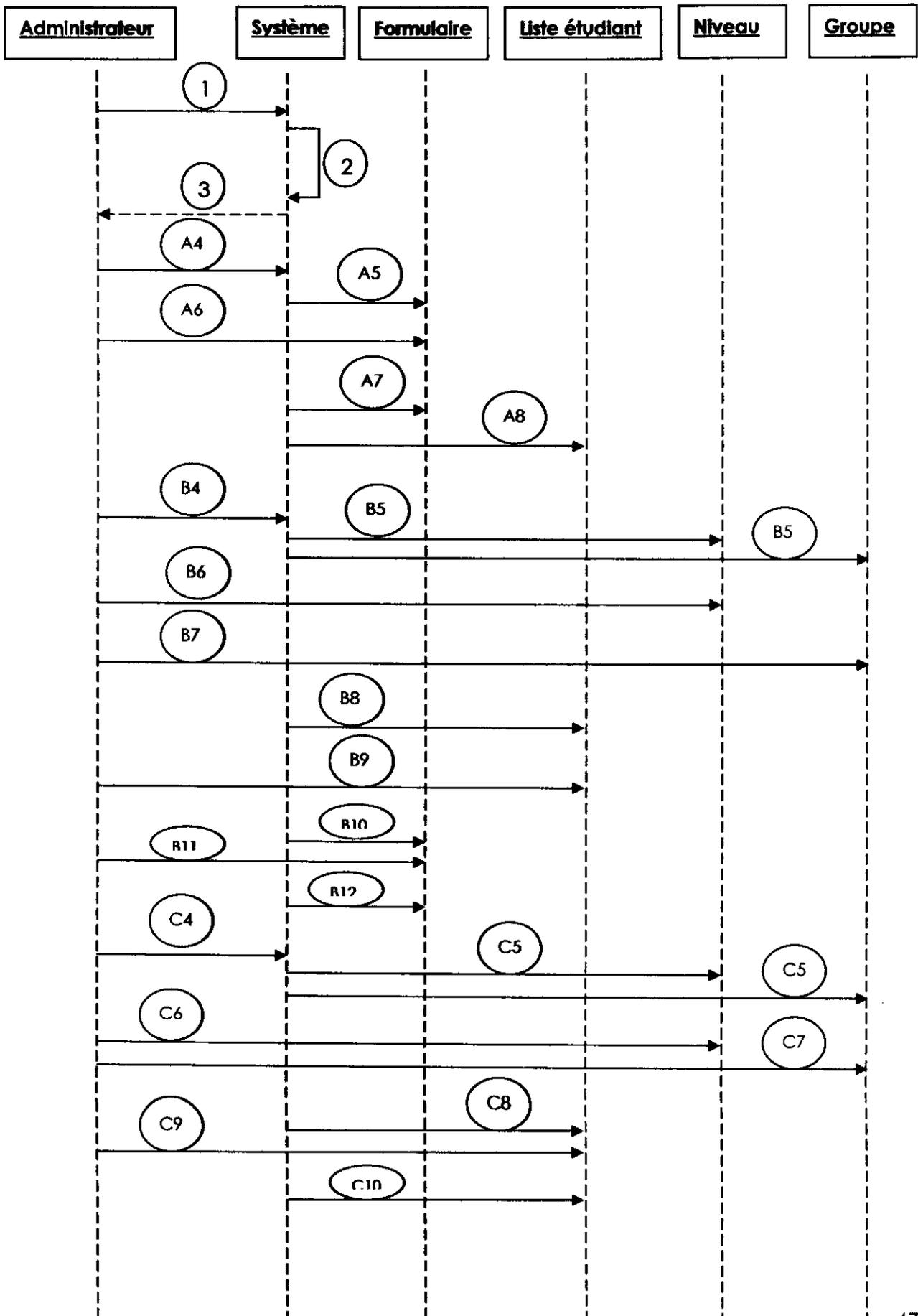
C. Supprimer un étudiant :

4. l'administrateur demande de supprimer un étudiant.
5. le système affiche une liste des niveaux et groupes.
6. l'administrateur sélectionne un niveau.
7. l'administrateur sélectionne un groupe puis valide.
8. le système affiche une liste des étudiants.
9. l'administrateur sélectionne un étudiant, puis confirme.
10. le système supprime l'étudiant et affiche la liste des étudiants.

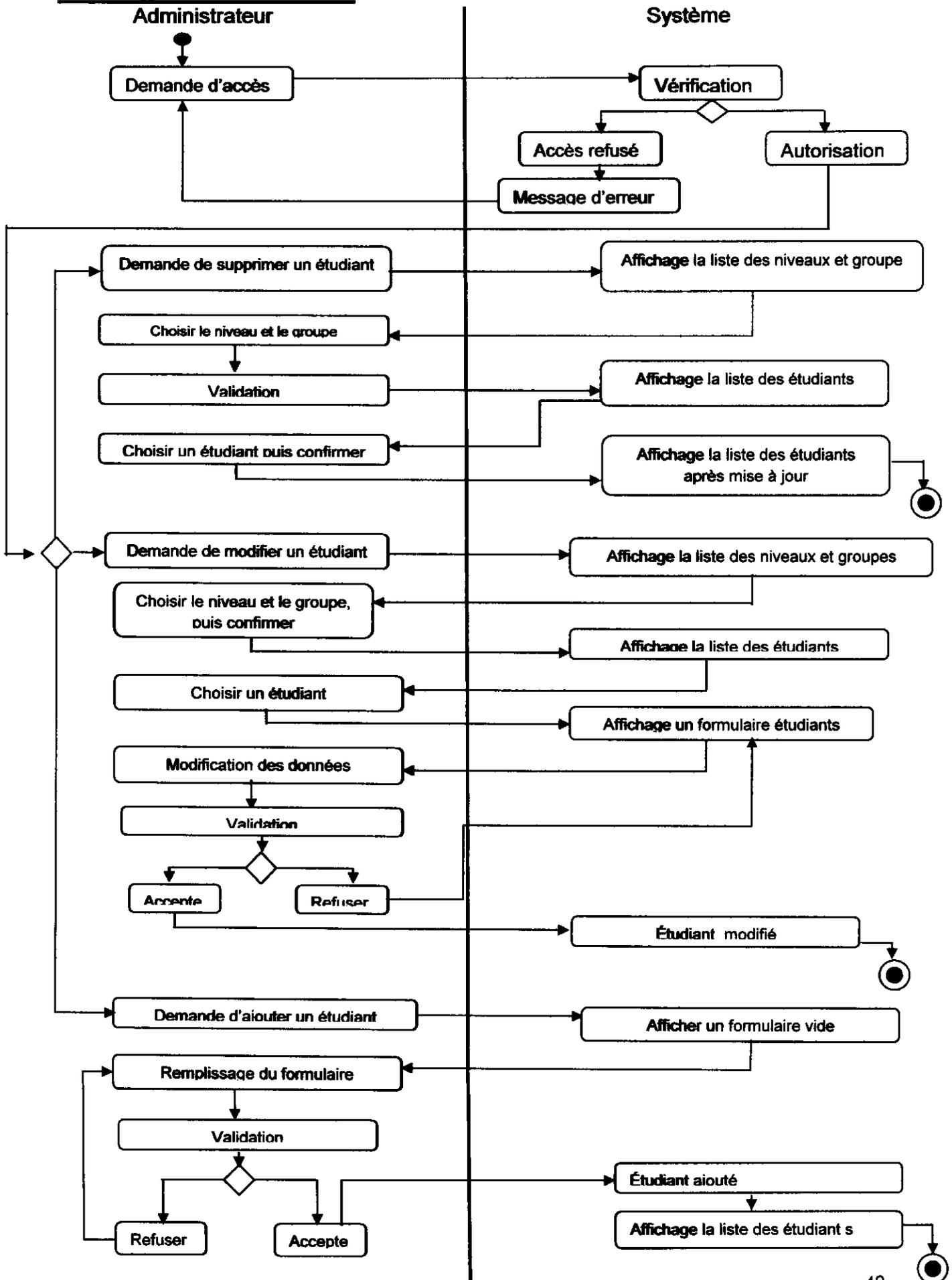
2.2-Diagramme de classe :



2.3-Diagramme de séquence :



2.4-Diagramme d'activité :



3. Cas d'utilisation << Gestion des modules >> :

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. Enregistrer un module :

4. l'administrateur demande d'enregistrer un module.
5. le système affiche un formulaire module.
6. l'administrateur saisit les données, puis valide.
7. le système contrôle puis enregistre les données.
8. le système affiche une liste des modules.

B. Modifier un module :

4. l'administrateur demande de modifier un module.
5. le système affiche une liste des modules.
6. l'administrateur sélectionne un module, puis valide.
7. le système affiche un formulaire module.
8. l'administrateur modifie les données, puis valide.
9. le système contrôle puis enregistre les données.

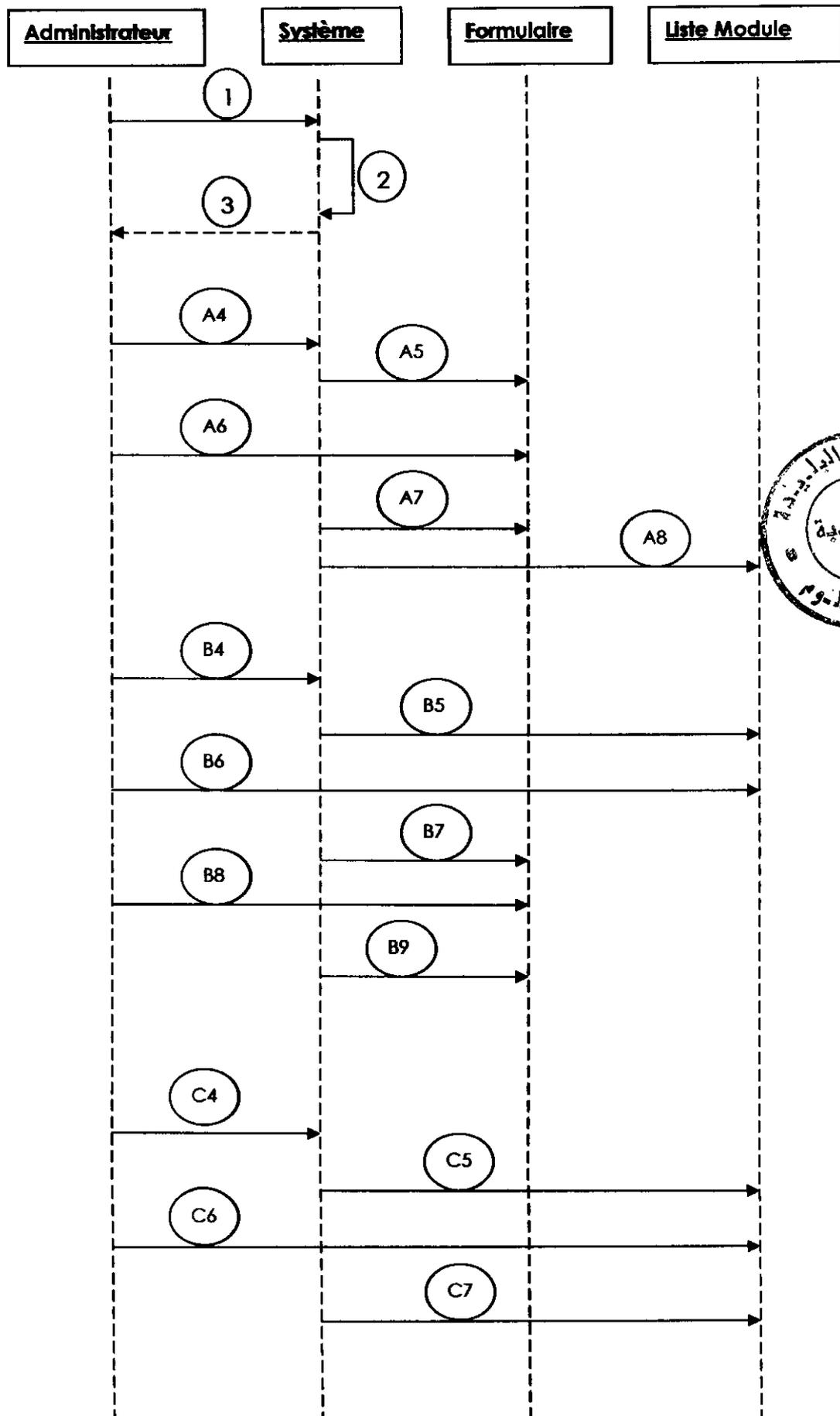
C. Supprimer un module :

4. l'administrateur demande de supprimer un module.
5. le système affiche une liste des modules.
6. l'administrateur sélectionne un module, puis confirme.
7. le système supprime le module et affiche la liste des modules.

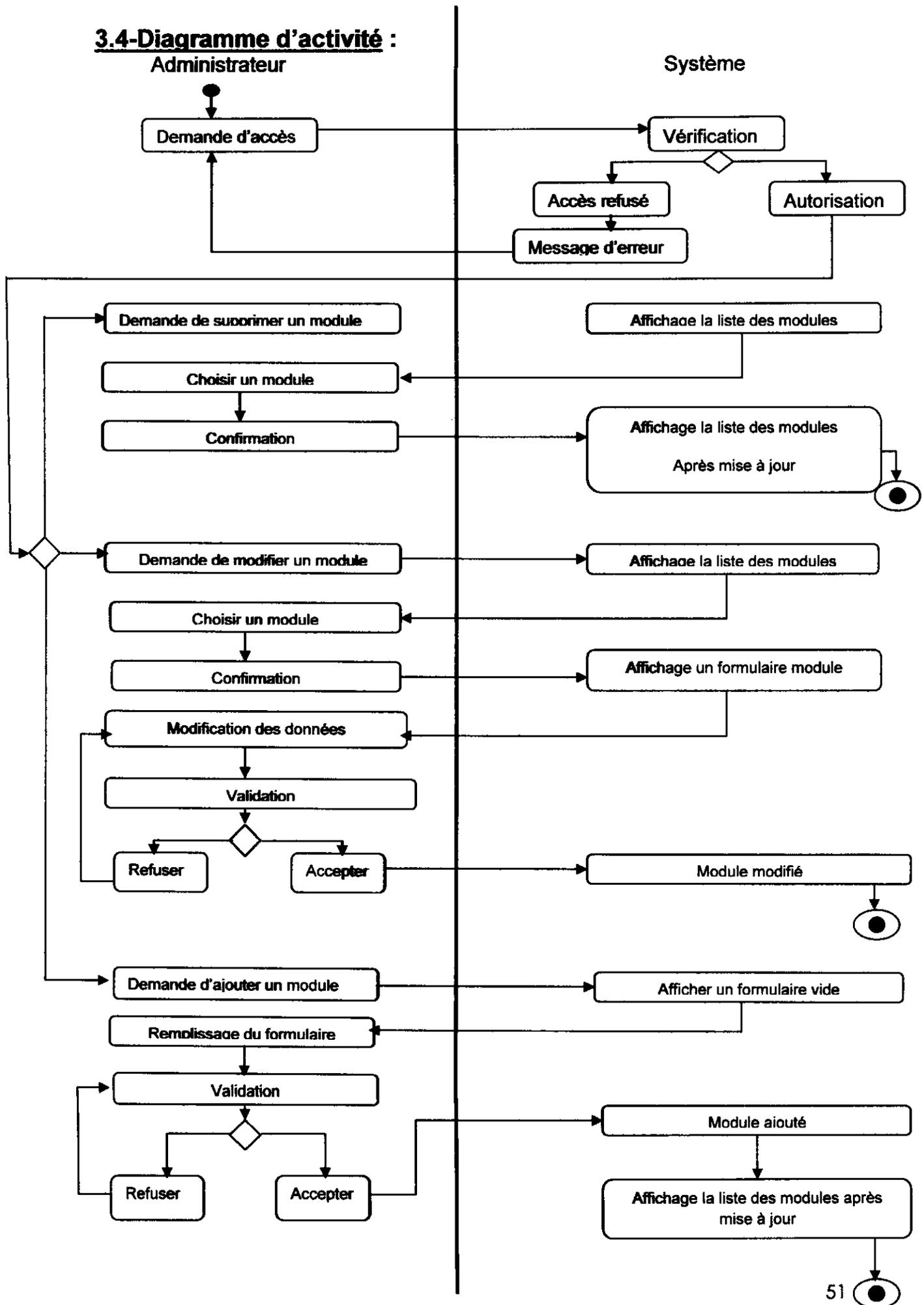
3.2-Diagramme de classe :



3.3-Diagramme de séquence :

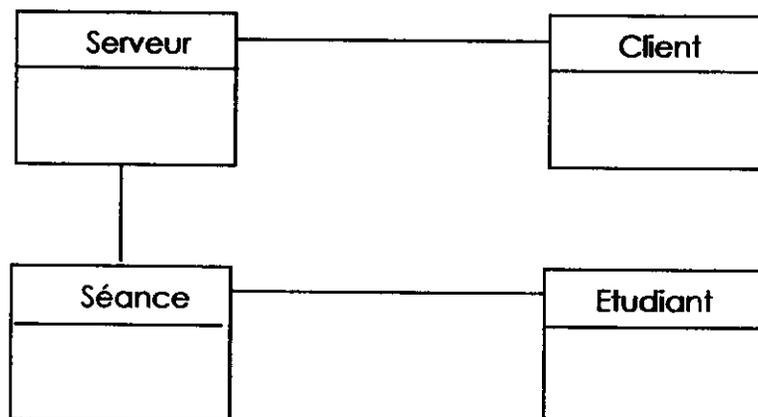
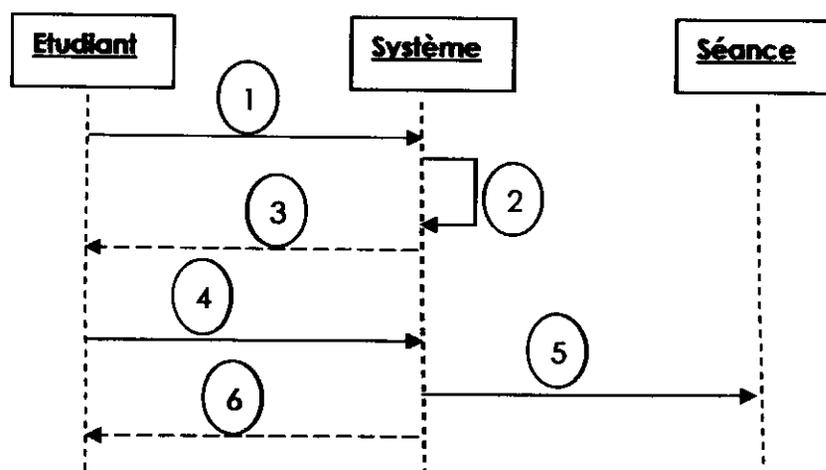


3.4-Diagramme d'activité :

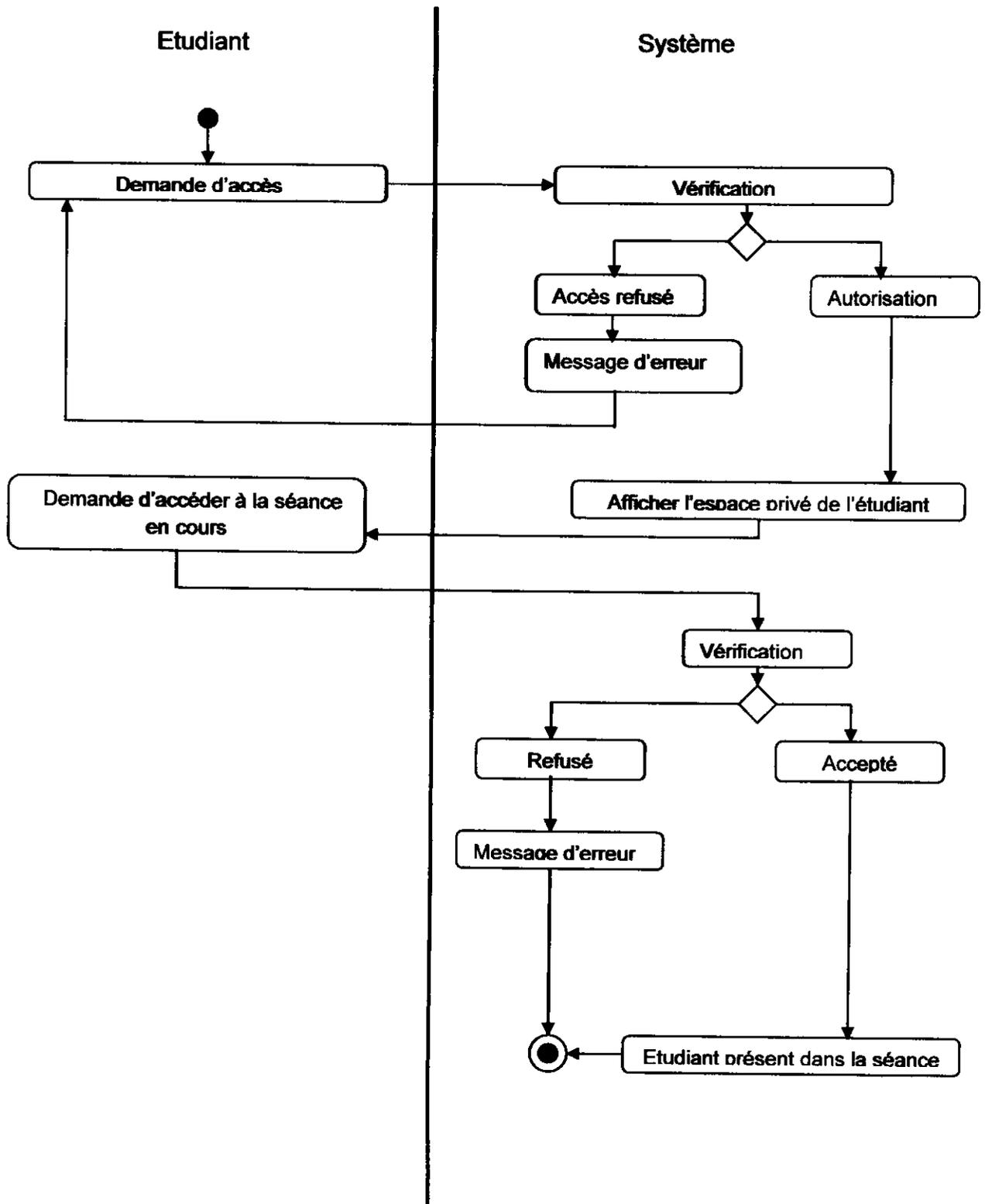


❖ Etudiant :**1. Cas d'utilisation << Accéder à la séance en cours >> :****1.1-Scénario :**

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 l'accès à la séance en cours.
5. le système vérifie l'accès à la séance.
6. le système donne l'autorisation à l'étudiant.

1.2-Diagramme de classe :**1.3-Diagramme de Séquence :**

1.4-Diagramme d'activité :



2. Cas d'utilisation << voir les notes du testes >> :

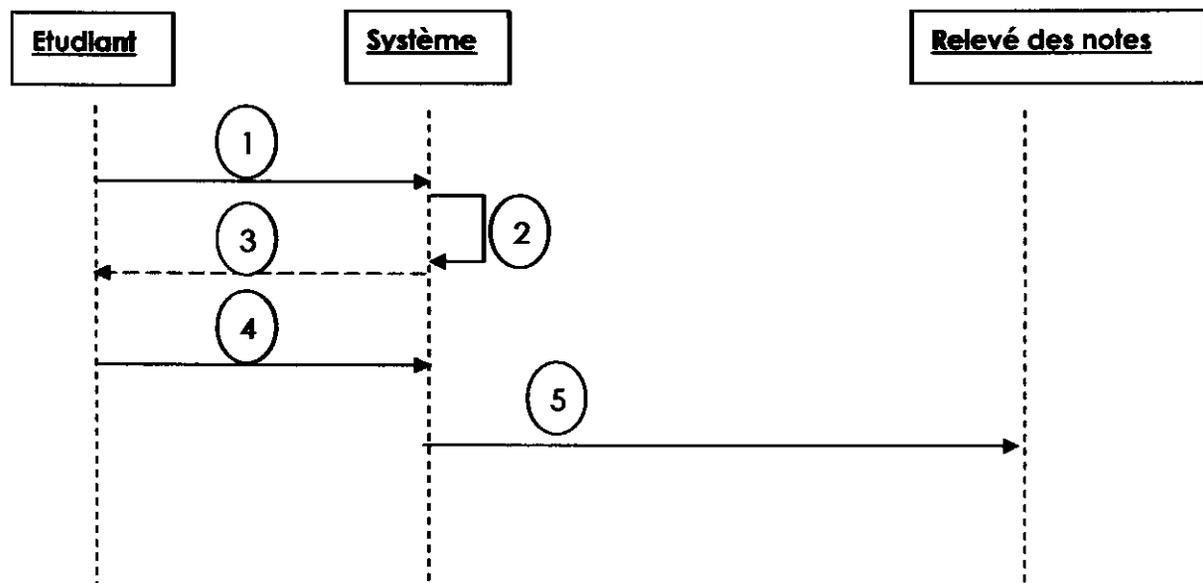
2.1-Scénario :

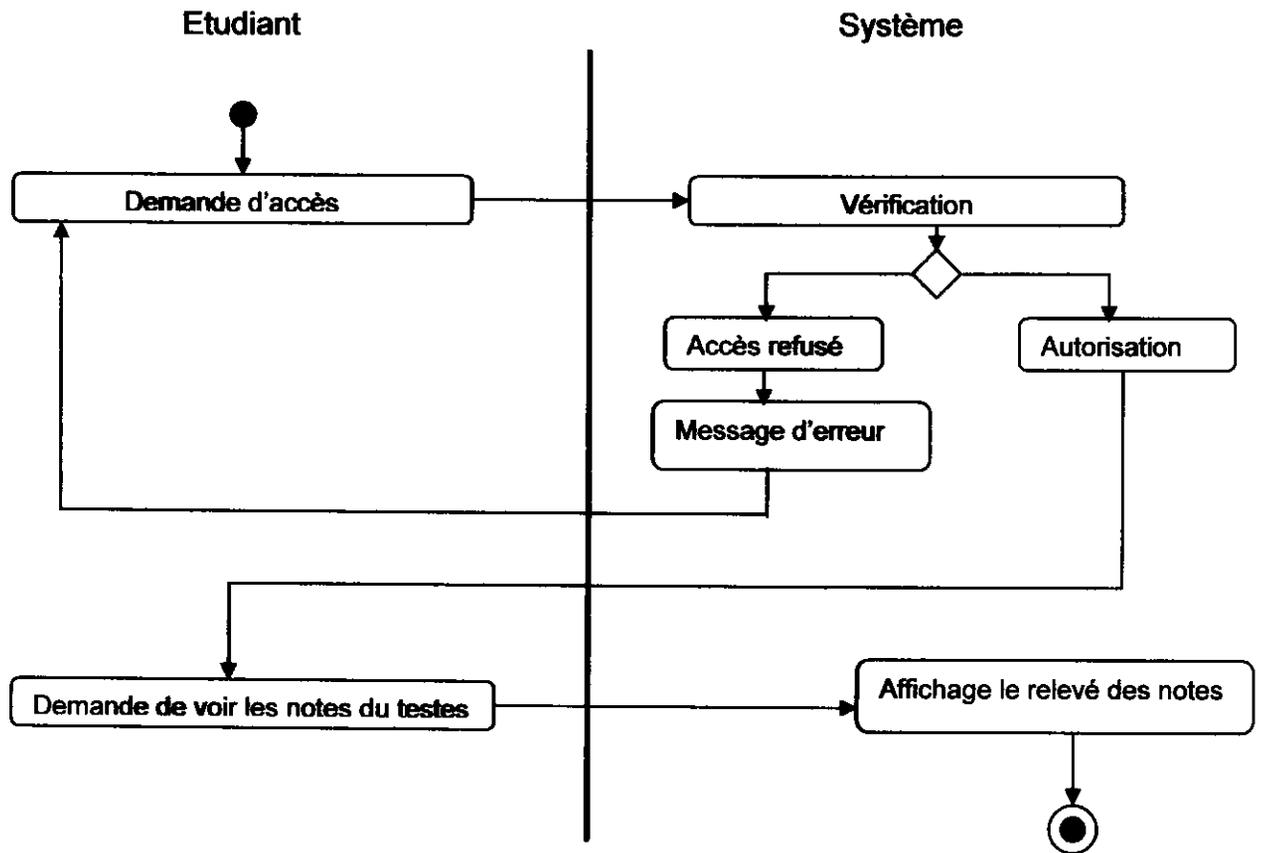
- 1- l'étudiant envoie ses paramètres d'identification
- 2- le système vérifie cette identité
- 3- le système donne l'accès à l'étudiant
- 4- l'étudiant demande de voir ses notes des tests
- 5- le système affiche le relevé des notes du module en cours

2.2-Diagramme de classe :



2.3-Diagramme de Séquence :



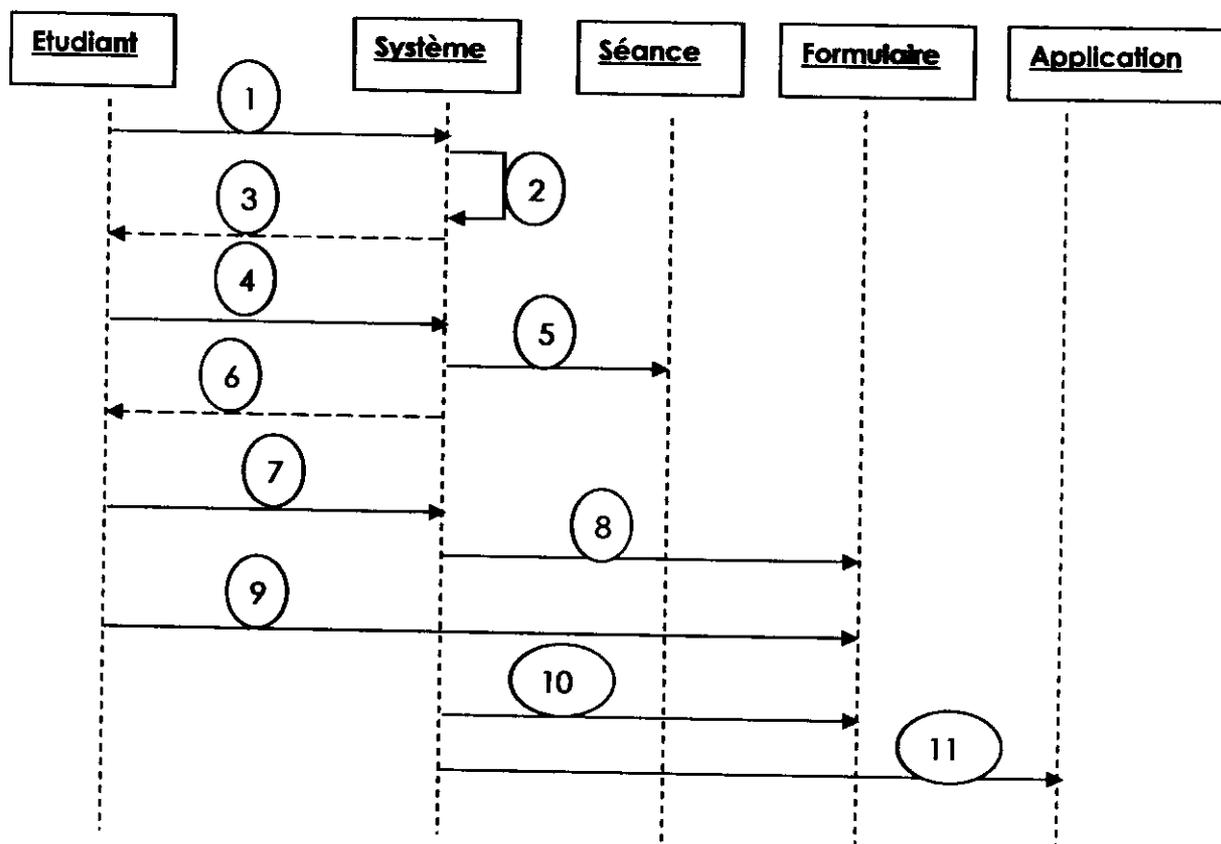
2.4-Diagramme d'activité :**3. Cas d'utilisation << Lancer une Application >> :****3.1-Scénario :**

- 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 l'accès à la séance
- 5- le système vérifie l'accès à la séance
- 6- le système donne l'autorisation à l'étudiant
- 7- l'étudiant demande d'exécuter une application
- 8- le système affiche un formulaire vide
- 9- l'étudiant saisit le nom de l'application puis valide
- 10- le système vérifie le droit de lancement de cette application
- 11- le système exécute l'application

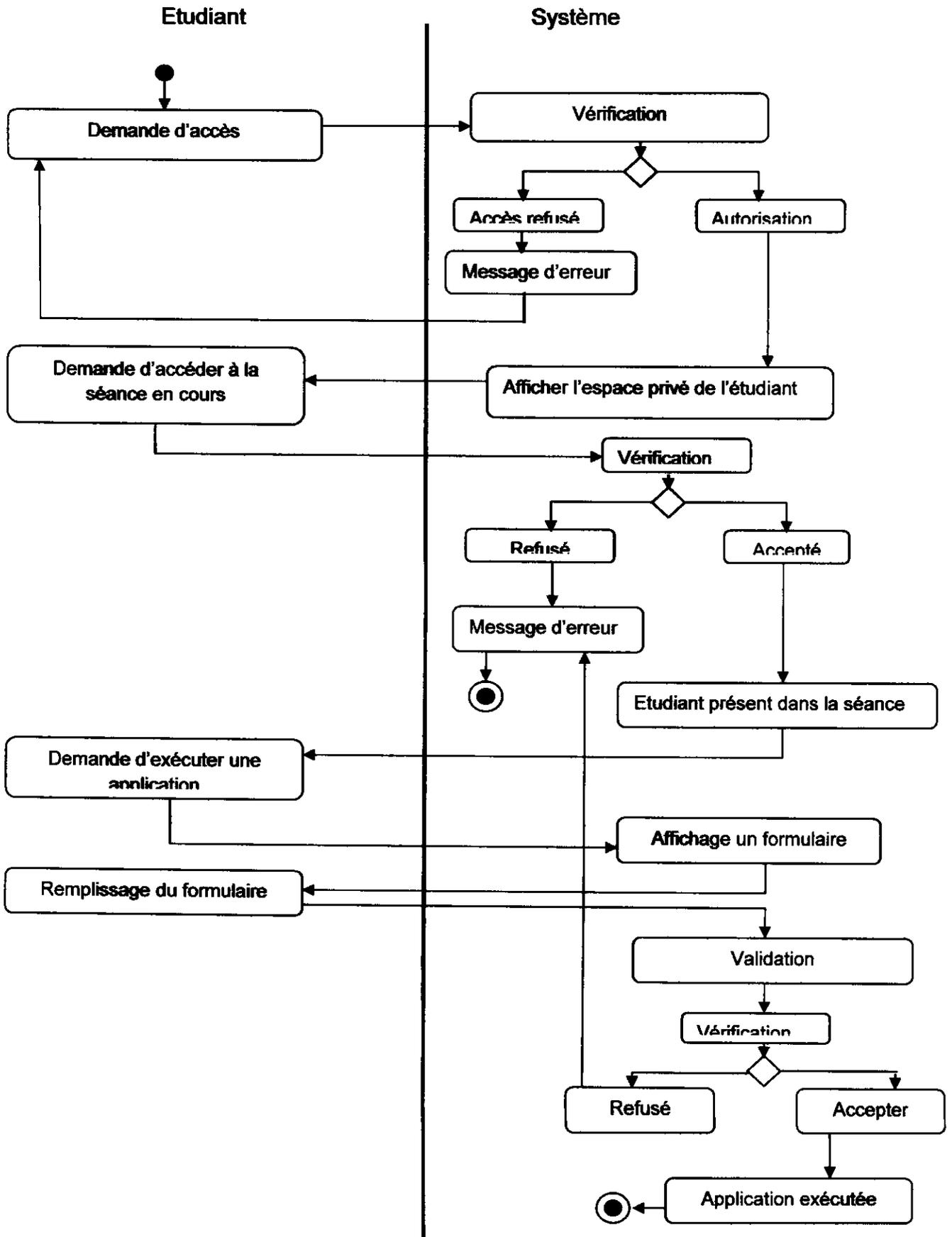
3.2-Diagramme de classe :



3.3-Diagramme de Séquence :



3.4-Diagramme d'activité :



4. Cas d'utilisation <<Gérer la boîte de réception >> :

4.1-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 l'accès à la séance
- 5- le système vérifie l'accès à la séance
- 6- le système donne l'autorisation à l'étudiant

A .Consultation :

- 7- l'étudiant demande de consulter sa boîte de réception
- 8- le système affiche la liste des fichiers concernant le module en cours

a. Voir un fichier :

9. L'étudiant sélectionne un fichier puis valider
10. le système vérifie et ouvre le fichier

b. Copier un fichier :

9. L'étudiant sélectionne un fichier puis valider
10. le système affiche un formulaire du chemin de destination
11. l'étudiant définit le chemin puis valider
12. le système contrôle le chemin et transfère le fichier

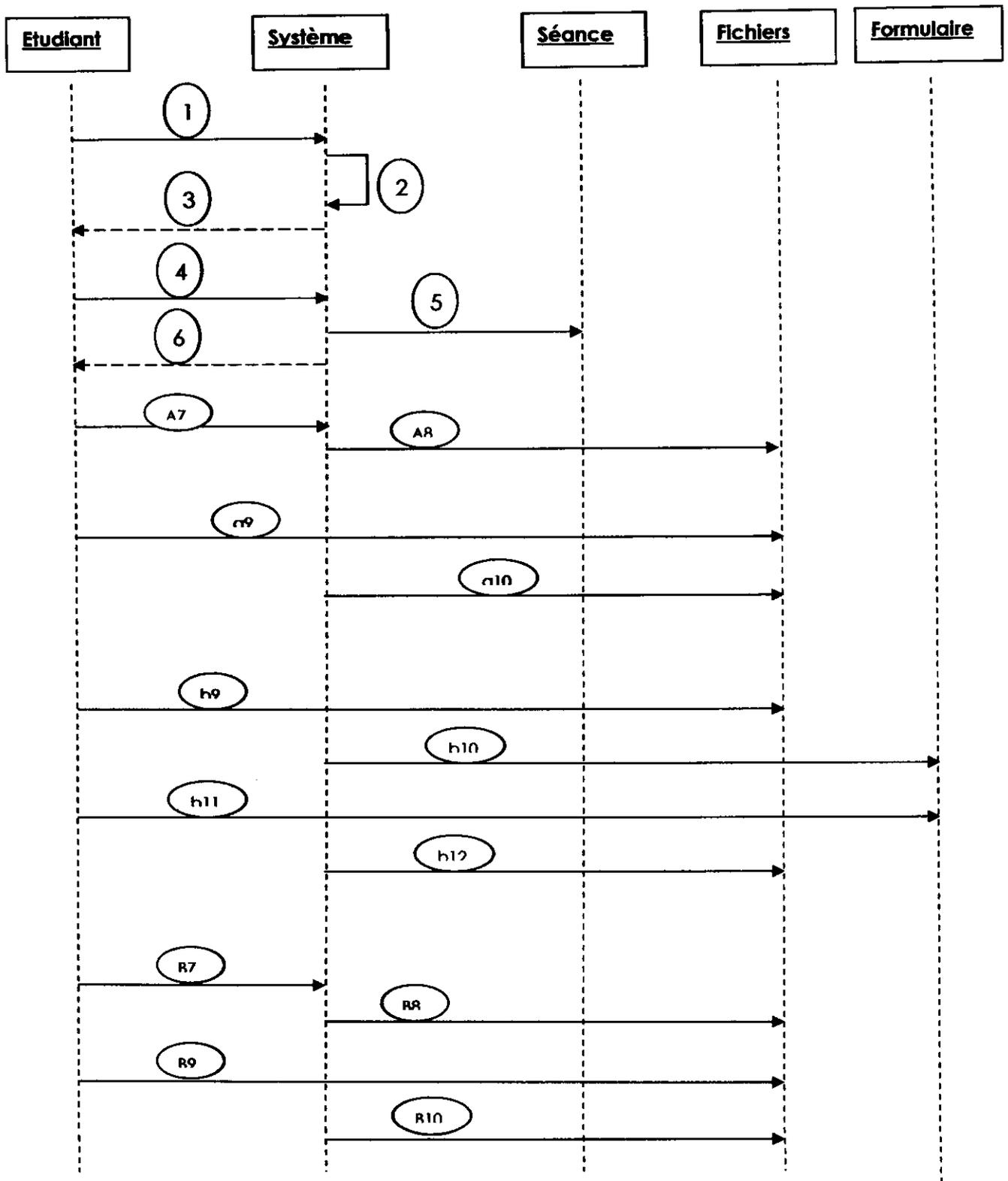
B. Suppression d'un fichier :

7. L'étudiant demande de supprimer un fichier
8. Le système affiche la liste des fichiers concernant le module en cours
9. L'étudiant sélectionne un fichier puis valider
10. Le système supprime le fichier et affiche la nouvelle liste des fichiers

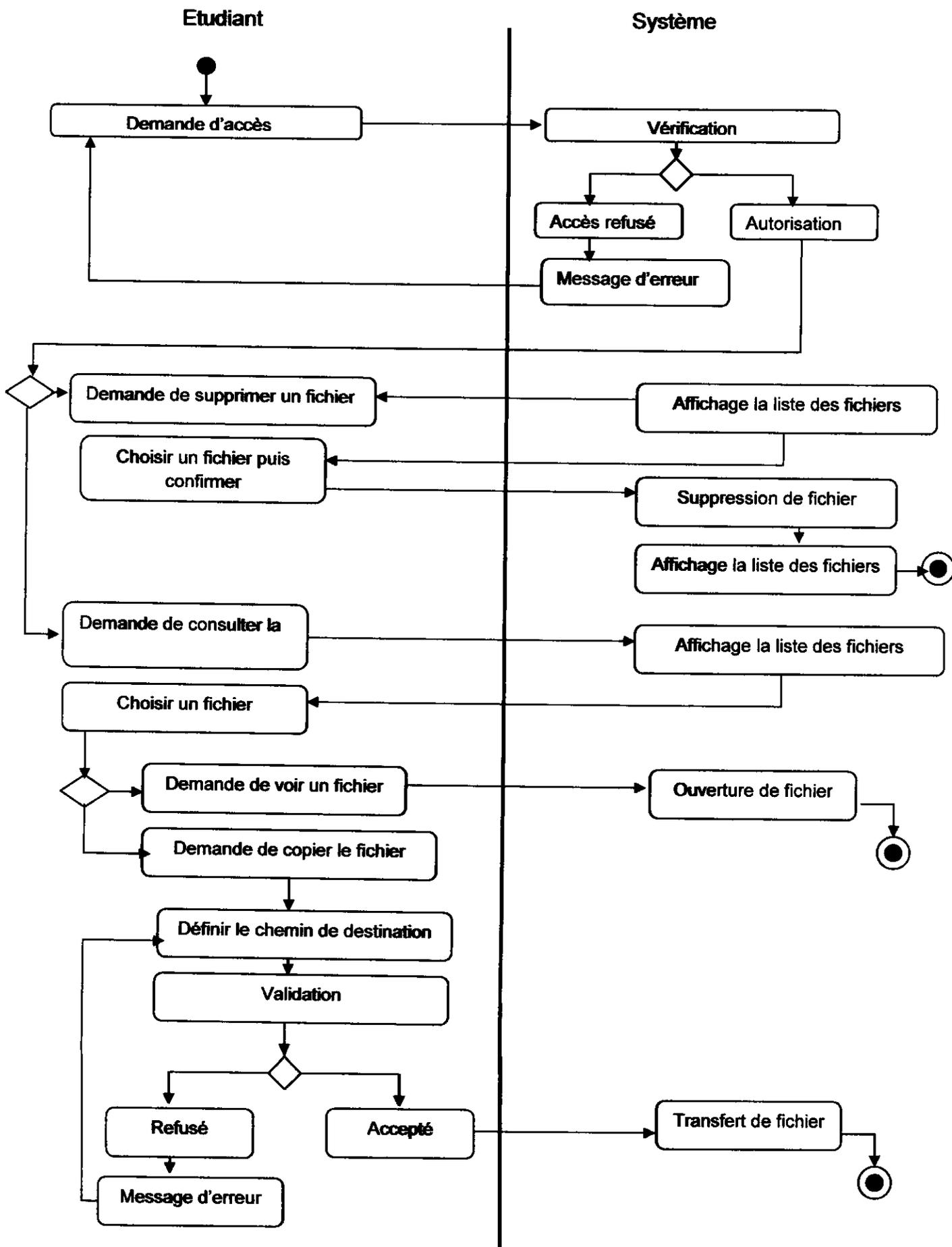
4.2-Diagramme de classe :



4.3-Diagramme de Séquence :



4.4- Diagramme d'activité :

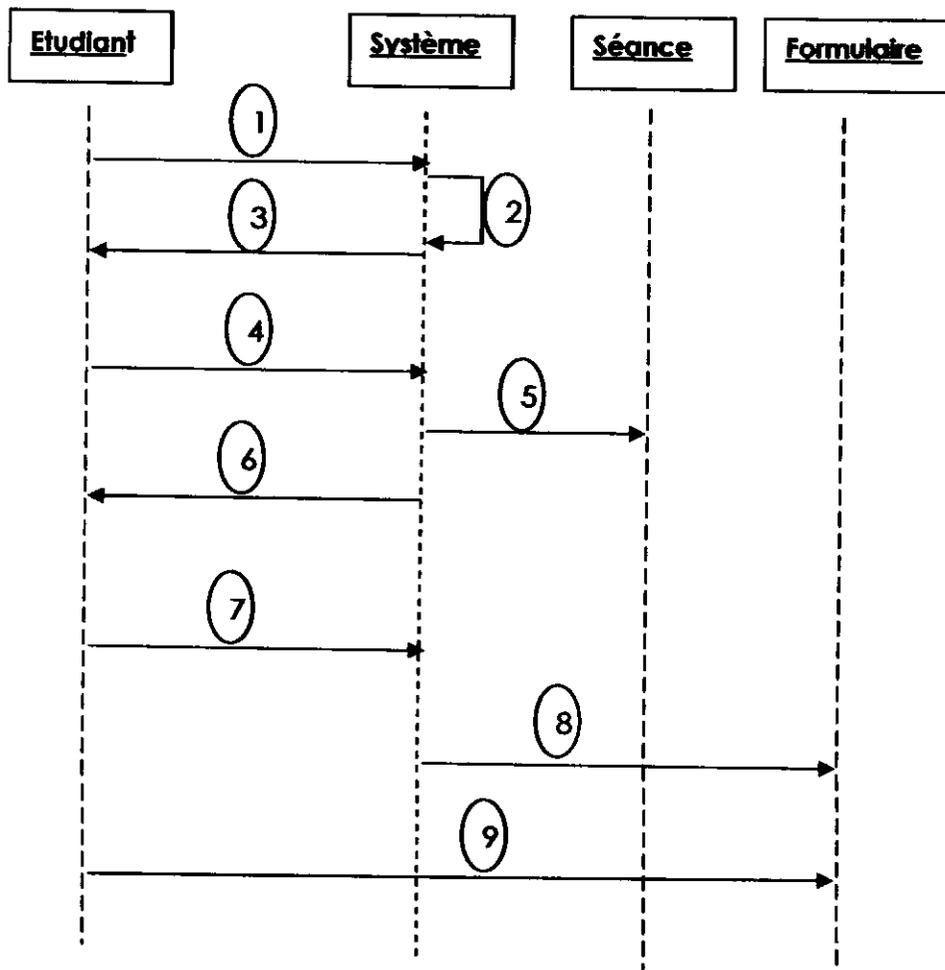


5. Cas d'utilisation << demande d'intervention >> :

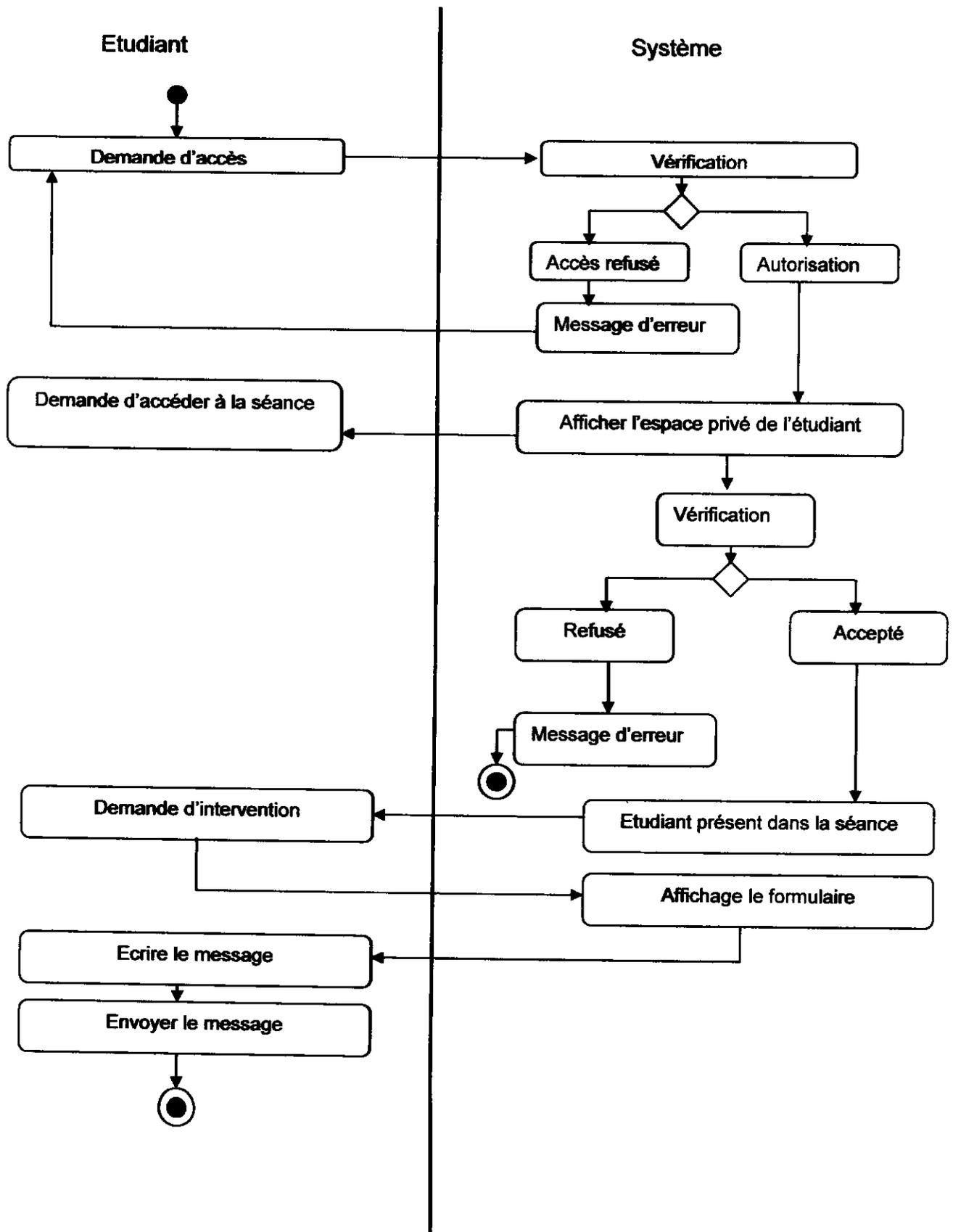
5.1-Scénario :

- 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 l'accès à la séance
- 5- le système vérifie l'accès à la séance
- 6- le système donne l'autorisation à l'étudiant
- 7- l'étudiant demande l'intervention
- 8- le système affiche un formulaire
- 9- l'étudiant écrit le message et le envoyer

5.2-Diagramme de Séquence :



5.3-Diagramme d'activité :

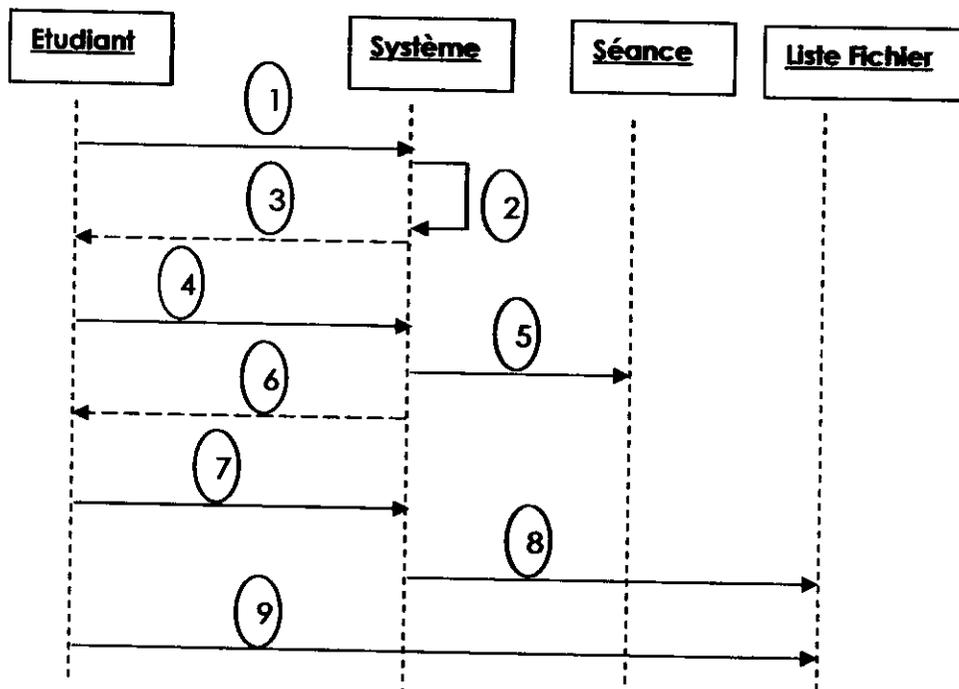


6. Cas d'utilisation <<Envoyer un fichier à l'enseignant>> :

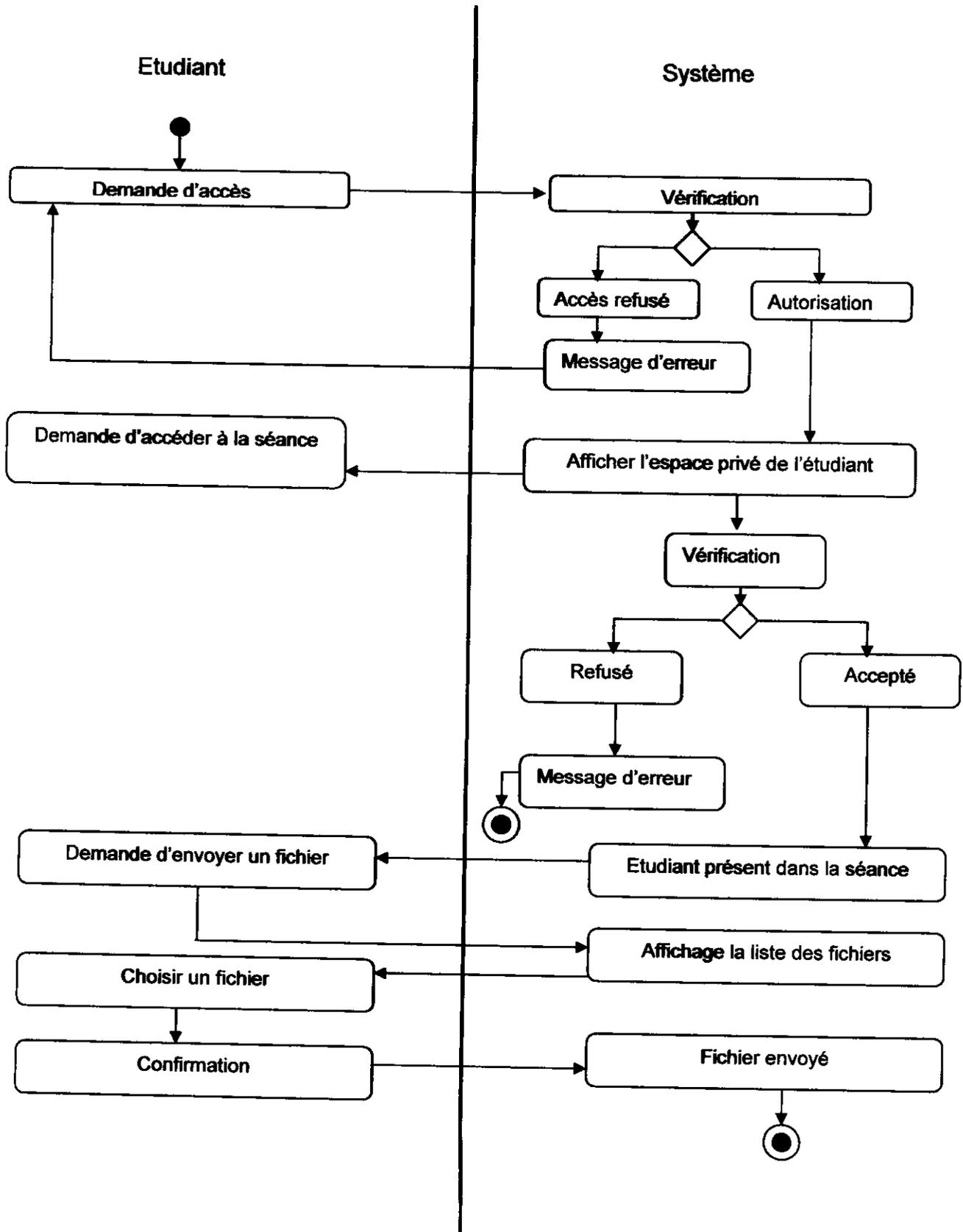
6.1-Scénario :

- 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 l'accès à la séance
- 5- le système vérifie l'accès à la séance
- 6- le système donne l'autorisation à l'étudiant
- 7- l'étudiant demande d'envoyer un fichier
- 8- le système affiche une liste des fichiers
- 9- l'étudiant sélectionne un fichier puis valide

6.2-Diagramme de Séquence :



6.3-Diagramme d'activité :

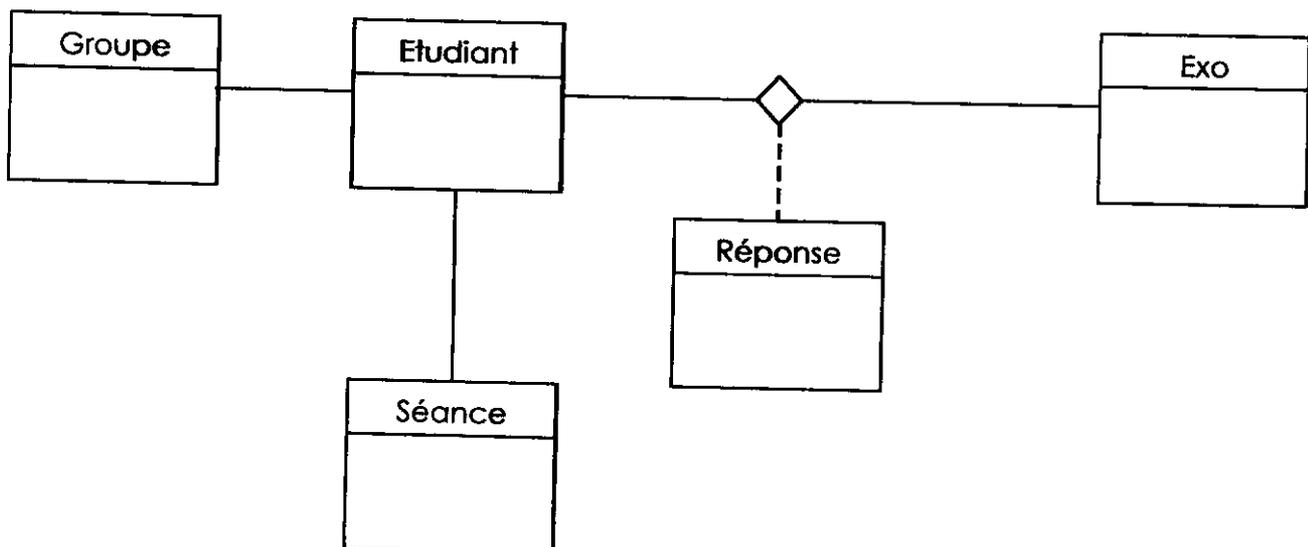


7. Cas d'utilisation << Participer au teste >> :

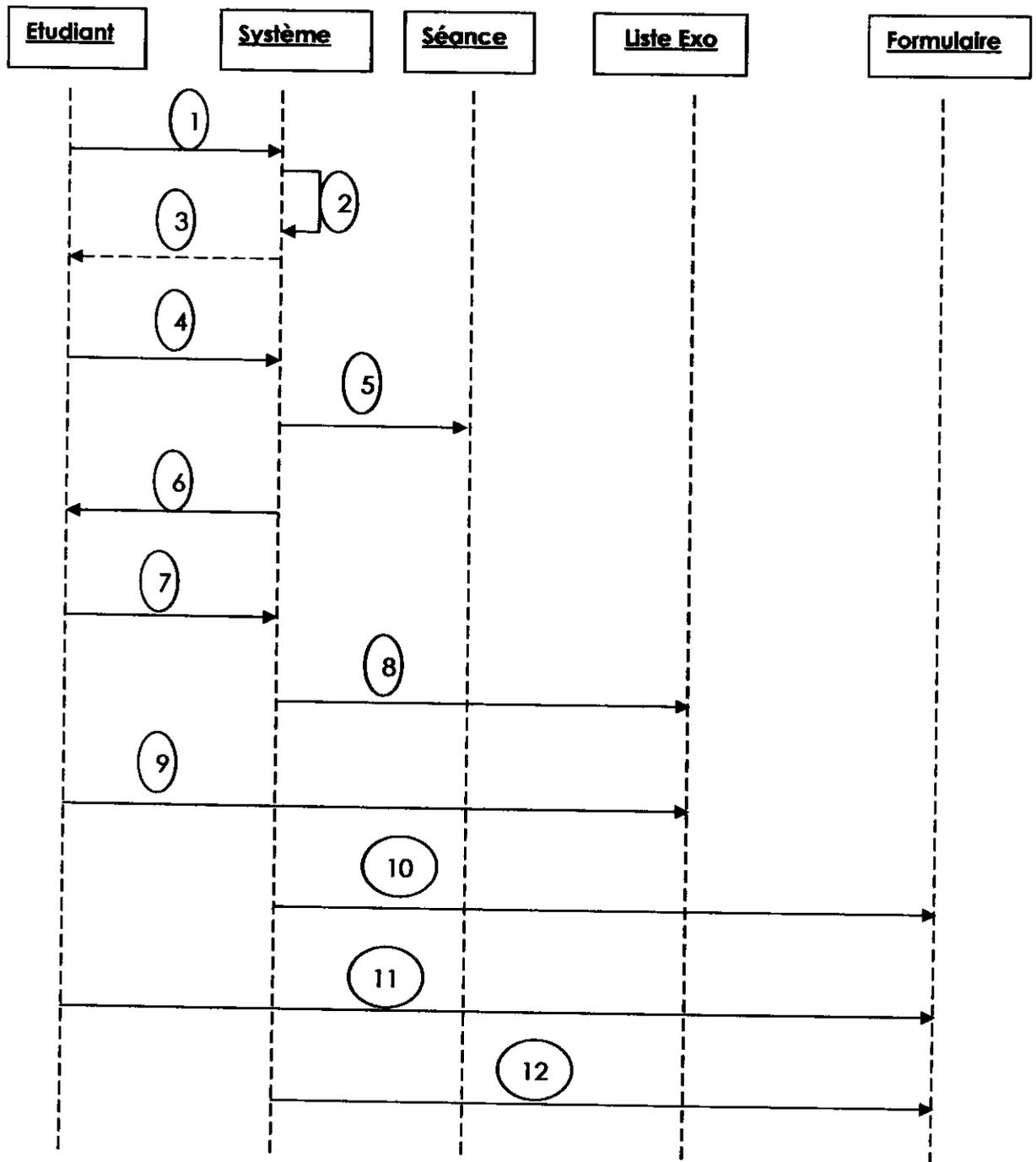
7.1-Scénario :

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 l'accès à la séance en cours.
5. le système vérifie l'accès à la séance.
6. le système donne l'autorisation à l'étudiant.
7. l'étudiant demande de participer au teste.
8. le système affiche une liste des exercices.
9. l'étudiant sélectionne un exercice puis valide.
10. le système affiche un formulaire teste.
11. l'étudiant saisit les données puis valide.
12. le système sauvegarde les données.

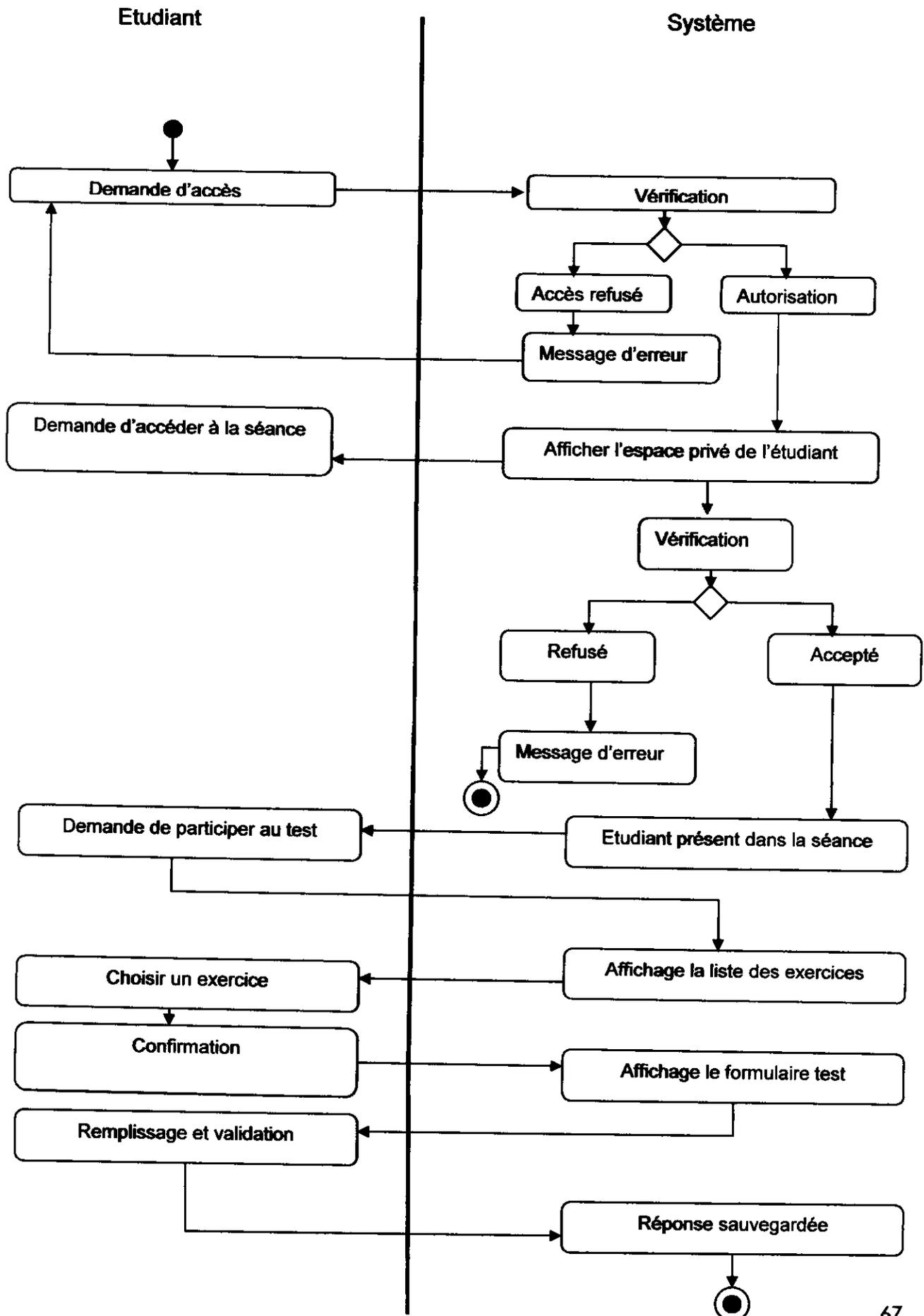
7.2-Diagramme de classe :



7.3-Diagramme de séquence :

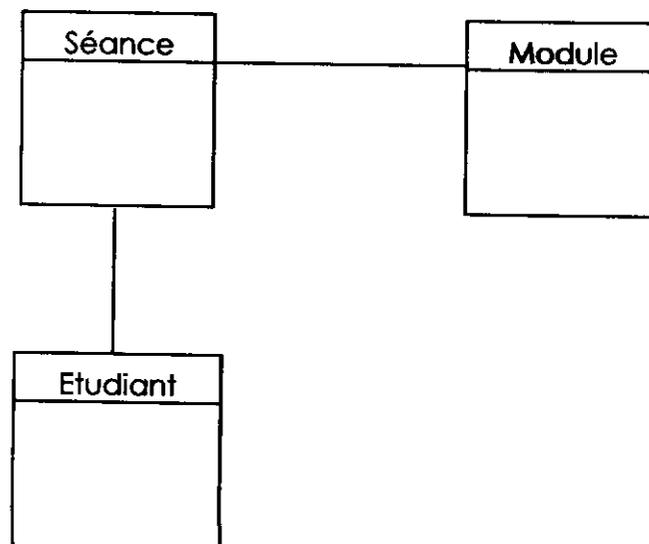


7.4-Diagramme d'activité :

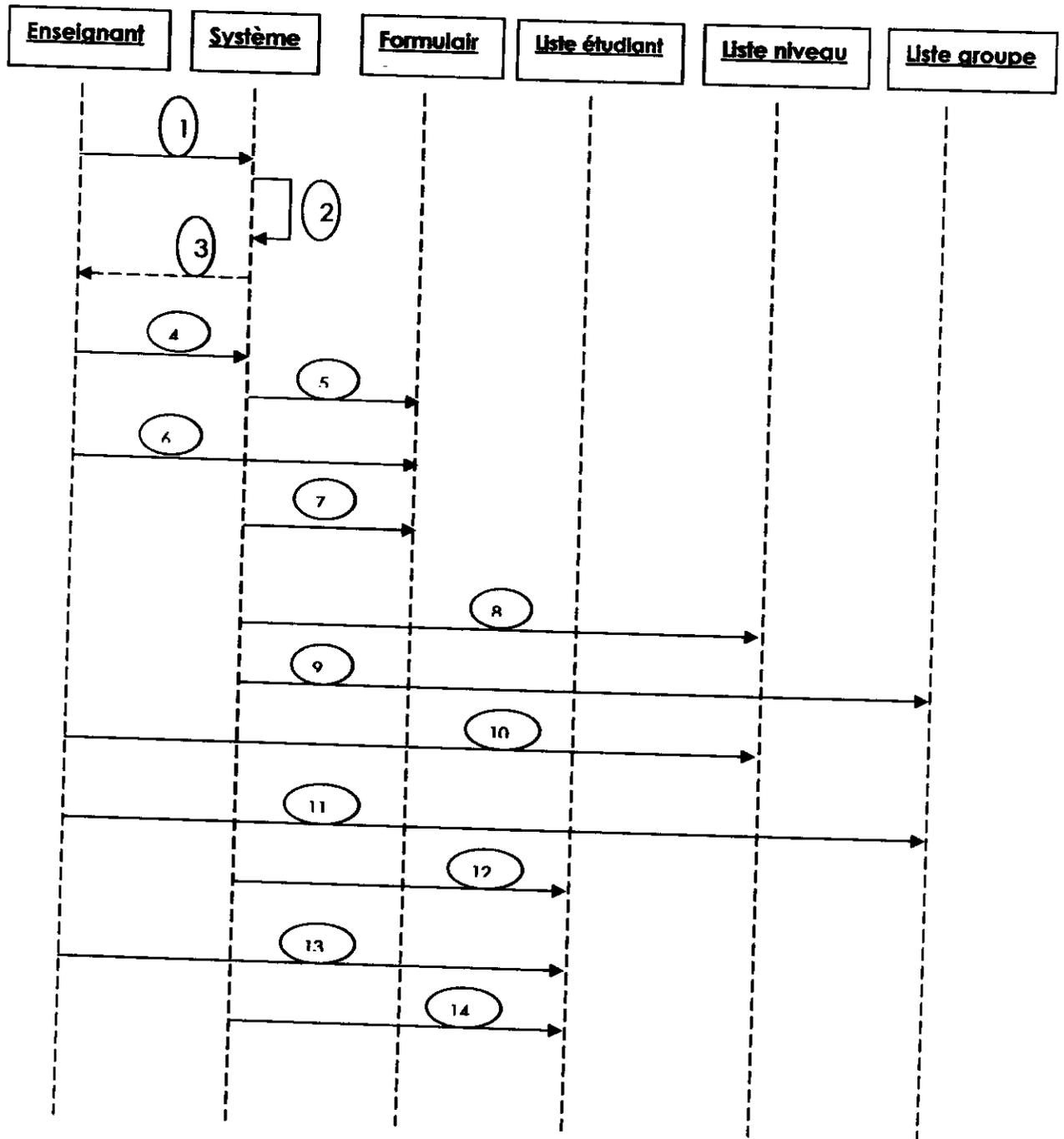


❖ Enseignant :**1. Cas d'utilisation << Créer une séance >> :****1.1-Scénario :**

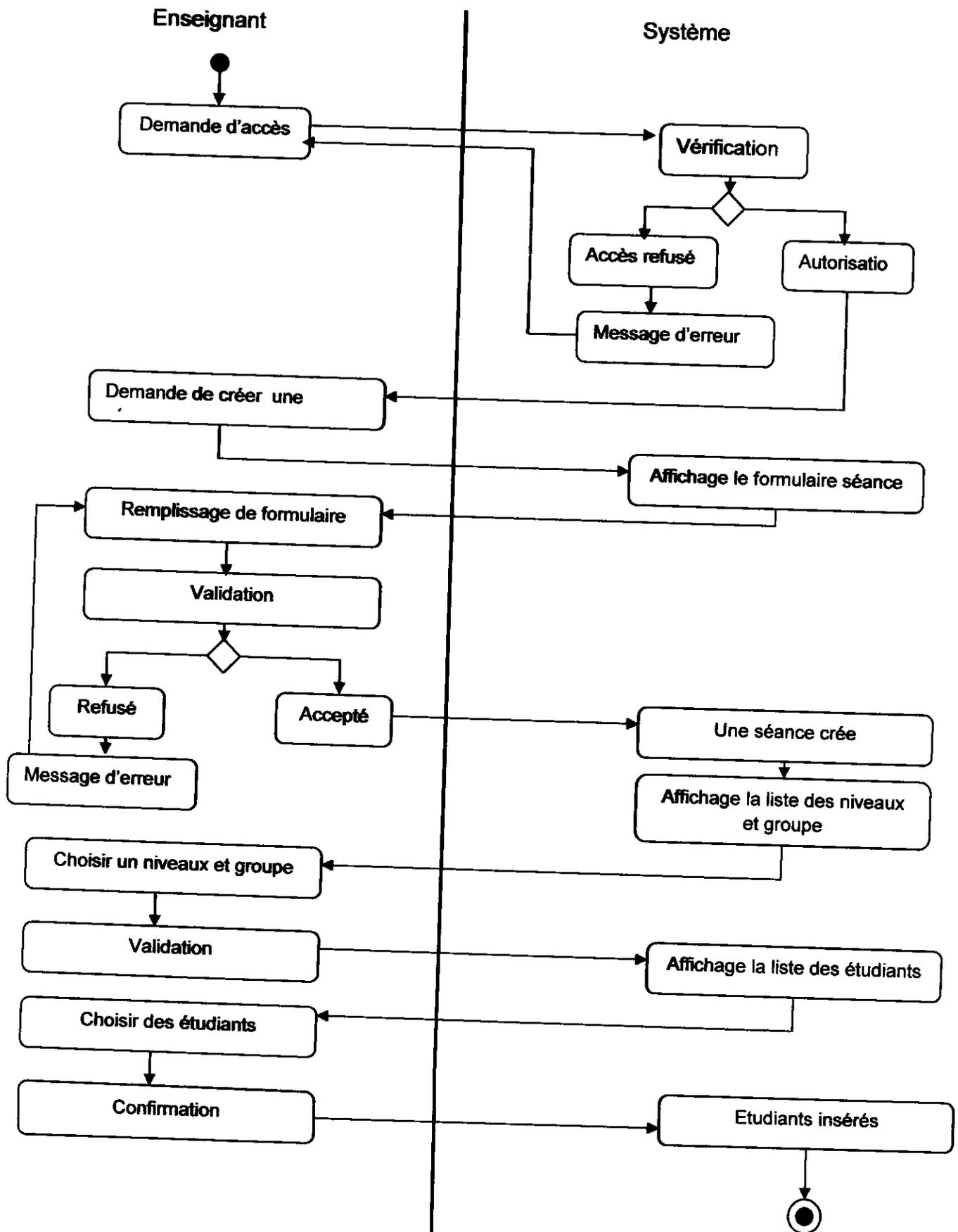
- 1- l'enseignant envoie ses paramètres d'identification
- 2- le système vérifie cette identité
- 3- le système donne l'accès à l'enseignant
- 4- l'enseignant demande de créer une séance
- 5- le système affiche un formulaire séance
- 6- l'enseignant remplit le formulaire puis valide
- 7- le système contrôle puis enregistre les données
- 8- le système affiche une liste des niveaux et groupes
- 9- l'enseignant sélectionne un niveau et un groupe puis valide
- 10- le système affiche une liste des étudiants
- 11- l'enseignant confirme l'insertion des étudiants
- 12- le système insère les étudiants

1.2-Diagramme de classe :

1.3-Diagramme de séquence :



1.4-Diagramme d'activité :



2. Cas d'utilisation << mettre à jour les séances >> :

2.1-Scénario :

1. l'enseignant envoie ses paramètres d'identification
2. le système vérifie cette identité
3. le système donne l'accès à l'enseignant

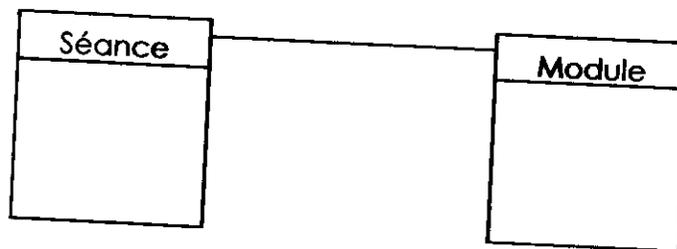
A. Modifier une séance

4. l'enseignant demande de modifier une séance
5. le système affiche une liste des séances.
6. L'enseignant sélectionne une séance puis valide
7. Le système affiche un formulaire de la séance
8. L'enseignant modifie les paramètres puis valide
9. Le système contrôle puis sauvegarde les données.

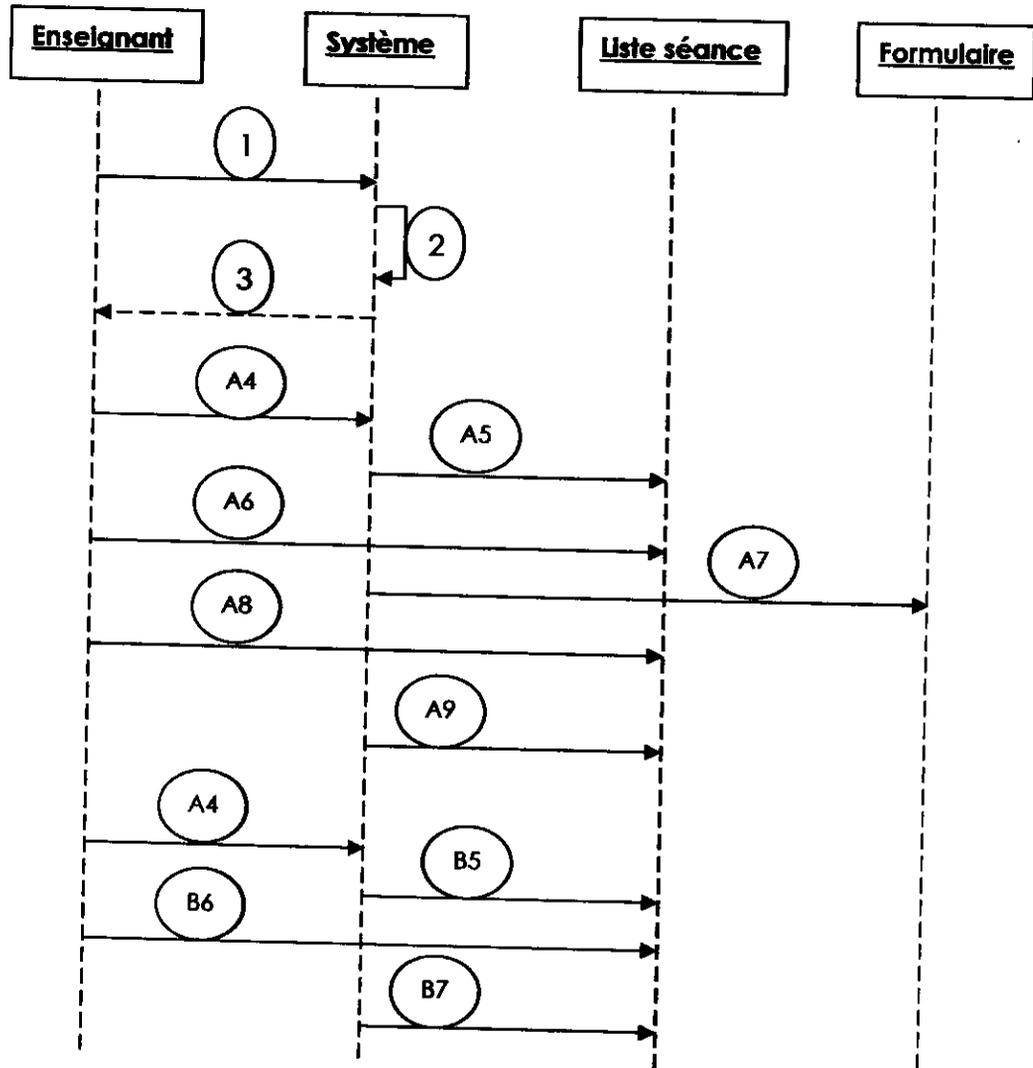
B. Supprimer une séance :

4. l'enseignant demande de supprimer une séance
5. le système affiche une liste des séances.
6. L'enseignant sélectionne une séance puis valide
7. Le système supprime la séance et affiche la liste des séances

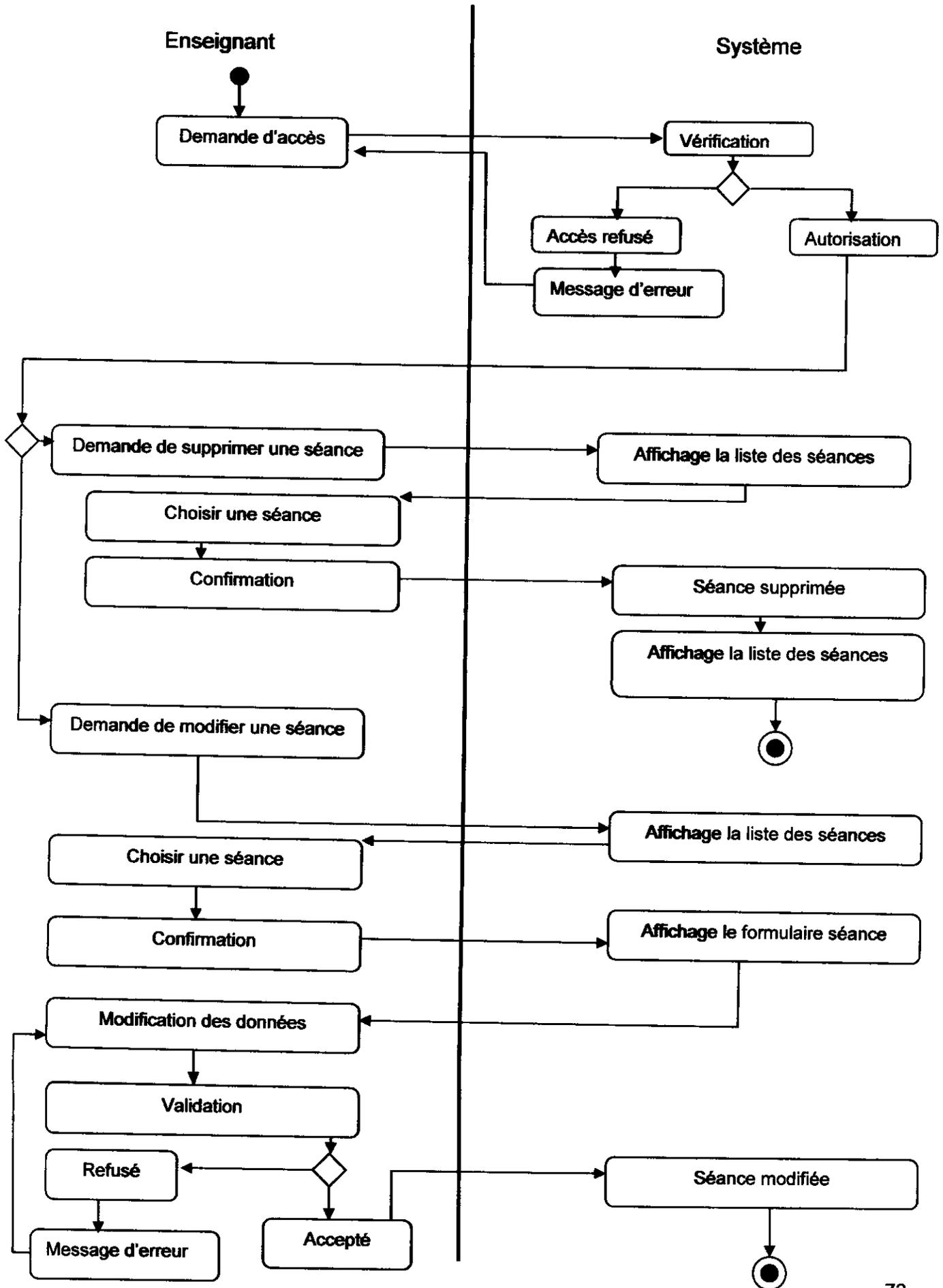
2.2-Diagramme de classe :



2.3-Diagramme de séquence :



2.4-Diagramme d'activité :

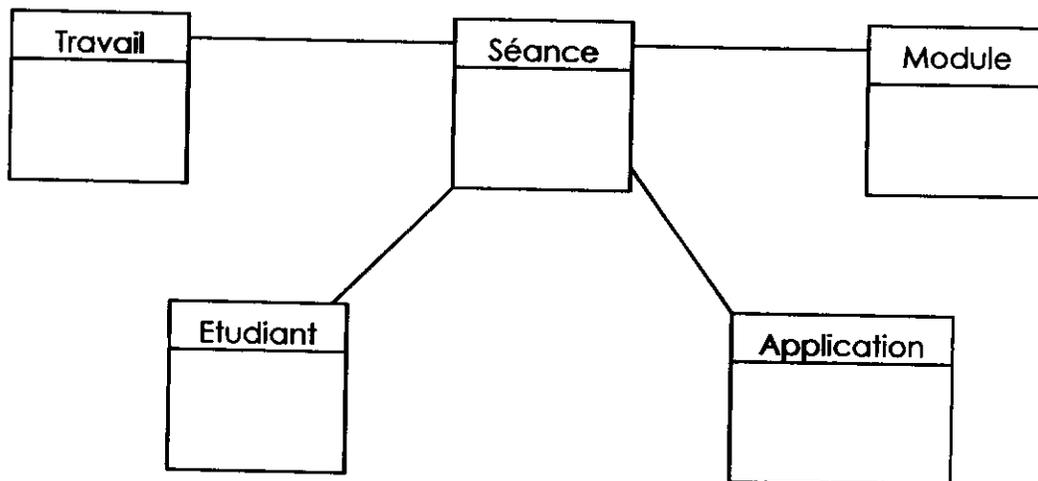


3. cas d'utilisation << lancer une séance >> :

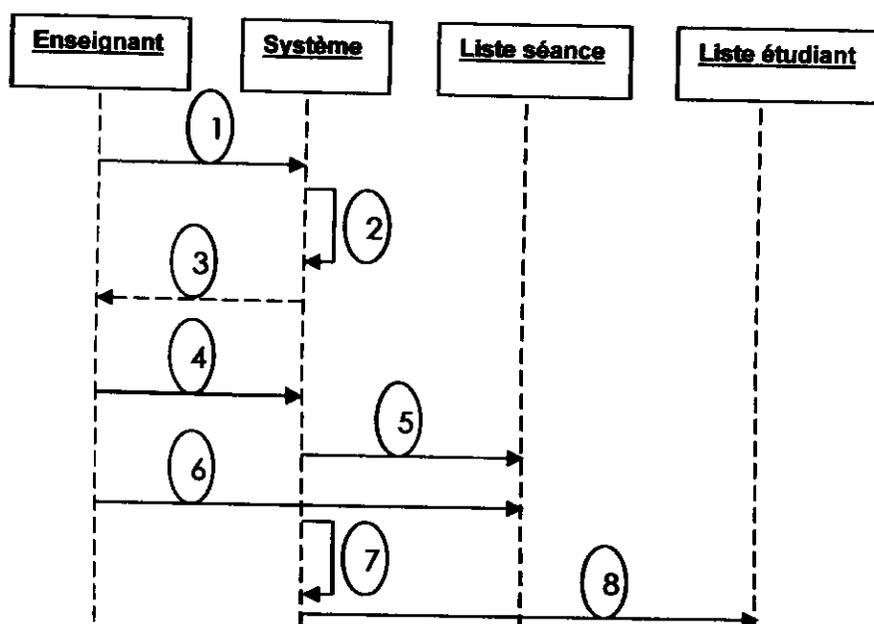
3.1-Scénario :

- 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 de démarrer une séance
- 5- le système affiche une liste des séances.
- 6- L'enseignant sélectionne une séance puis valide
- 7- Le système vérifie l'accès à séance
- 8- Le système affiche une liste des étudiants

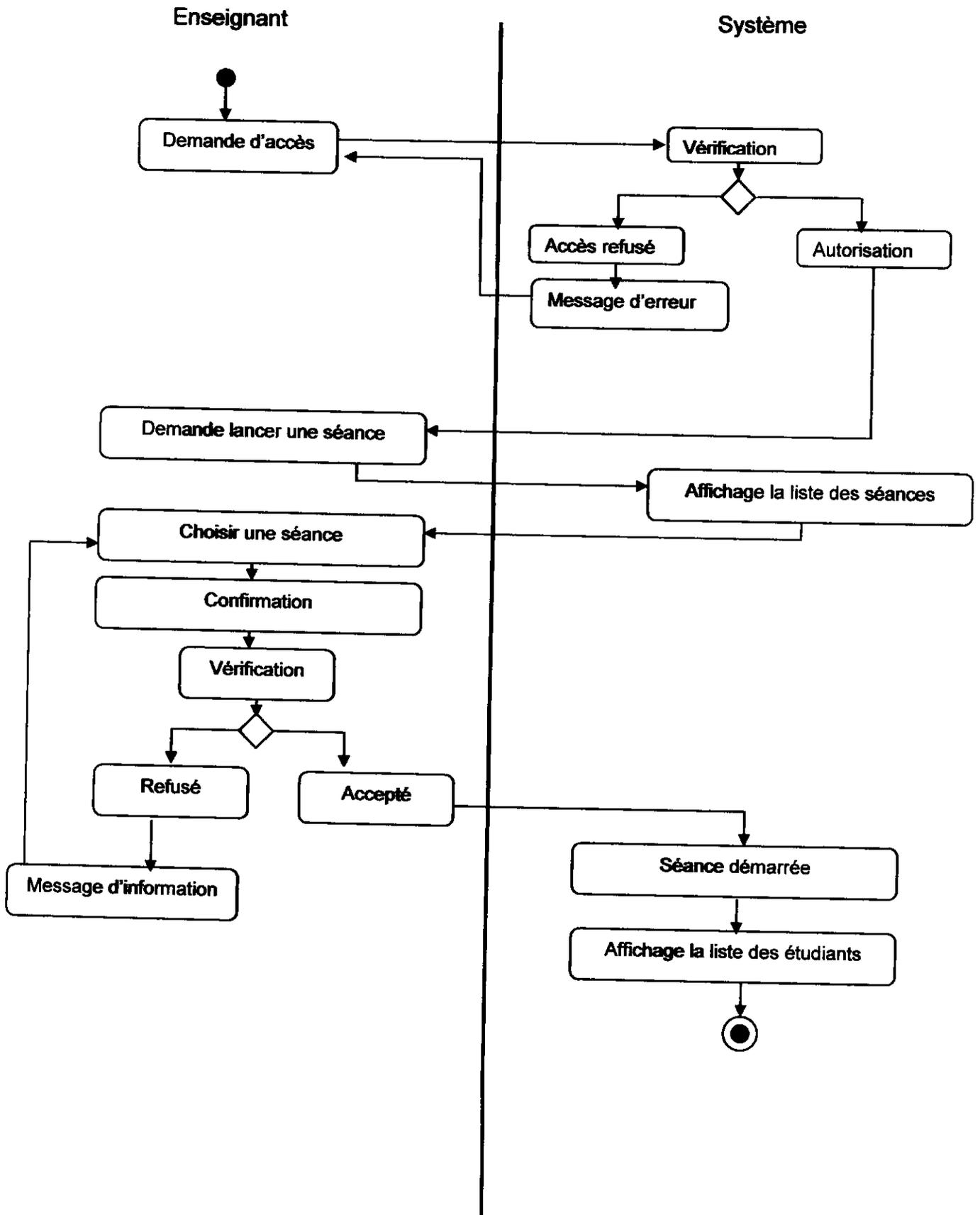
3.2-Diagramme de classe :



3.3-Diagramme de séquence :



3.4-Diagramme d'activité :



4. Cas d'utilisation <<mettre à jour les étudiants de la séance >> :

4.1-Scénario :

- 1- l'enseignant envoie ses paramètres d'identification
- 2- le système vérifie cette identité
- 3- le système donne l'accès à l'enseignant

A. Affecter des étudiants :

- 4- l'enseignant demande d'affecter des étudiants pour une séance
- 5- le système affiche une liste des séances, niveaux et groupes
- 6- l'enseignant sélectionne une séance, un niveau et un groupe puis valide
- 7- le système affiche une liste des étudiants
- 8- l'enseignant confirme l'insertion des étudiants
- 9- le système insère les étudiants et affiche la liste

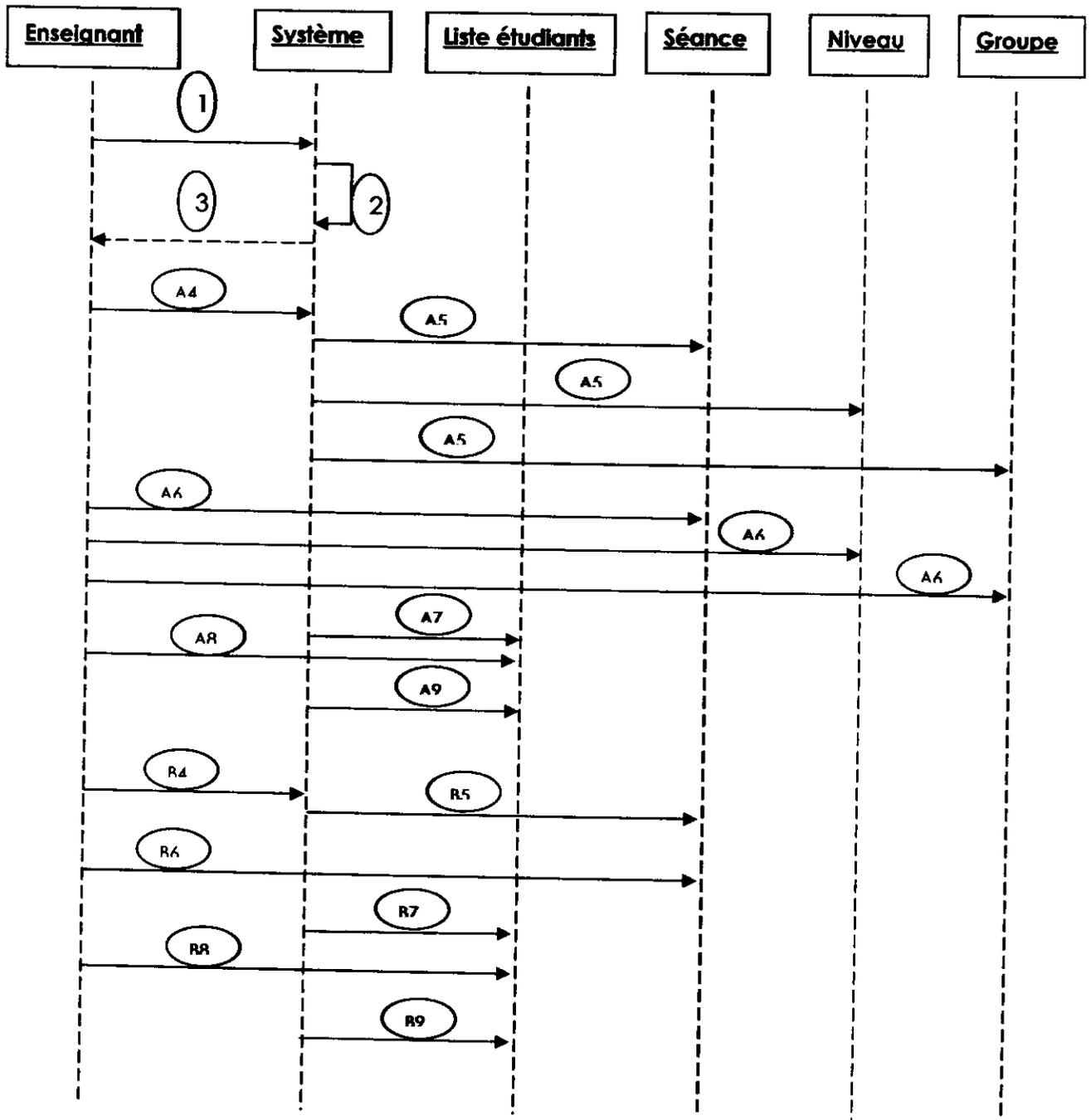
B. Supprimer des étudiants :

- 4- l'enseignant demande de supprimer des étudiants
- 5- le système affiche une liste des séances
- 6- l'enseignant sélectionne une séance, puis valide
- 7- le système affiche une liste des étudiants
- 8- l'enseignant sélectionne un étudiant puis valide
- 9- le système supprime l'étudiant et affiche la liste des étudiants

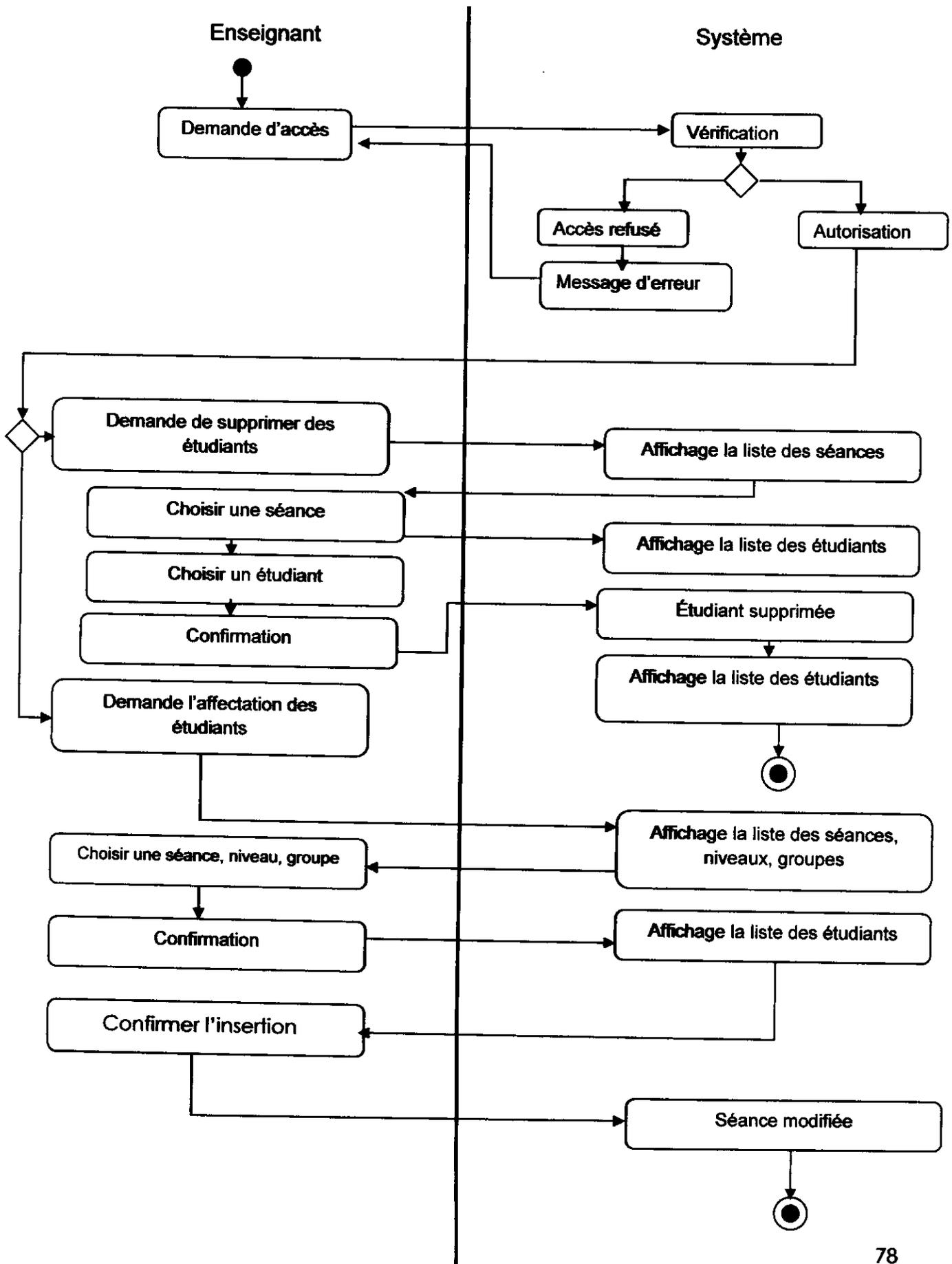
4.2-Diagramme de classe :



4.3-Diagramme de Séquence :



4.4-Diagramme d'activité :

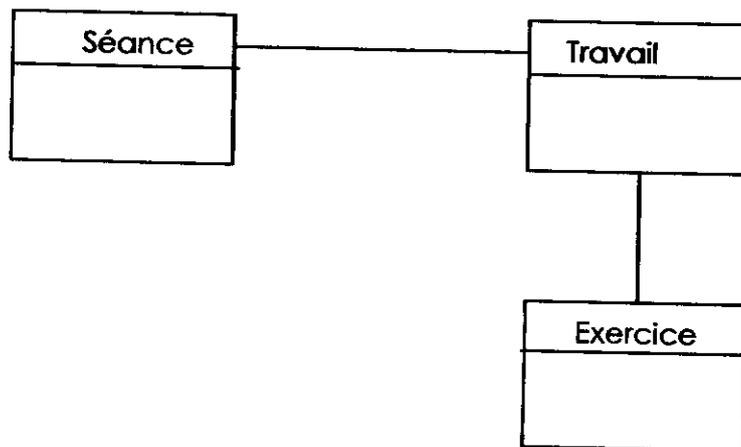


5. Cas d'utilisation <<Préparer un TP >>

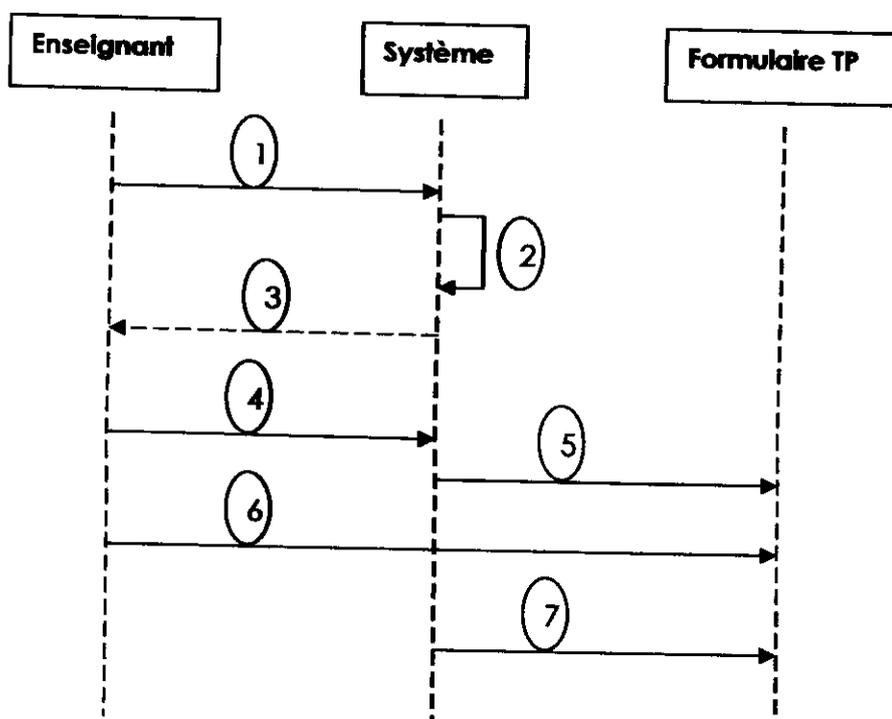
5.1-Scénario :

- 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 de préparer un TP
- 5- le système affiche un formulaire TP
- 6- l'enseignant remplir le formulaire puis valide
- 7- le système sauvegarde les données

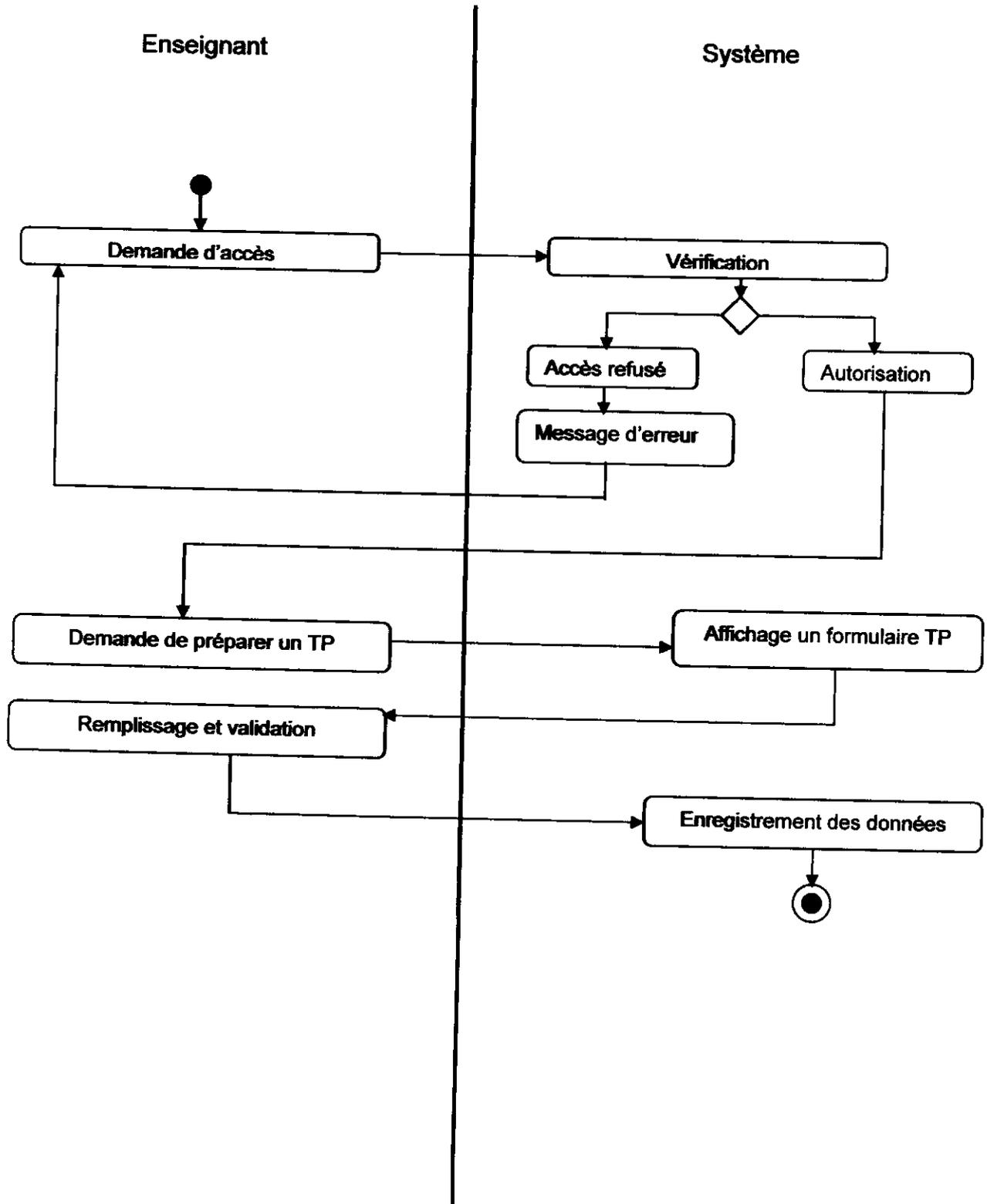
5.2-Diagramme de classe :



5.3-Diagramme de séquence :



5.4-Diagramme d'activité :

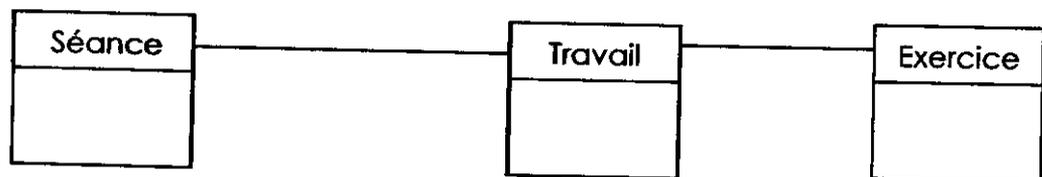


6. Cas d'utilisation <<Préparer un Test >> :

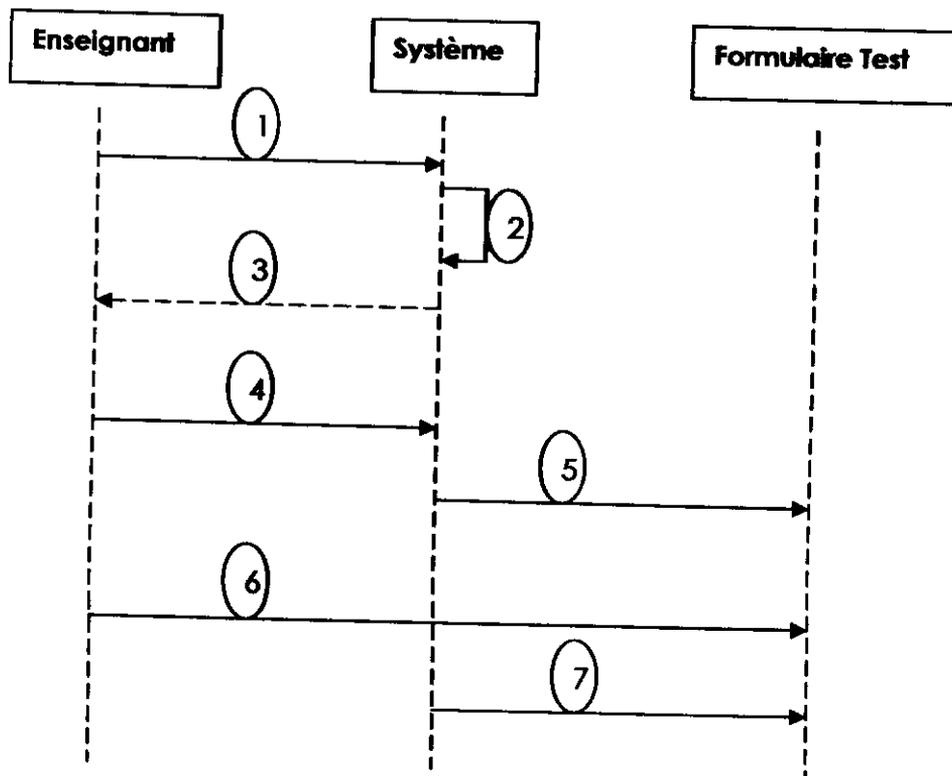
6.1-Scénario :

- 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 de préparer un Test
- 5- le système affiche un formulaire Test
- 6- l'enseignant remplit le formulaire puis valide
- 7- le système sauvegarde les données

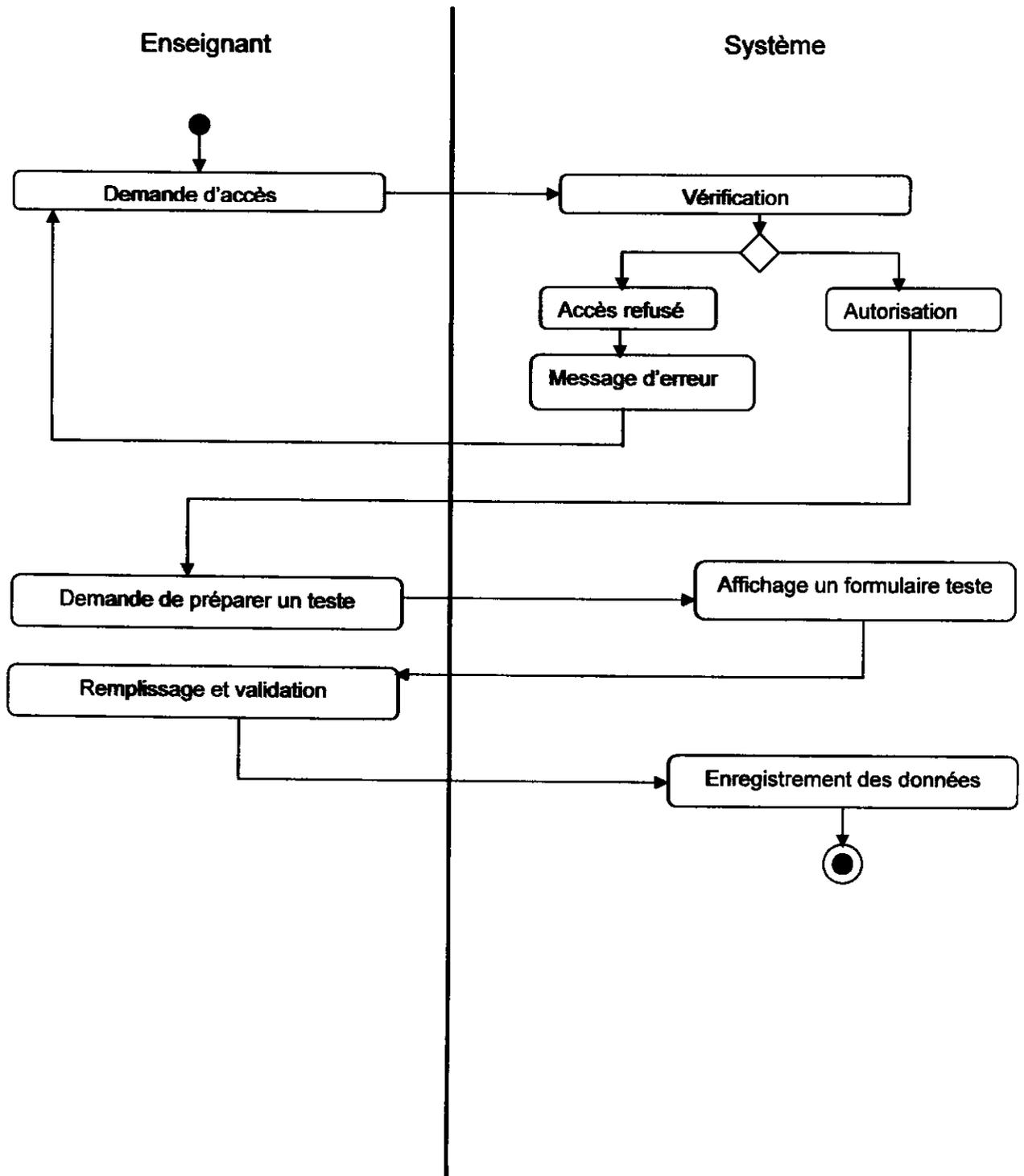
6.2-Diagramme de classe :



6.3-Diagramme de séquence :



6.4-Diagramme d'activité :

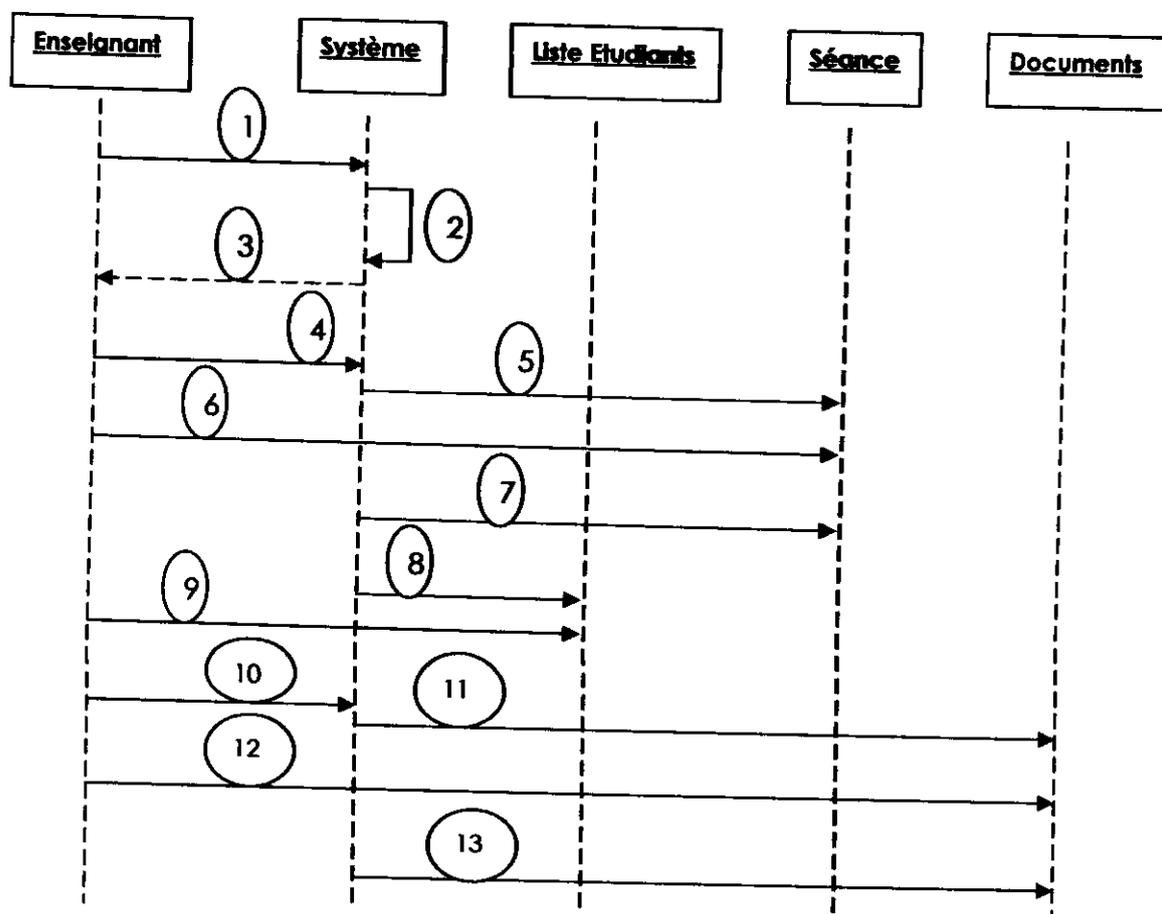


7. Cas d'utilisation <<Envoyer un Document à l'étudiant>> :

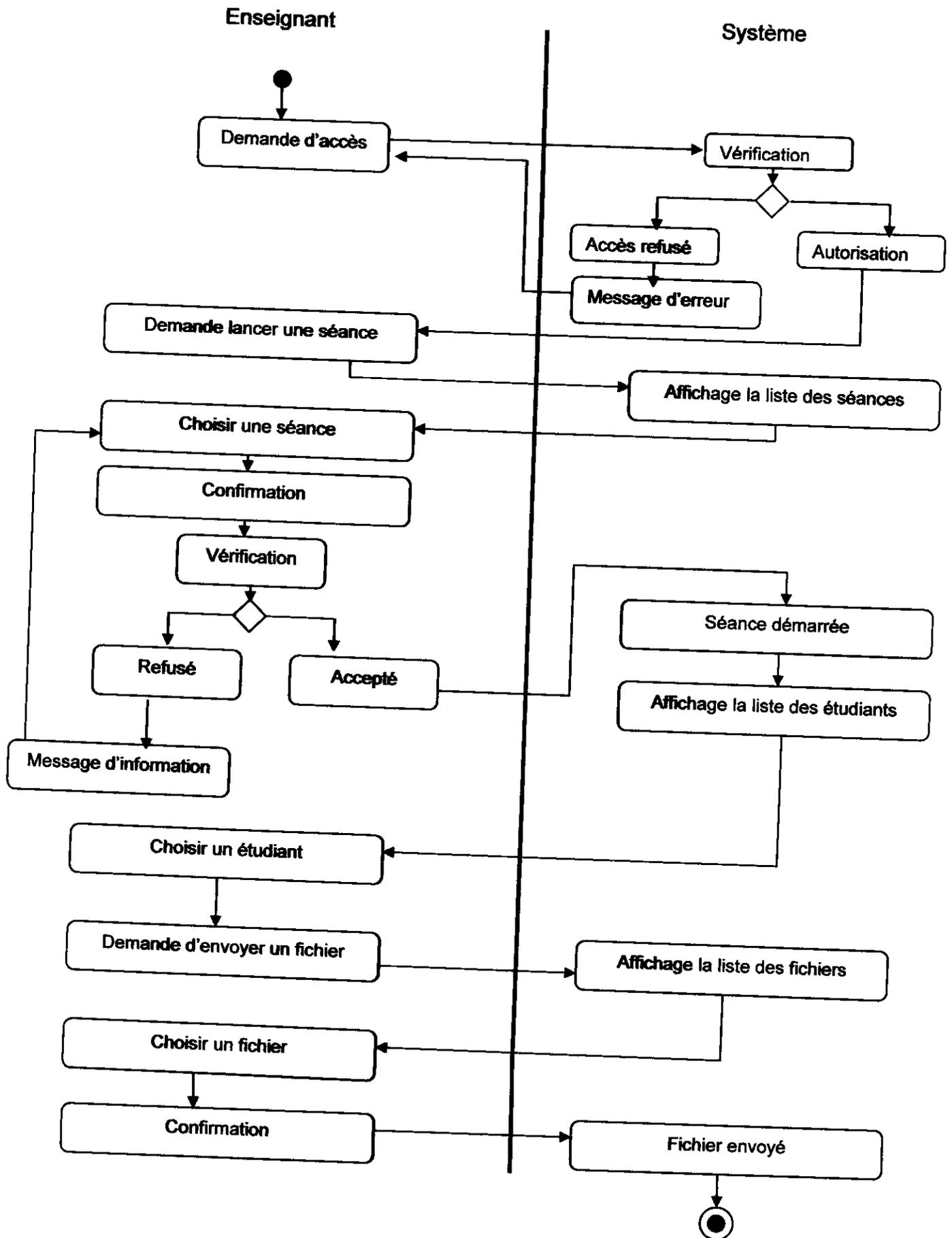
7.1-Scénario :

- 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 de démarrer une séance
- 5- le système affiche une liste des séances
- 6- l'enseignant sélectionne une séance puis valide
- 7- le système vérifie l'accès à la séance
- 8- le système affiche une liste des étudiants
- 9- l'enseignant sélectionne un étudiant de la liste
- 10- l'enseignant demande d'envoyer un document
- 11- le système affiche une liste des documents
- 12- l'enseignant sélectionne un document puis valide
- 13- le système envoyer le document

7.2-Diagramme de Séquence :



7.3-Diagramme d'activité :

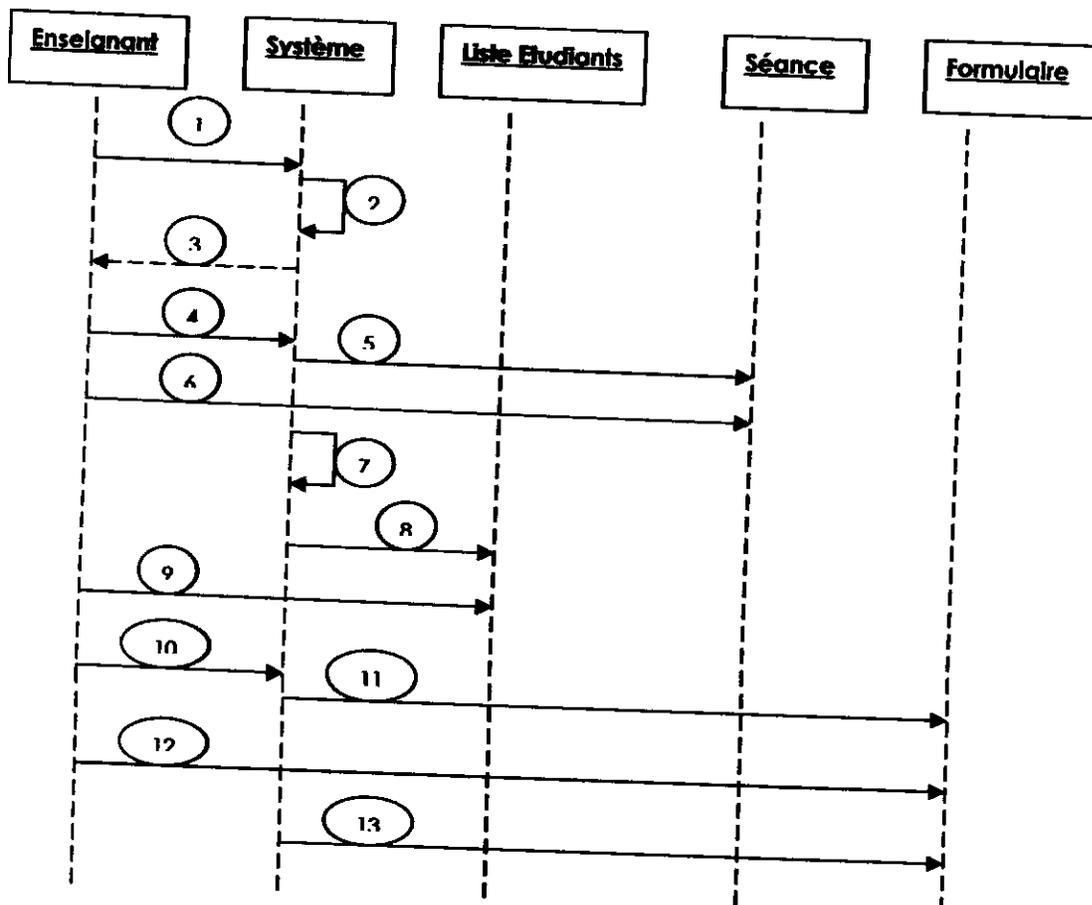


8. Cas d'utilisation <<Envoyer un Message à l'étudiant>> :

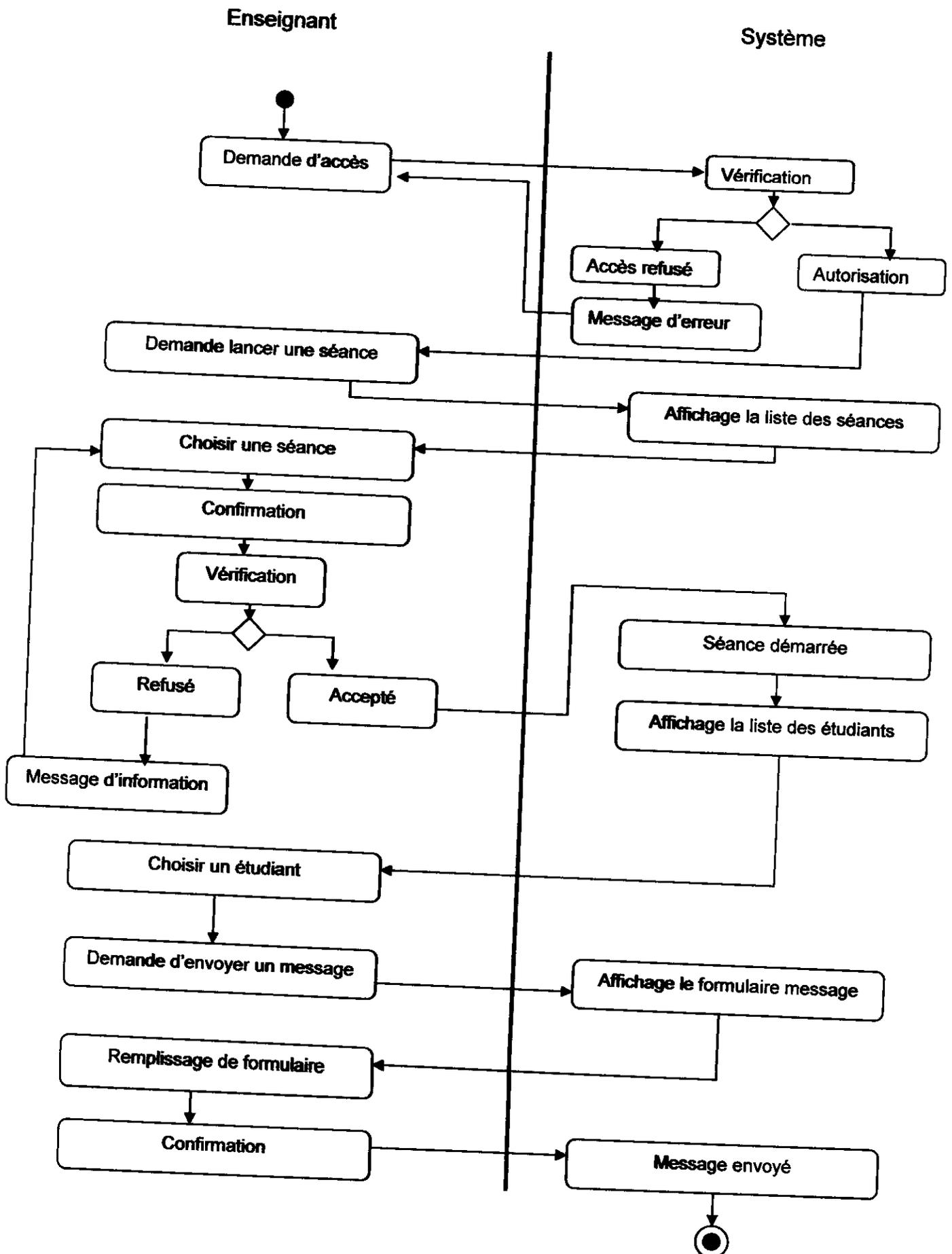
8.1-Scénario :

- 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 de démarrer une séance
- 5- le système affiche une liste des séances
- 6- l'enseignant sélectionne une séance puis valide
- 7- le système vérifie l'accès à la séance
- 8- le système affiche une liste des étudiants
- 9- l'enseignant sélectionne un étudiant de la liste
- 10- l'enseignant demande d'envoyer un message
- 11- le système affiche un formulaire
- 12- l'enseignant saisit le message sur le formulaire puis valide
- 13- le système envoyer le message

8.2-Diagramme de Séquence :

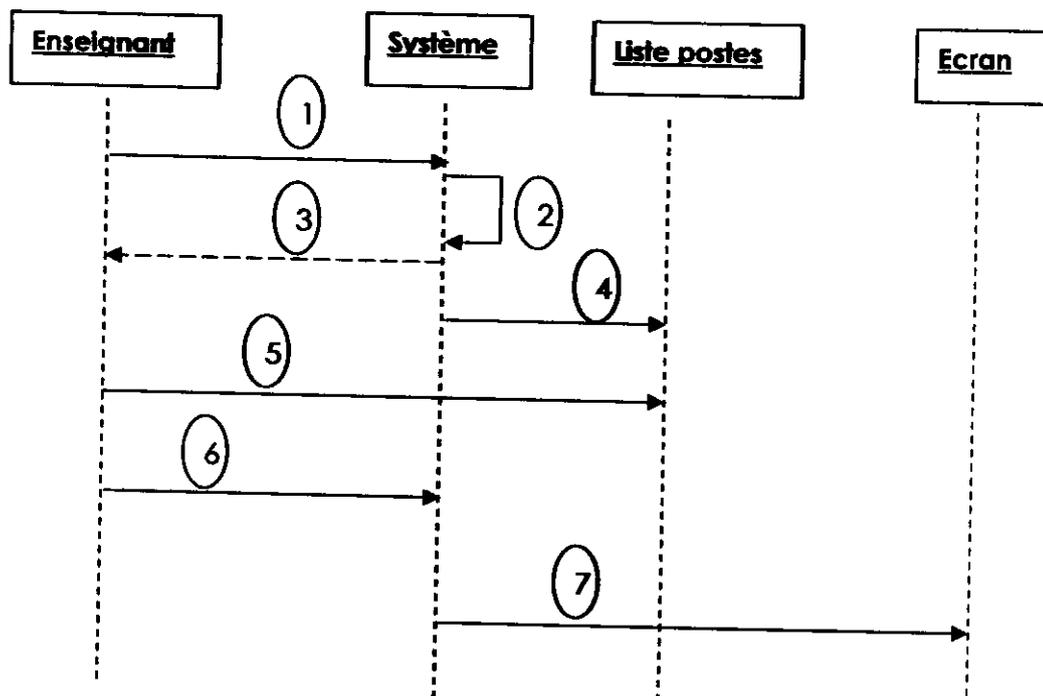


8.3-Diagramme d'activité :

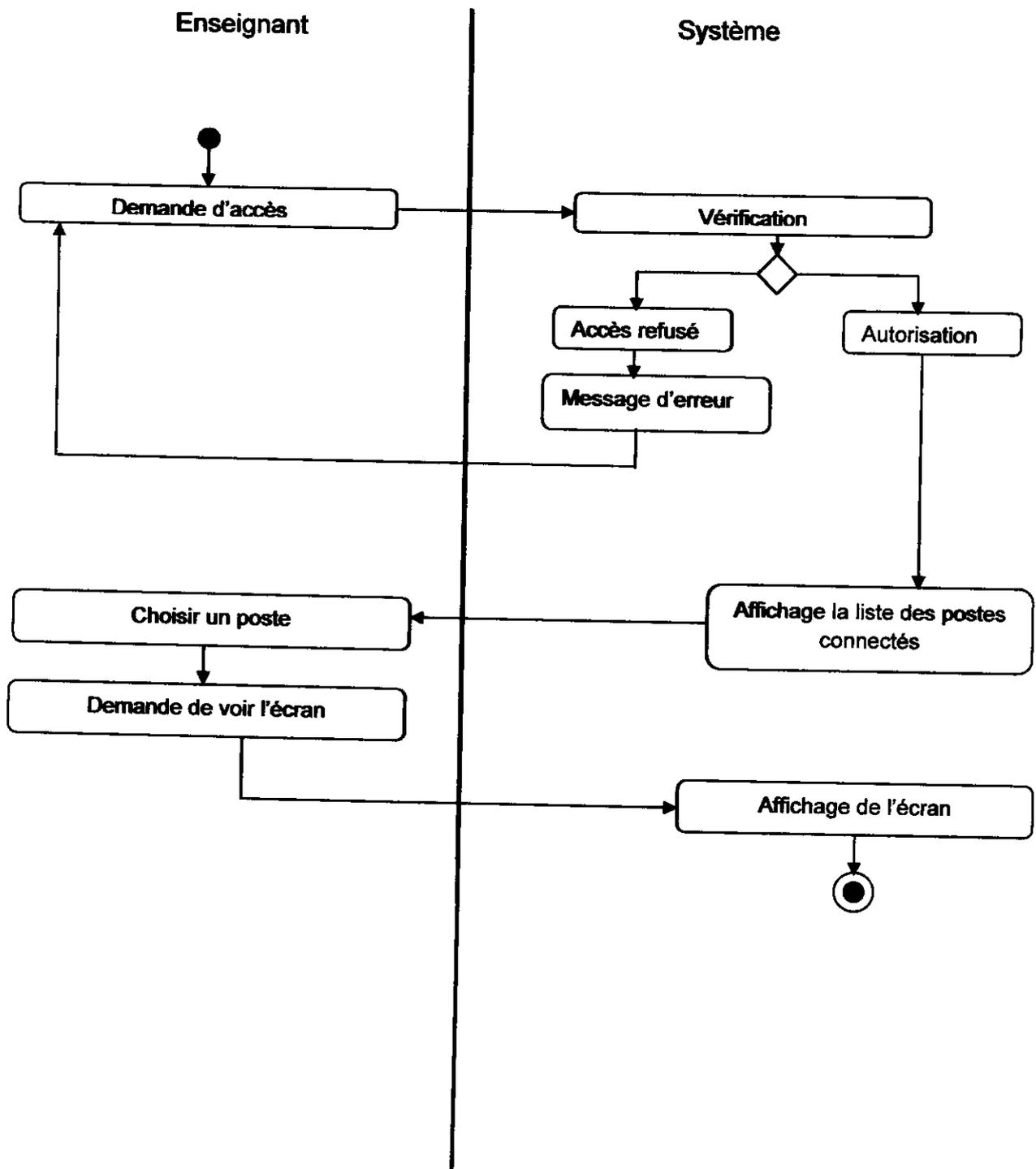


9. Cas d'utilisation <<Voir l'écran d'un étudiant>> :**9.1-Scénario :**

- 1- l'enseignant envoie ses paramètres d'identification
- 2- le système vérifie cette identité
- 3- le système donne l'accès à l'enseignant
- 4- le système affiche une liste des postes connectés
- 5- l'enseignant demande voir l'écran d'un poste
- 6- l'enseignant sélectionne un poste de la liste
- 7- le système affiche l'écran de poste

9.2-Diagramme de Séquence :

9.3-Diagramme d'activité :



10. Cas d'utilisation <<Faire une Démonstration >> :

10.1-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 de démarrer une séance
- 5- le système affiche une liste des séances
- 6- l'enseignant sélectionne une séance puis valide
- 7- le système vérifie l'accès à la séance
- 8- le système donne l'accès à l'enseignant

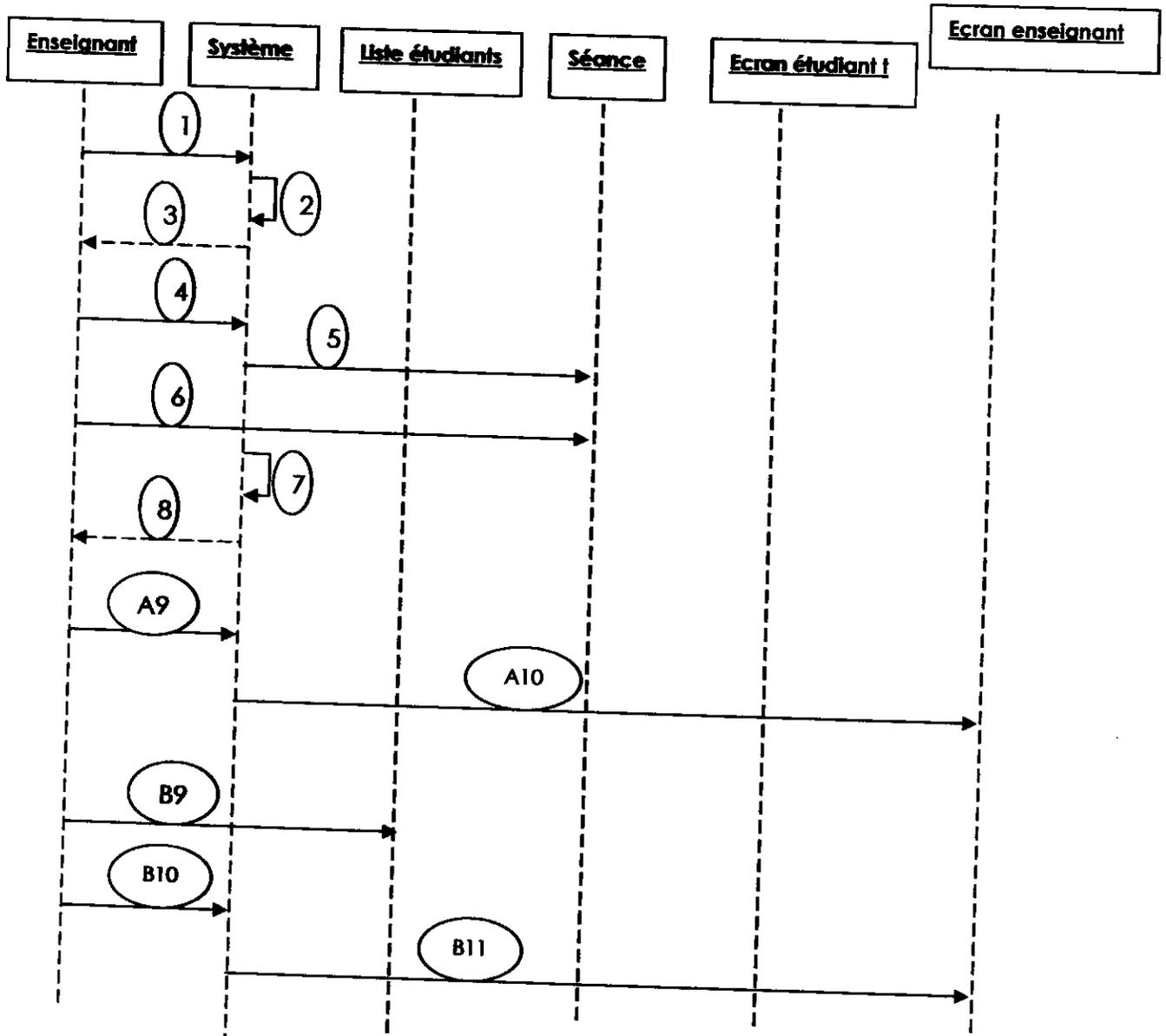
A. Démonstration Enseignant :

- 9- l'enseignant demande de faire une démonstration enseignant
- 10- le système affiche l'écran de l'enseignant sur tous les postes

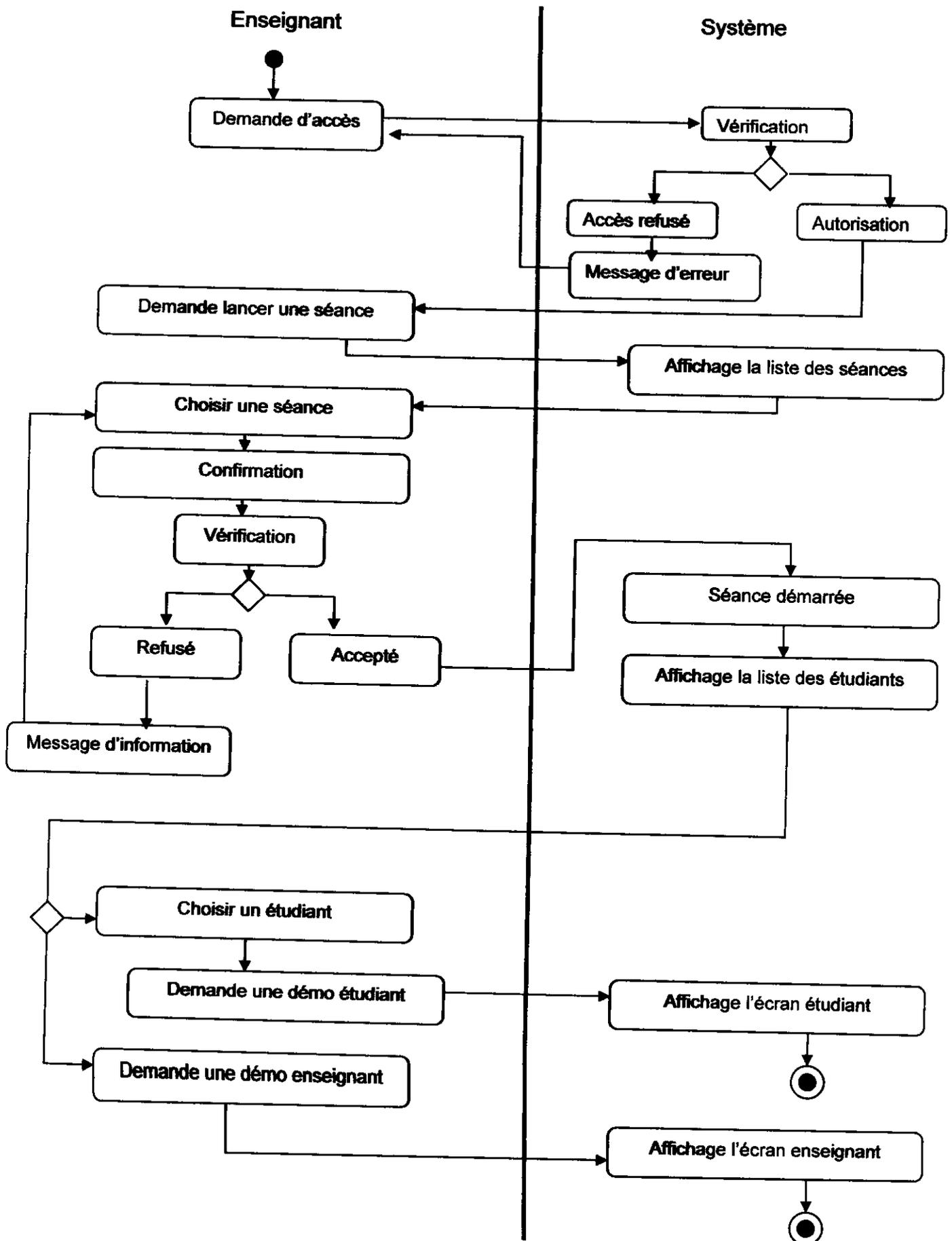
B. Démonstration Etudiant :

- 9- l'enseignant sélectionne un étudiant de la liste
- 10- l'enseignant demande de faire une démonstration Etudiant
- 11- le système affiche l'écran de l'étudiant sélectionné sur tous les postes

10.2-Diagramme de Séquence :



10.3-Diagramme d'activité :



11. Cas utilisation <<gérer les notes des étudiants >> :

11.1-Scénario :

1. l'enseignant envoie ses paramètres d'identification.
2. le système vérifie cette identité.
3. le système donne l'accès à l'enseignant.

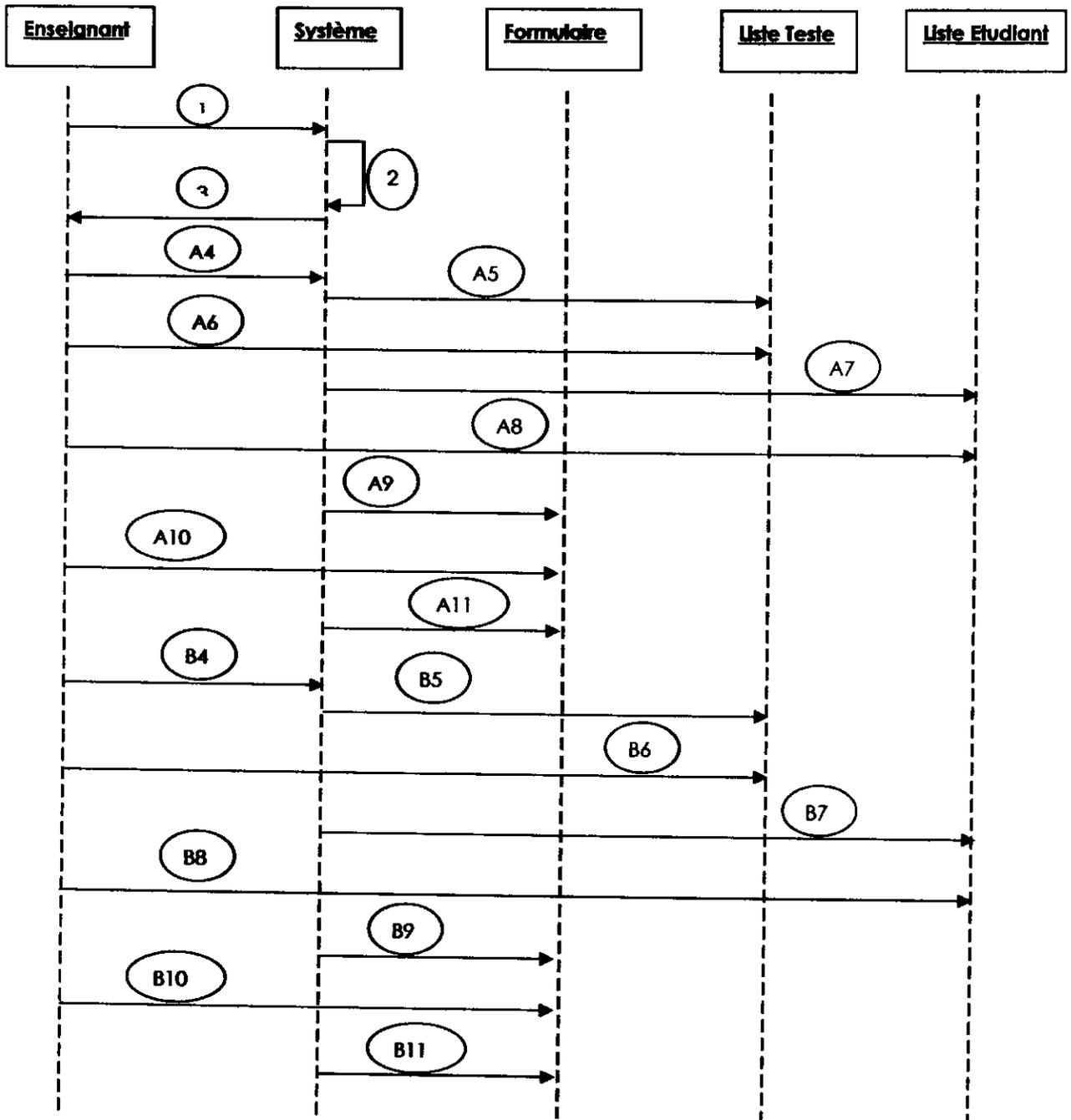
A. Noter les étudiants :

4. l'enseignant demande de noter les étudiants.
5. le système affiche une liste des testes.
6. l'enseignant sélectionne un teste.
7. le système affiche une liste des étudiants.
8. l'enseignant sélectionne un étudiant puis valide.
9. le système affiche un formulaire.
10. l'enseignant saisit les données puis valide.
11. le système contrôle puis enregistre les données.

B. Modifier la note d'un étudiant :

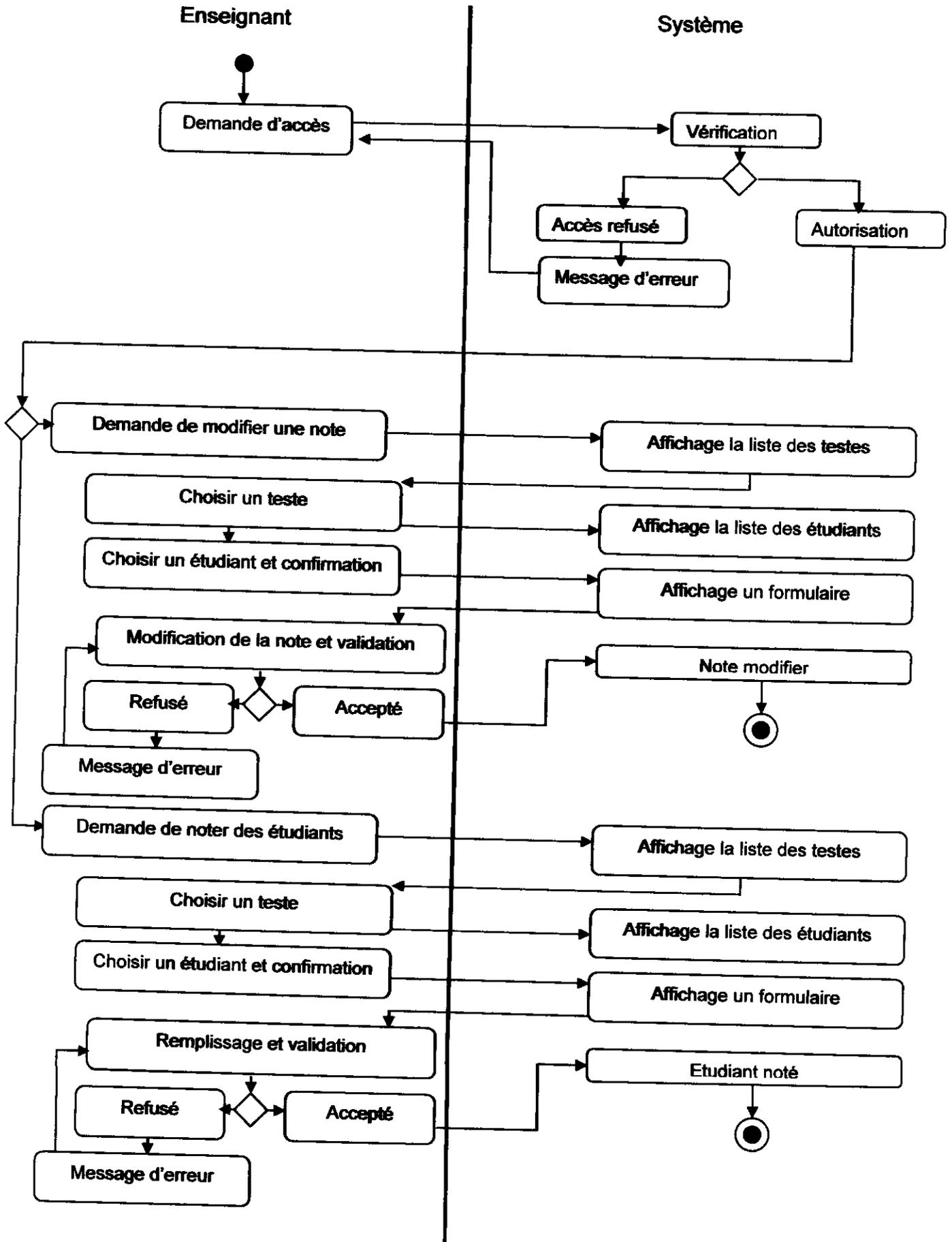
4. l'enseignant demande de modifier la note d'un étudiant.
5. le système affiche une liste des testes.
6. l'enseignant sélectionne un teste puis valide.
7. le système affiche une liste des étudiants.
8. l'enseignant sélectionne un étudiant puis valide.
9. le système affiche un formulaire.
10. l'enseignant modifie la note puis valide.
11. le système contrôle puis enregistre les données.

11.2-Diagramme de séquence :



Handwritten signature or mark.

11.3-Diagramme d'activité :

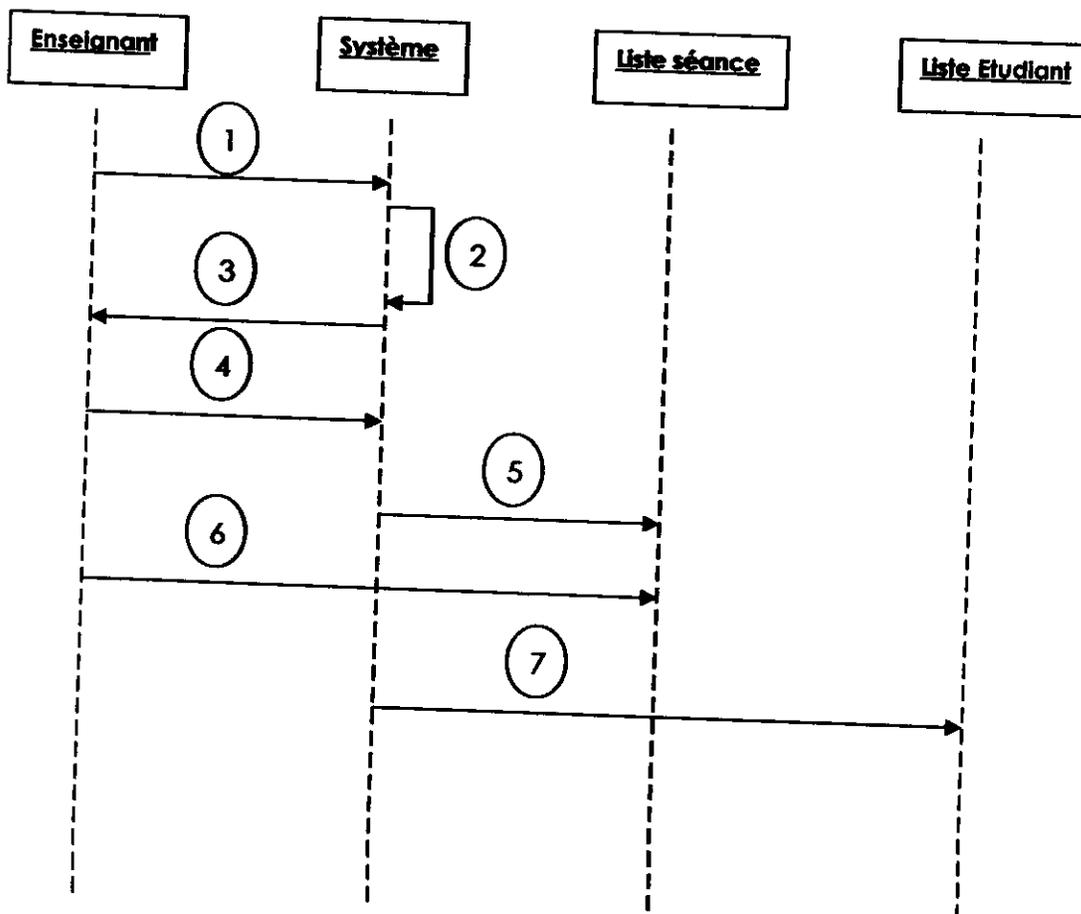


12. Cas d'utilisation << Etablir la liste de présence >> :

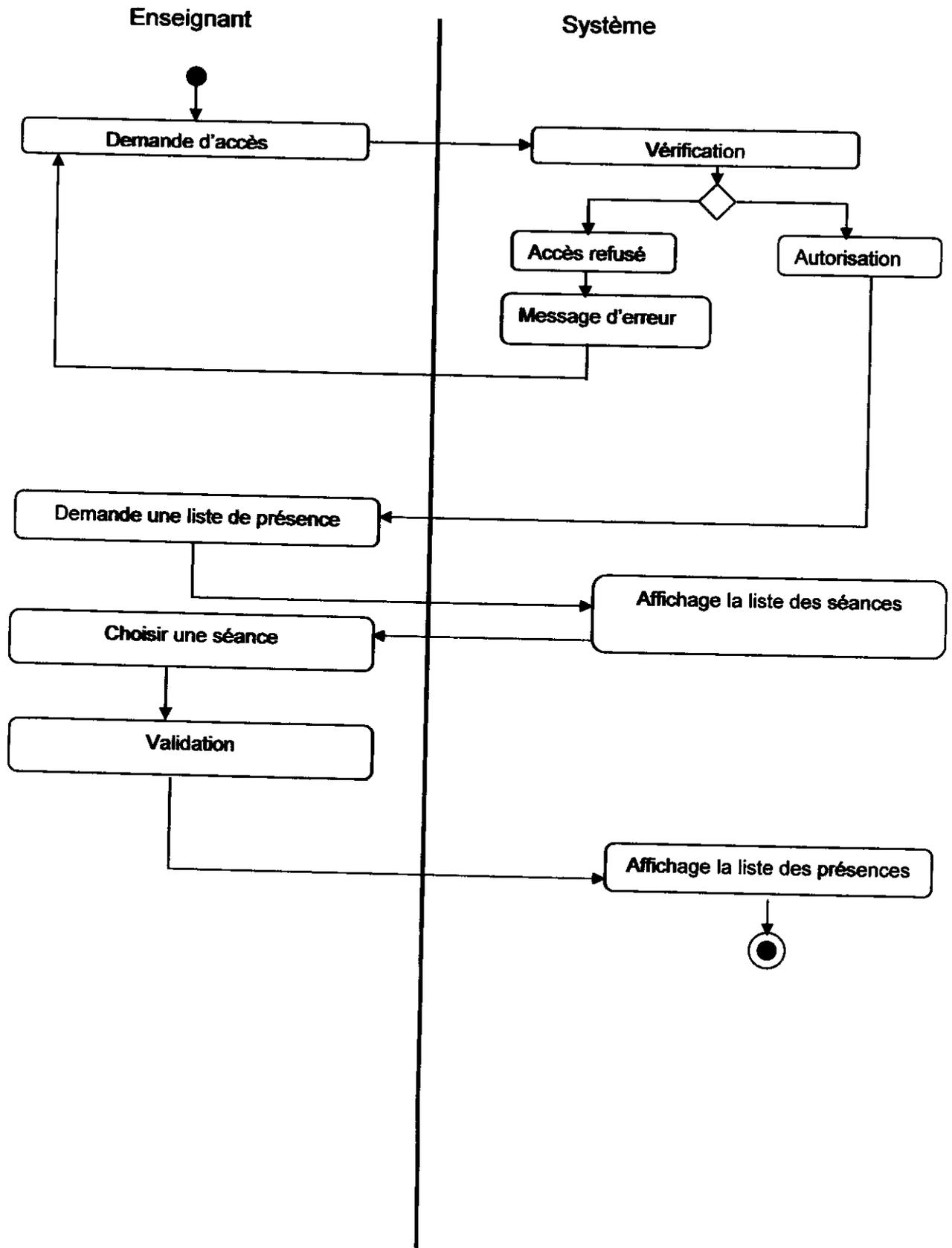
12.1-Scénario :

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 une liste de présence.
5. le système affiche une liste des séances.
6. l'enseignant sélectionne une séance.
7. le système affiche une liste d'étudiants.

12.2-Diagramme de séquence :



12.3-Diagramme d'activité :



13. Cas d'utilisation << Redémarrer, Arrêter et Verrouiller des postes >> :**13.1-Scénario :**

1. l'enseignant envoie ses paramètres d'identification.
2. le système vérifie cette identité.
3. le système donne l'accès à l'enseignant.

A. Redémarrer des postes :

4. l'enseignant demande de redémarrer des postes.
5. le système affiche une liste des postes.
6. l'enseignant sélectionne des postes puis valide.
7. le système redémarre les postes.

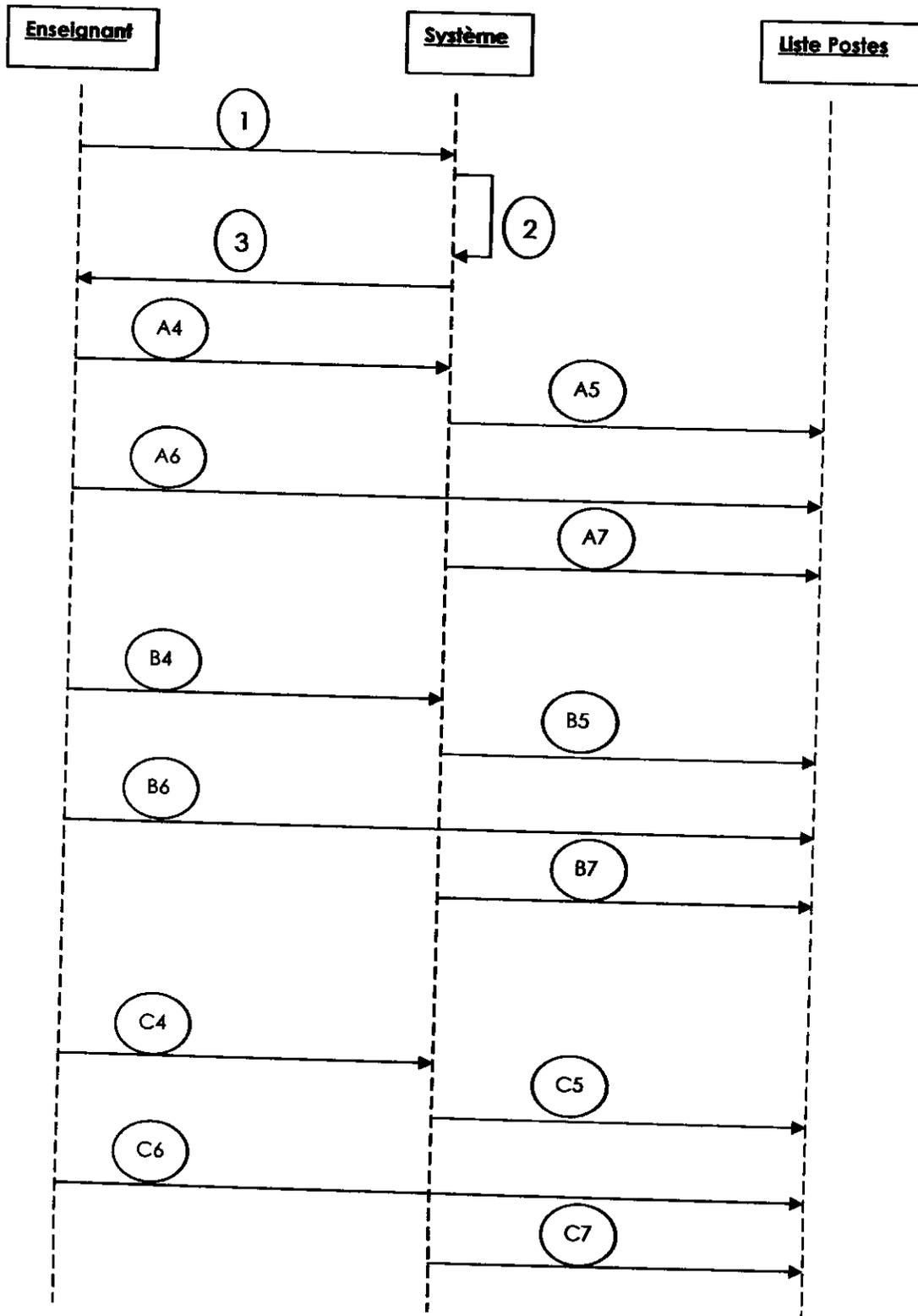
B. Arrêter des postes :

4. l'enseignant demande d'arrêter des postes.
5. le système affiche une liste des postes.
6. l'enseignant sélectionne des postes puis valide.
7. le système arrête les postes.

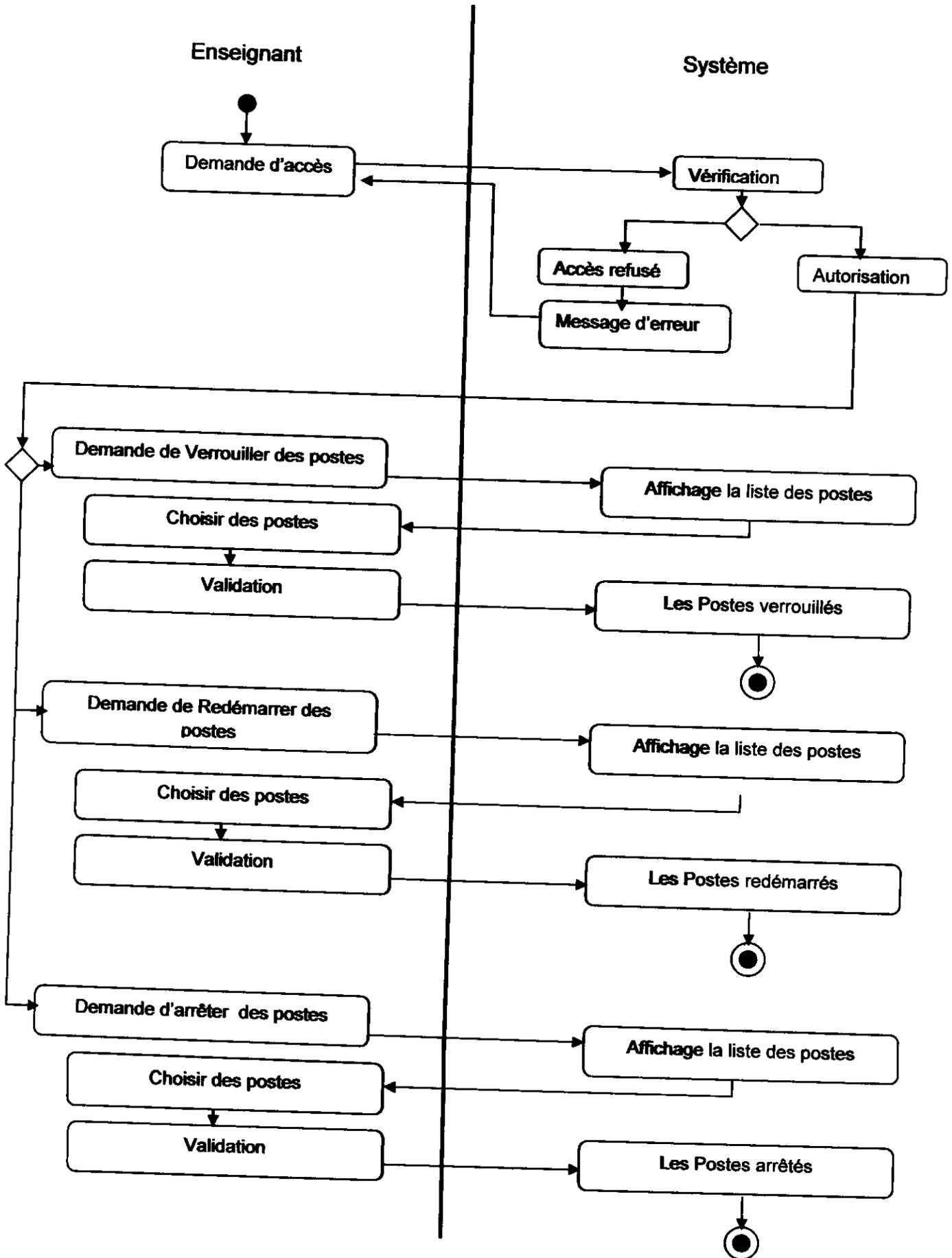
C. Verrouiller des postes :

4. l'enseignant demande de verrouiller des postes.
5. le système affiche une liste des postes.
6. l'enseignant sélectionne des postes puis valide.
7. le système verrouille e les postes.

13.2-Diagramme de séquence :



13.3-Diagramme d'activité :

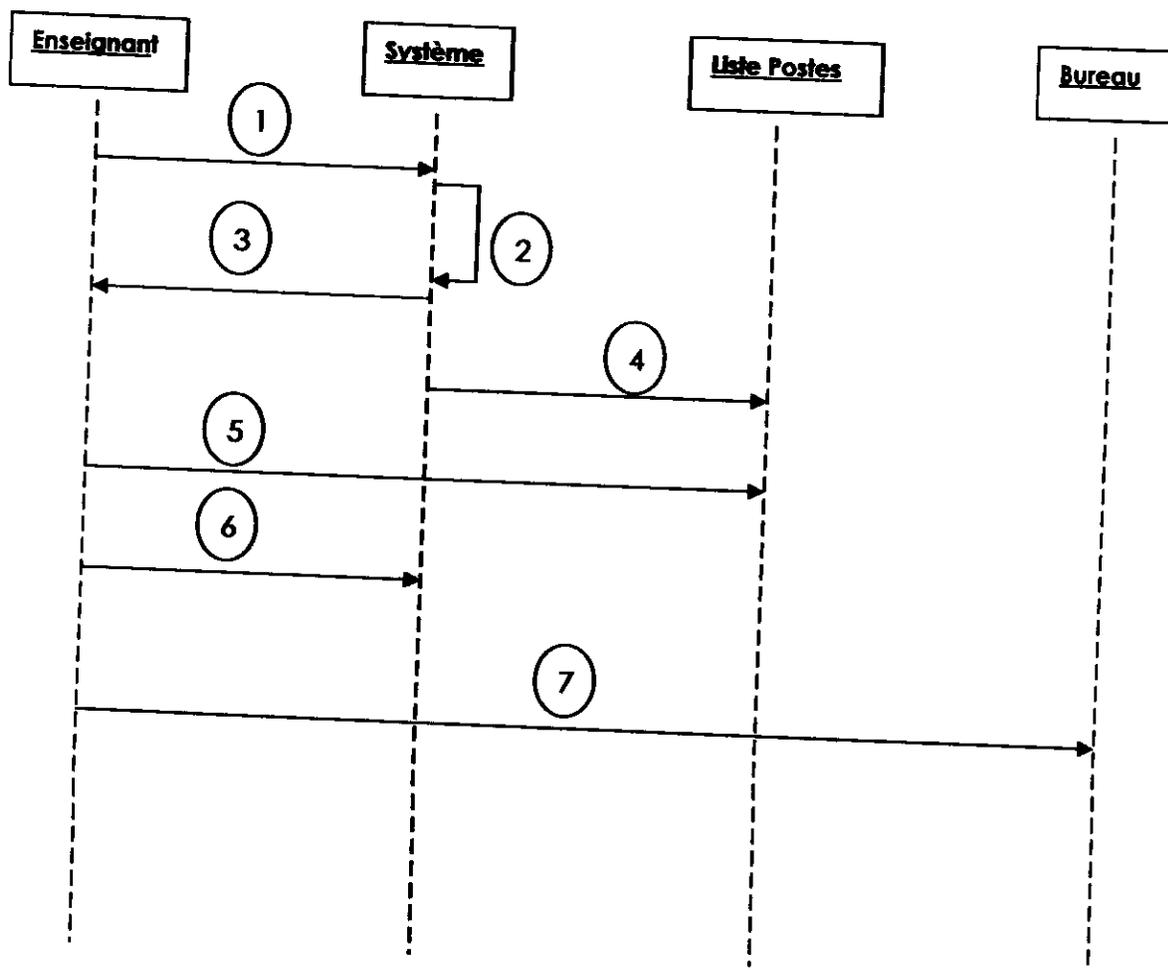


14. Cas d'utilisation << contrôler le bureau d'étudiant >> :

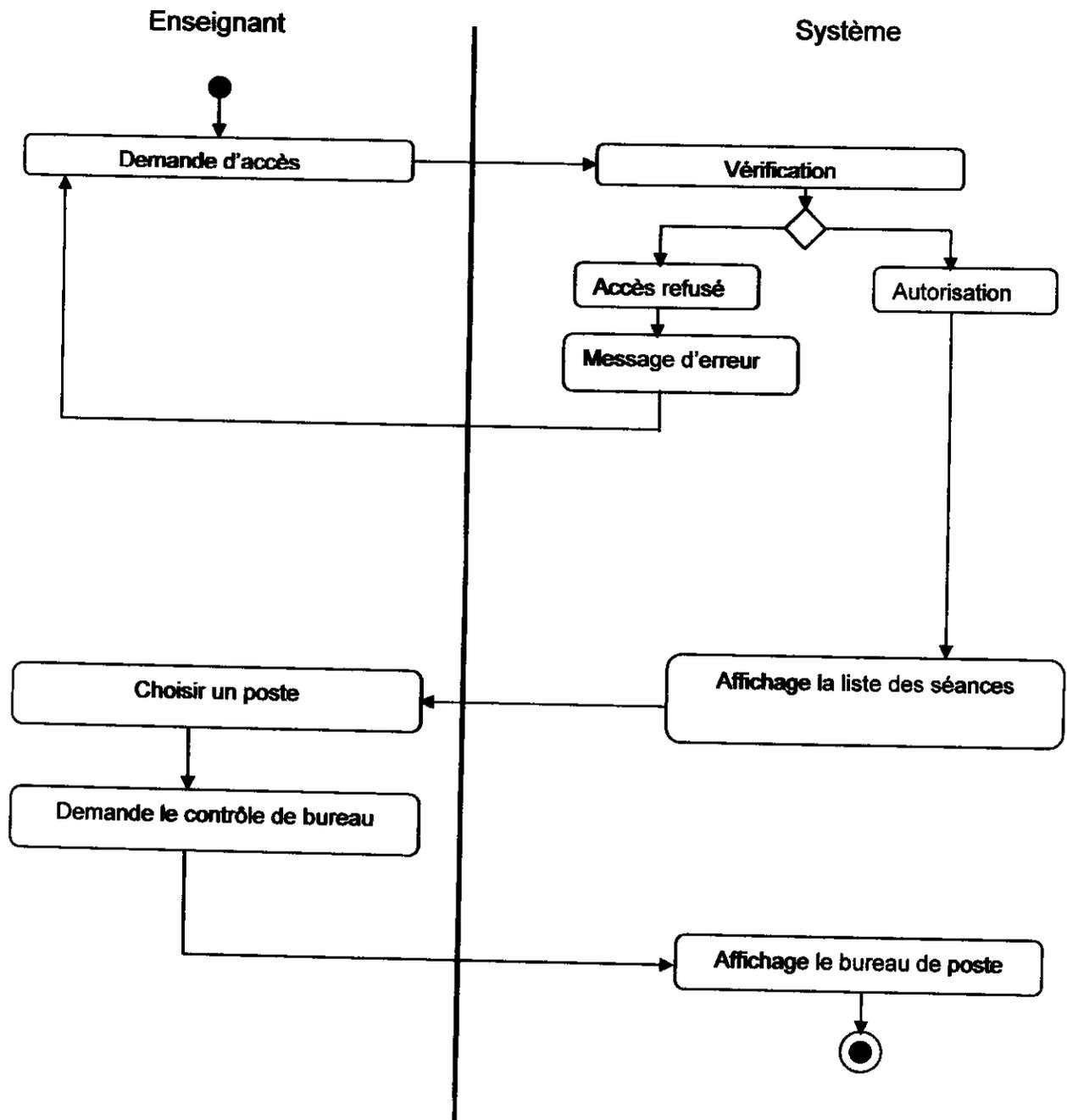
14.1-Scénario :

2. l'enseignant envoie ses paramètres d'identification.
3. le système vérifie cette identité.
4. le système donne l'accès à l'enseignant.
5. le système affiche une liste des postes.
6. l'enseignant sélectionne un poste.
7. l'enseignant demande le contrôle de bureau.
8. le système affiche le bureau.

14.2-Diagramme de séquence :



14.3-Diagramme d'activité :



15. Cas d'utilisation << Gérer la boîte de réception >> :

15.1-Scénario :

- 1- l'enseignant envoie ses paramètres d'identification
- 2- le système vérifie cette identité
- 3- le système donne l'accès à l'enseignant

A .Consultation :

- 4- l'enseignant demande de consulter sa boîte de réception
- 5- le système affiche la liste des fichiers concernant le module en cours

a. Voir un fichier :

6. l'enseignant sélectionne un fichier puis valider
7. le système vérifie et ouvrir le fichier

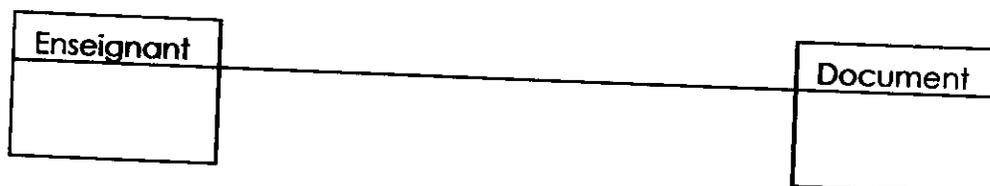
b. copier un fichier :

6. l'enseignant sélectionne un fichier puis valider
7. le système affiche un formulaire
8. l'enseignant saisit le paramètres puis valider
9. le système contrôle les paramètres et transfère le fichier

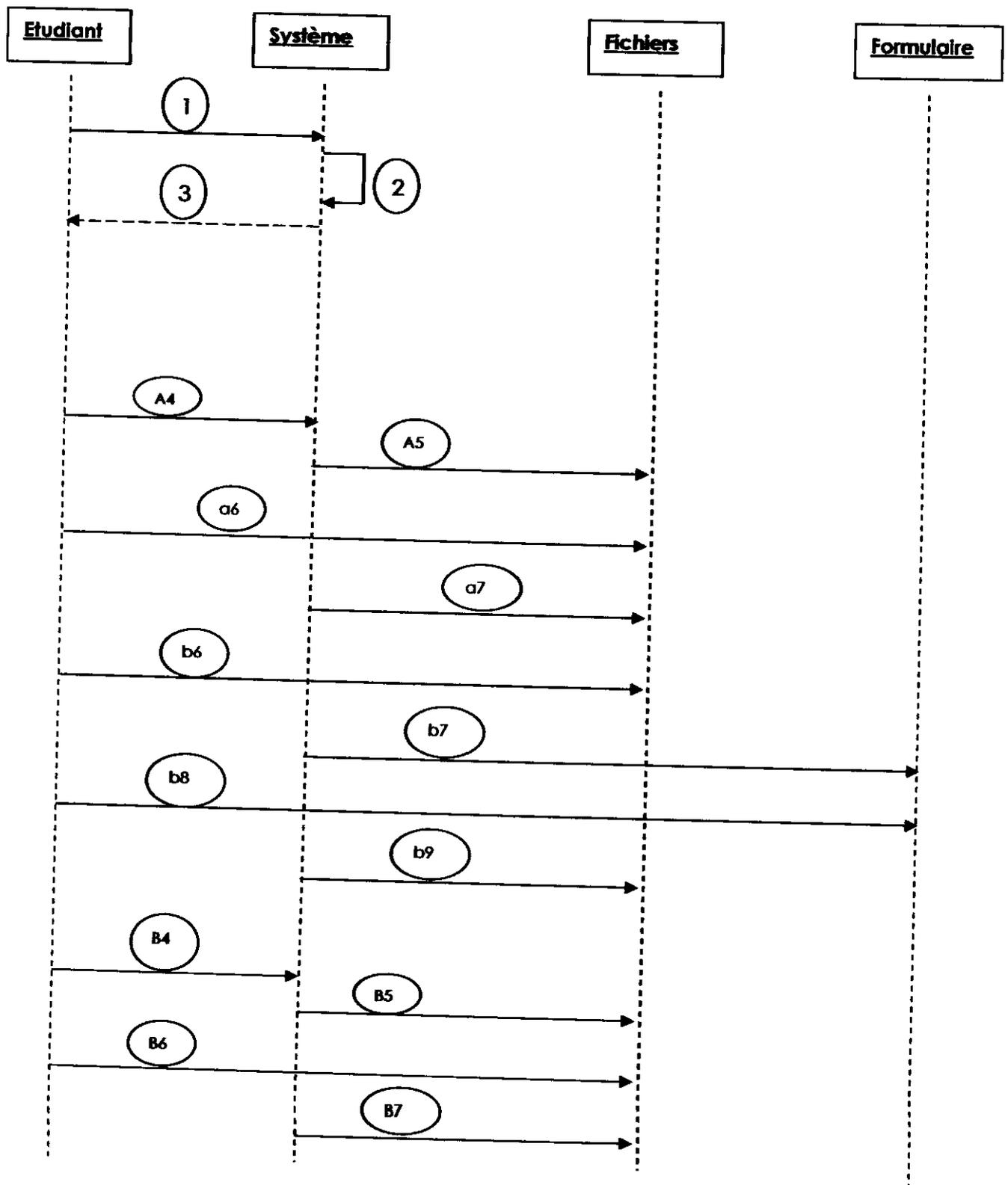
B. Suppression d'un fichier :

4. l'enseignant demande de supprimer un fichier
5. Le système affiche la liste des fichiers concernant le module en cours
6. l'enseignant sélectionne un fichier puis valider
7. Le système supprime le fichier et affiche la nouvelle liste des fichiers

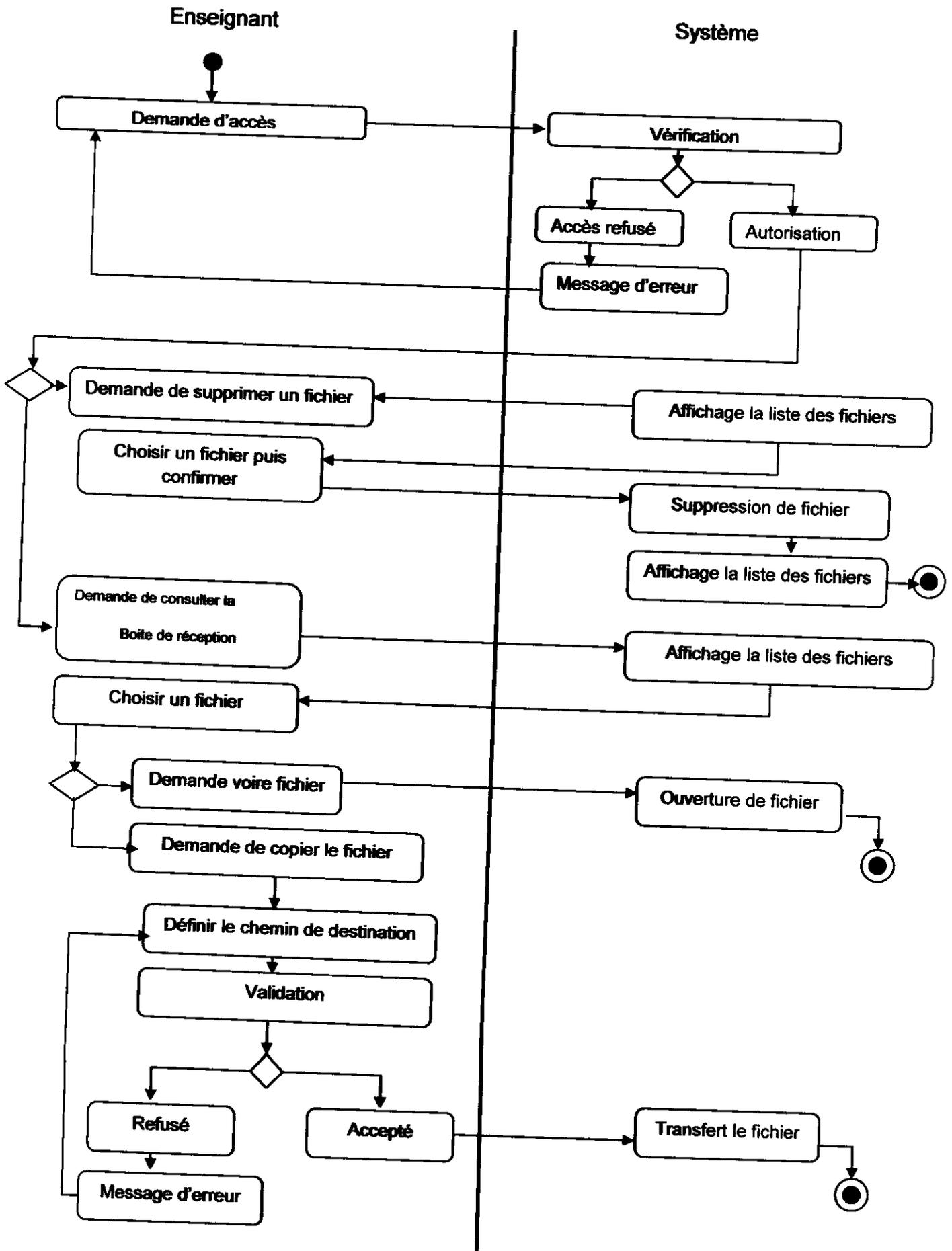
15.2-Diagramme de classe :



15.3-Diagramme de séquence :



15.4-Diagramme d'activité :



V. Diagramme de classe :

1. Diagramme de classe global :

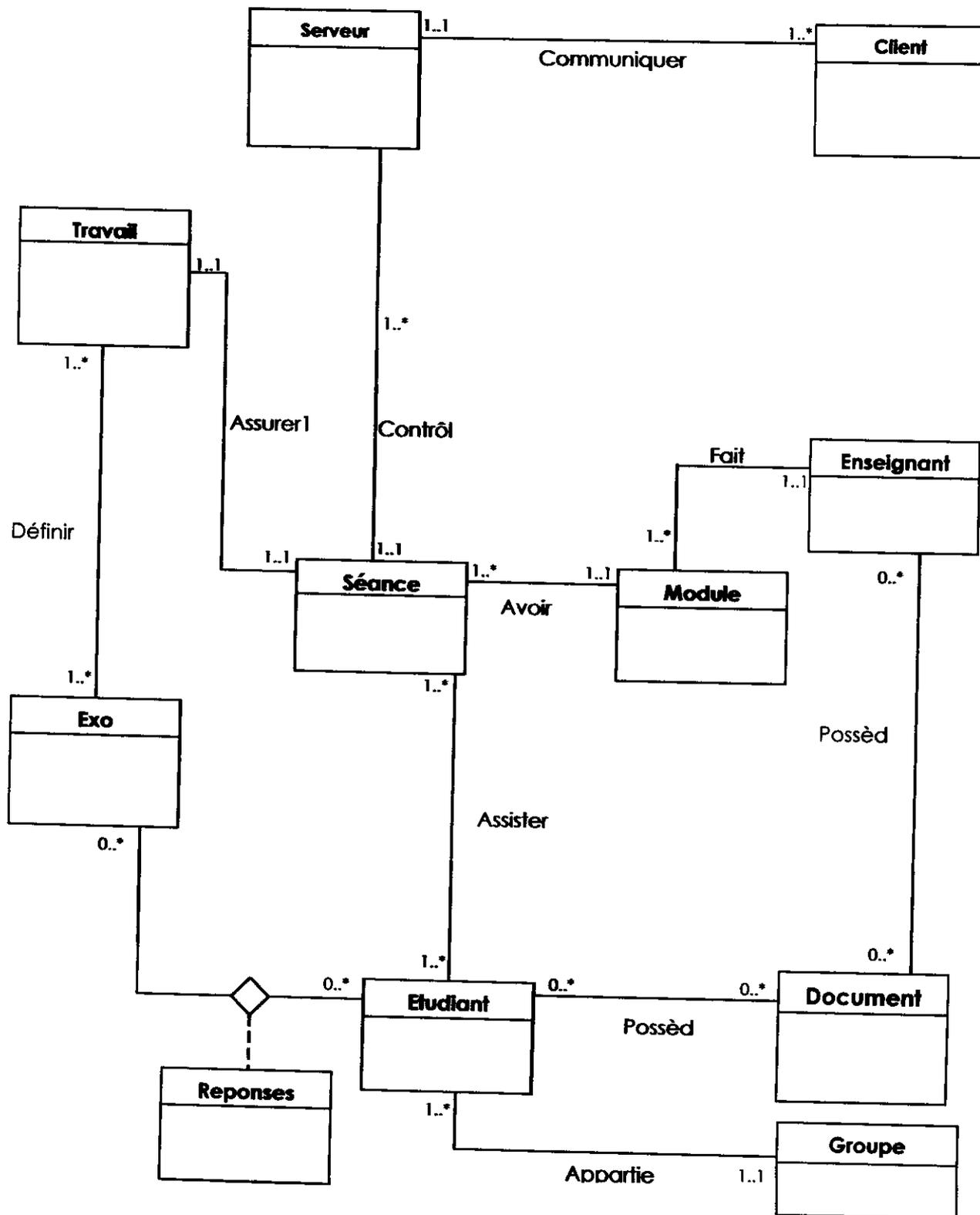


Figure 17 : Diagramme de classe global

2. Description des classes :

Nom Classe	Les Attributs	Description des Attributs	Les Méthodes
Séance	Num_séance Nom_séance Etat_séance Date_séance Heur_debut Durée_séance Code_Module Num_Trav	Numéros séance Nom de la séance Etat de la séance (<i>en attente, démarré ou fermé</i>) La date de la séance Heur début de la séance La durée de la séance Code Module Numéro de travail	Ajouter_sean() Modifier_sean() Supprimer_sean()
Etudiant	Mat_Etud Nom_Etud Pnom_Etud Num_groupe	Matricule d'étudiant Nom de l'étudiant Prénom de l'étudiant Numéros de groupe	Ajouter_etud() Modifier_etud() Supprimer_etud()
Enseignant	Num_Ens Nom_Ens Pnom_Ens Grade_Ens Mot_Pass	Numéros d'enseignant Nom de l'enseignant Prénom de l'enseignant Grade de l'enseignant Mot de passe	Ajouter_ens () Modifier_ens () Supprimer_ens ()
Module	Code_Module Desg_Module Num_Ens	Code module Désignation module Numéros d'enseignant	Ajouter_mod() Modifier_mod() Supprimer_mod()
Travail	Num_Trav Desg_Trav Type_Trav	Numéros Travail Désignation Travail Type Travail (<i>TP, Test</i>)	Ajouter_Trav () Modifier_Trav () Supprimer_Trav()
Exo	Num_exo Enoncé_exo Titre_exo	Numéros exercice Enoncé d'exercice Titre d'exercice	Ajouter_exo() Modifier_exo() Supprimer_exo()
Réponse	Mat_Etud Num_exo Texte_rep Note_rep	Matricule étudiant Numéros exercice Texte de réponse Note réponse	Ajouter_rep() Supprimer_rep()

CHAPITRE IV

Analyse et Conception

Définir	Num_Trav Num_exo	Numéro Travail Numéro Exercice	
Groupe	Num_groupe Niv_groupe Spéc_groupe	Numéros de groupe Niveau d'étudiants de groupe Spécialité de groupe	Ajouter_g() Supprimer_g() Modifier_g()
Document	Num_doc Nom_doc	Numéro document Nom de document	Supprimer_doc()
Possède1	Num_Ens Num_doc Date_recept Heur_recept Chemin_doc	Numéro enseignant Numéro document Date réception de document Heur réception de document Chemin de document	
Possède2	Mat_Etud Num_doc Date_recept Heur_recept Chemin_doc	Matricule étudiant Numéro document Date réception de document Heur réception de document Chemin de document	
Assister	Mat_Etud Num_séance Etat_prés	Matricule étudiant Numéro de la séance Etat de présence de l'étudiant	
Service client	Adr_IP Port	Adresse IP de la machine utilisée Port de connexion	CapterEcran () EnvoyerFile () ControleSouris ()
Service serveur	Adr_IP Port	Adresse IP de la machine serveur Port de connexion	Connexion () ValiderEntrées () ContrôleAccès () CapterEcran () EnvoyerFile () ControleSouris ()

VI. Architecture de Système :

Serveur

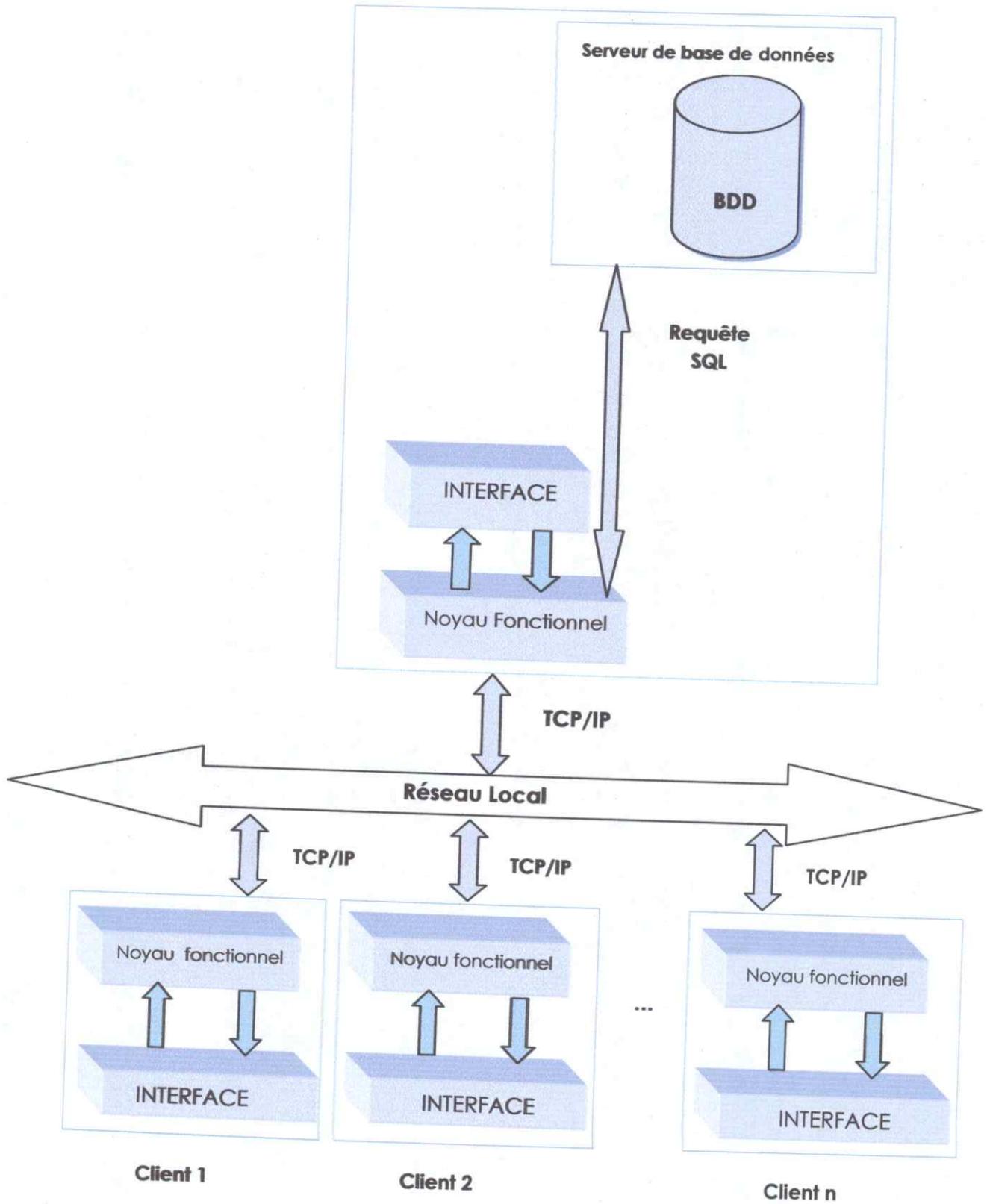


Figure 18 : Architecture de système

Nous avons intéressé dans notre système par deux applications client et serveur donc on a choisi un modèle centralisé basé sur l'architecture client/serveur telle que :

- Dans le niveau serveur :
 1. **Noyau fonctionnel** : contient tous les fonctions nécessaires pour la bonne marche du serveur, il s'agit principalement de contrôle de données introduites au système et l'acheminement des différents taches de base, telle que la connexion avec la base de données, la connexion avec les postes clients, la synchronisation des différents accès du client.
 2. **Interface** : représente l'interaction homme-mahine en mode graphique, il sert à l'affichage des données introduit par l'utilisateur ou fournit par le système.
 3. **serveur de base de données** : manipulation et stockage des donnée viennent de l'extérieur ou bien générées par le noyau fonctionnel.
- Dans le niveau client :
 1. **Noyau fonctionnel** : contient tous les fonctions nécessaires pour la communication avec le serveur et interprète les demandes du client via le réseau.
 2. **Interface** : représente l'interaction de système avec l'utilisateur client en mode graphique, c'est le module qui a la responsabilité d'affichage des données fournit par le noyau ou bien introduit par l'utilisateur client

Conclusion :

La conception nous a permet de bien comprendre le système en détail, de connaître le fonctionnement de chaque module et les liens entre les modules elle-même.

La modélisation UML nous a permet d'éclairer les besoins de notre système et comment satisfaire ces besoin.

Au prochain chapitre nous allons présenter en détail l'implémentation de note futur système.

Chapitre V

Implémentation

&

Réalisation



Dans ce Chapitre :

- Environnement de développement.
- Présentation de l'Application.
- Conclusion

I. Environnement de développement :

I.1 Langage de programmation :

Pour réaliser notre application on a choisi le langage Java puisqu'il est efficace, performant, standard et facile à apprendre (et, de plus, gratuit) et disponible. Il satisfait aux besoins de l'immense majorité des développeurs.

Java est à la fois u langage de programmation informatique orienté objet et un environnement d'exécution informatique portable créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (Cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWord*.

Pourquoi Java ?

Les applications Java peuvent être exécutés sur tous les systèmes d'exploitation pour lesquels a été développé une plat forme Java, dont le nom technique est JRE (Java Runtime Environment – Environnement d'exécution Java). Cette dernière est constituée d'une JVM (Java Virtual Machine – Machine Virtuelle Java), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d'une bibliothèque standard à partir de laquelle doivent être développés tous les programmes en Java. C'est la garantie de portabilité qui a fait la réussite de Java dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes.

On a utilisé exactement les deux environnements « netbeans-6.1-windows » pour générer l'interface et « eclipse » pour programmer et gérer les fonctionnalités de notre système et aussi pour faire la connexion à la base de données. Pour cette dernière on a utilisé le SQL Server 2000.

1- Netbeans :

Netbeans est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, Netbeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend

toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

2. Eclipse :

Eclipse IDE est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de Plug-in (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

La base de cet environnement de développement intégré est l'Eclipse platform qui est composée de :

- Platform Runtime démarrant la plateforme et gérant les plug-ins.
- SWT la bibliothèque de base de l'EDI.
- JFace une bibliothèque graphique de plus haut niveau basé sur SWT
- Eclipse Workbench qui est la dernière couche graphique permettant de manipuler des composants tels que des vues, des éditeurs, des perspectives...

3. SQL Server 2000 :

SQL Server Enterprise Manager est le principal outil administratif de Microsoft® SQL Server™ 2000. Il fournit une interface utilisateur compatible Microsoft Management Console (MMC) – qui permet aux utilisateurs d'effectuer les tâches suivantes :

- Définir des groupes de serveurs exécutant SQL Server.
- Enregistrer des serveurs isolés dans un groupe.

- Configurer toutes les options SQL Server relatives à chaque serveur enregistré.
- Créer et administrer l'ensemble des bases de données, objets, connexions, utilisateurs et autorisations SQL Server concernant chaque serveur enregistré.
- Définir et exécuter toutes les tâches administratives SQL Server sur chaque serveur enregistré.
- Concevoir et tester, de façon interactive, des instructions, lots et scripts SQL par appel de l'Analyseur de requêtes SQL.
- Appeler les divers Assistants définis pour SQL Server.

I.2 Le Modèle MVC :

Pour le développement d'IHM de notre système on a étudié le modèle MVC qui permet de séparer l'interface d'un système de son noyau dans le but de pouvoir modifier l'interface sans toucher au noyau. Cette séparation permettra de migrer l'application facilement vers d'autres environnements.

1. Les origines du modèle MVC

Le Model-View-Controller (MVC) est né en 1980. Au départ créé par Xerox PARC pour Smalltalk (langage orienté objet), il a été depuis emprunté par les réalisateurs de Apple Lisa et Macintosh. Le modèle MVC constitue effectivement un élément très utile au développeur quel que soit le langage utilisé. De plus en plus apprécié des programmeurs Colfusion ou php, il a été récemment recommandé comme modèle pour plate-forme J2EE de SUN.

2. Qu'est ce qu'une architecture MVC ?

L'architecture *Modèle/Vue/Contrôleur* (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le *modèle*, la *vue* et le *contrôleur* ayant chacun un rôle précis dans l'interface.

1. les objets " view ", représentation visuelle du " model " :

Les objets " view " sont la vue de l'application, l'interface avec laquelle l'utilisateur communique. Ils ne s'occupent pas de la gestion ou du stockage des données puisque aucun traitement ne doit être effectué dans cette partie mais affichent les résultats provenant des objets " model " et s'assurent que ces données sont correctement affichées. Les objets " view " n'ayant pas de contact direct avec les objets " model " , ils sont notifiés des changements par l'intermédiaire des objets " controller ".

2. les objets " model ", représentation logique de l'interface utilisateur :

Les objets " model " représentent les données de l'application (les bases de données en faisant partie) et définissent la logique de manipulation de ces données. C'est dans cette partie que vont s'effectuer les traitements, on ne s'occupe absolument pas de la mise en forme mais bien des données seules. Dans une application respectant les règles du modèle MVC, les données les plus importantes seront encapsulées dans les objets "model".

3. les objets " controller ", gérant l'interaction avec l'utilisateur :

C'est donc ici que va se réaliser l'interaction entre les objets " view " et les objets " model ". En effet, les objets " controller " reçoivent les requêtes utilisateur puis déterminent quelles parties des objets " view " et " model " sont requises. Ils constituent donc l'intermédiaire entre les deux autres types d'objets.

3. Un Schéma du Model-View-Controller :

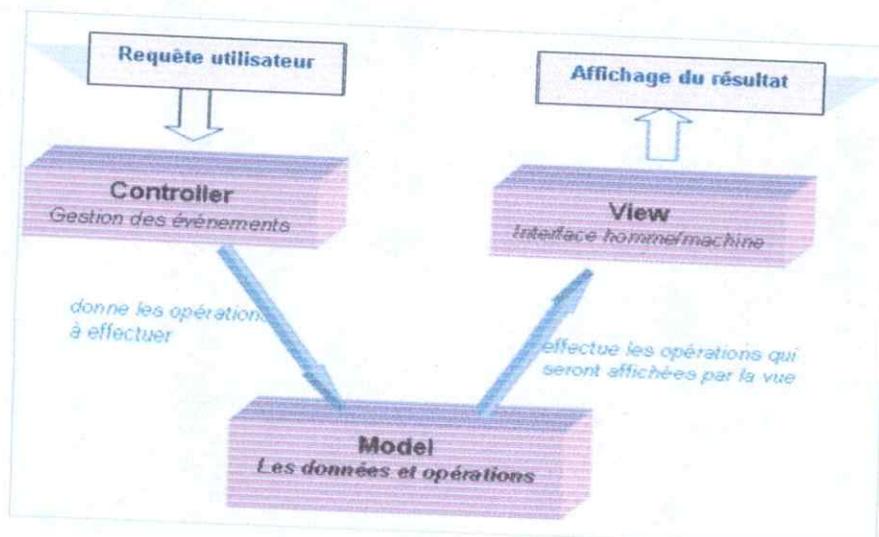


Figure 19 : Schéma du MVC

4. Quels sont ses avantages ?

Le modèle MVC impose donc une séparation totale entre le traitement, l'interface et la communication entre ses deux parties. Cela permet d'avoir non seulement des objets réutilisables pour d'autres applications, mais aussi de pouvoir faire évoluer aisément son programme. Ainsi, si l'on souhaite modifier sa base de données il suffit de revoir son " model " et cela est valable pour le cas où l'on souhaite changer d'interface. Les 3 parties du model MVC sont réellement autonomes. Aucune d'elles ne s'occupent du fonctionnement de l'autre.

5. Quels sont ses inconvénients ?

L'aspect pouvant freiner certains développeurs est que l'utilisation de l'architecture MVC n'est pas des plus simples. Elle nécessite effectivement un travail de conception préalable pouvant être assez fastidieux. De plus, la séparation de l'application en 3 couches implique, d'une part, l'utilisation d'un plus grand nombre de fichiers, et d'autre part un débogage plus difficile. Cependant après avoir géré toutes les exceptions d'un objet, il sera parfaitement réutilisable. De tout ceci résulte que l'architecture MVC est recommandée pour des " grosses " applications. Effectivement, le temps de travail de conception serait trop important pour une " petite " voire " moyenne " application. Ainsi en obligeant le développeur à séparer le modèle de la vue et du contrôleur, cela lui permet de repenser son application et à réfléchir correctement à la structure de son application.

II. Présentation de l'application :

Dans notre application on a deux interfaces une pour le serveur qui est soit le responsable de la salle (Administrateur) ou bien l'enseignant, et l'autre pour les clients qui sont les étudiants de la salle.

1. Interface Serveur (Enseignant – Administrateur) :

Les touches sont activées soit par l'authentification de l'administrateur soit par l'ouverture d'une session par l'enseignant.

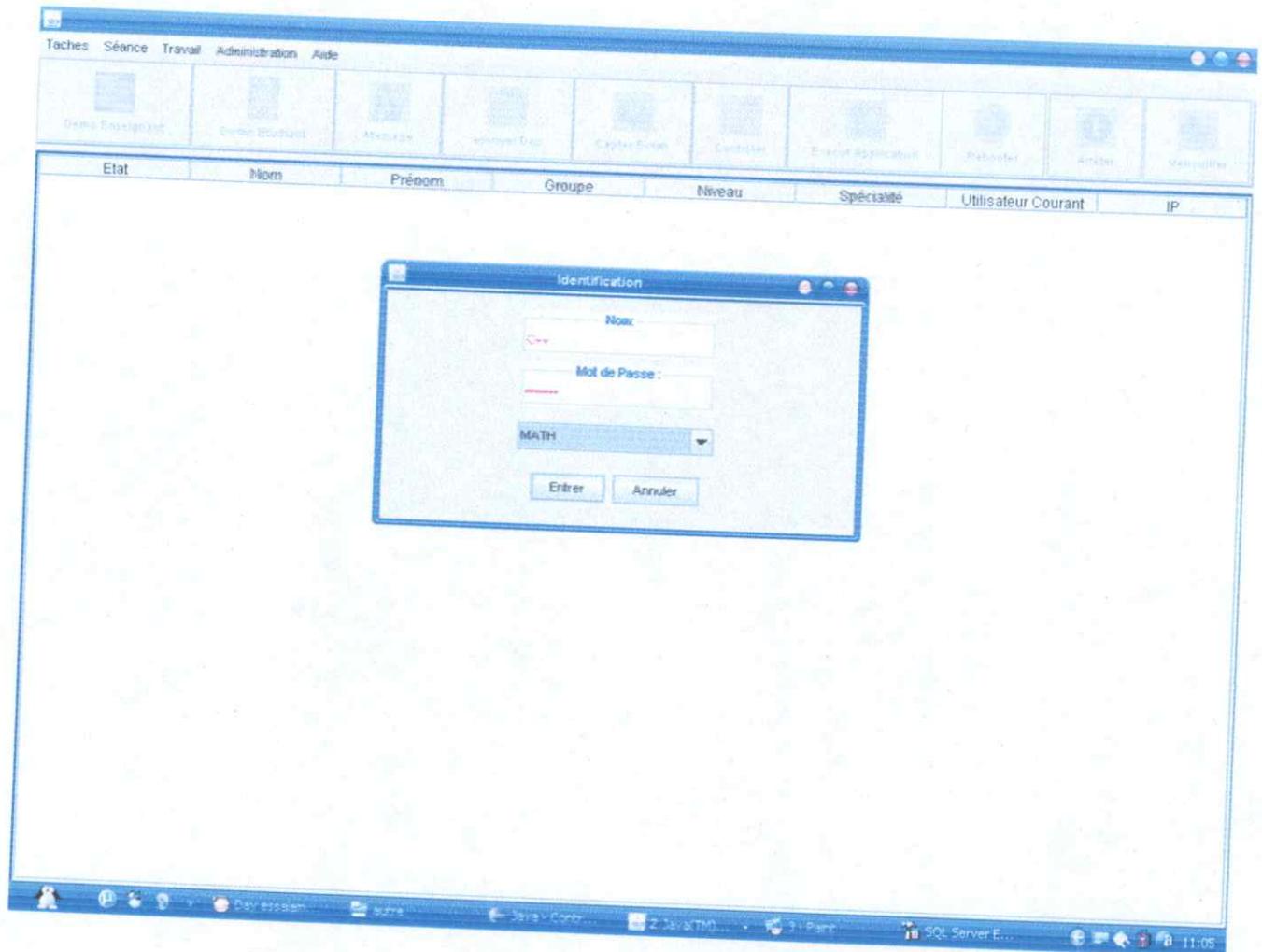


Figure 20 : Interface principale de l'application (Serveur)

La barre de travail devient :



Figure 21 : La barre de travail de serveur

1. Menu :

Taches Séance Travail Administration Aide

2. **Démo Enseignant** : Cette fonction affiche l'écran de l'enseignant sur tous les écrans des étudiants, afin de réaliser une démonstration.
3. **Démo Etudiant** : L'écran de l'étudiant sera affiché sur tous les autres écrans de la salle.
4. **Message** : Envoyer un message à un étudiant sélectionné.
5. **envoyer Doc** : Envoyer un document à un étudiant sélectionné.
6. **Capter Ecran** : Capter l'écran d'un étudiant.
7. **Contrôler** : contrôler à distance le bureau d'un étudiant.
8. **Exécute Application** : Exécuter une application sur un poste étudiant au choix.
9. **Arrêter / Verrouiller / Rebooter** : pour Arrêter, Verrouiller ou Rebooter des postes étudiants.

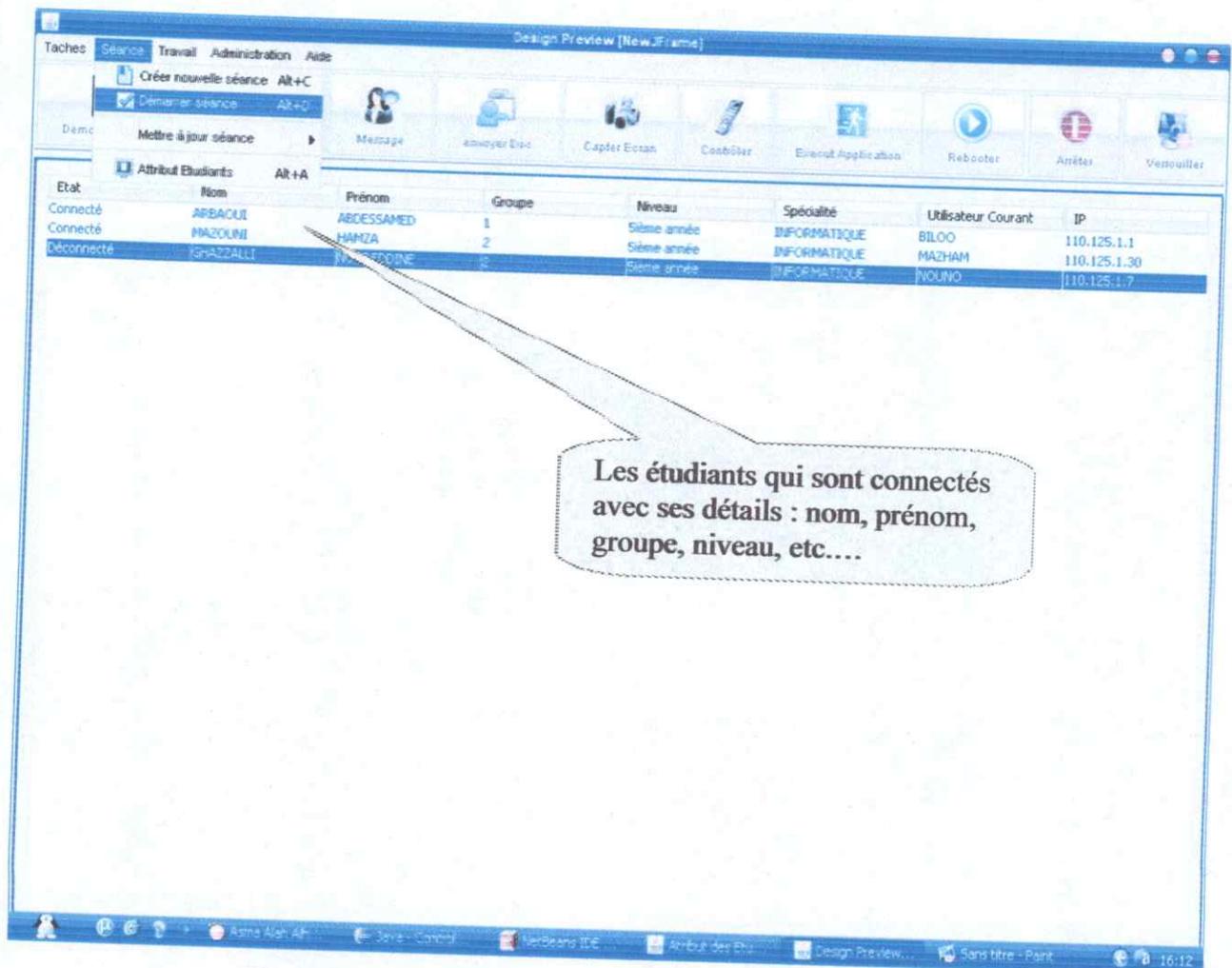


Figure 22 : Affichage des étudiants sur serveur

→ Créer une nouvelle séance : Aller au menu Séance, puis Nouvelle Séance, la fenêtre suivante (figure 23) est affichée. Remplir les données puis OK.

Figure 23 : Créer une nouvelle Séance

→ Pour affecter des étudiants à une séance : Aller au menu Séance, puis Attribut Etudiants, la fenêtre suivante (figure 24) est affichée. Cette dernière contient la liste de tous les étudiants d'un groupe et d'une séance choisit. On sélectionne les étudiants attribués puis valider

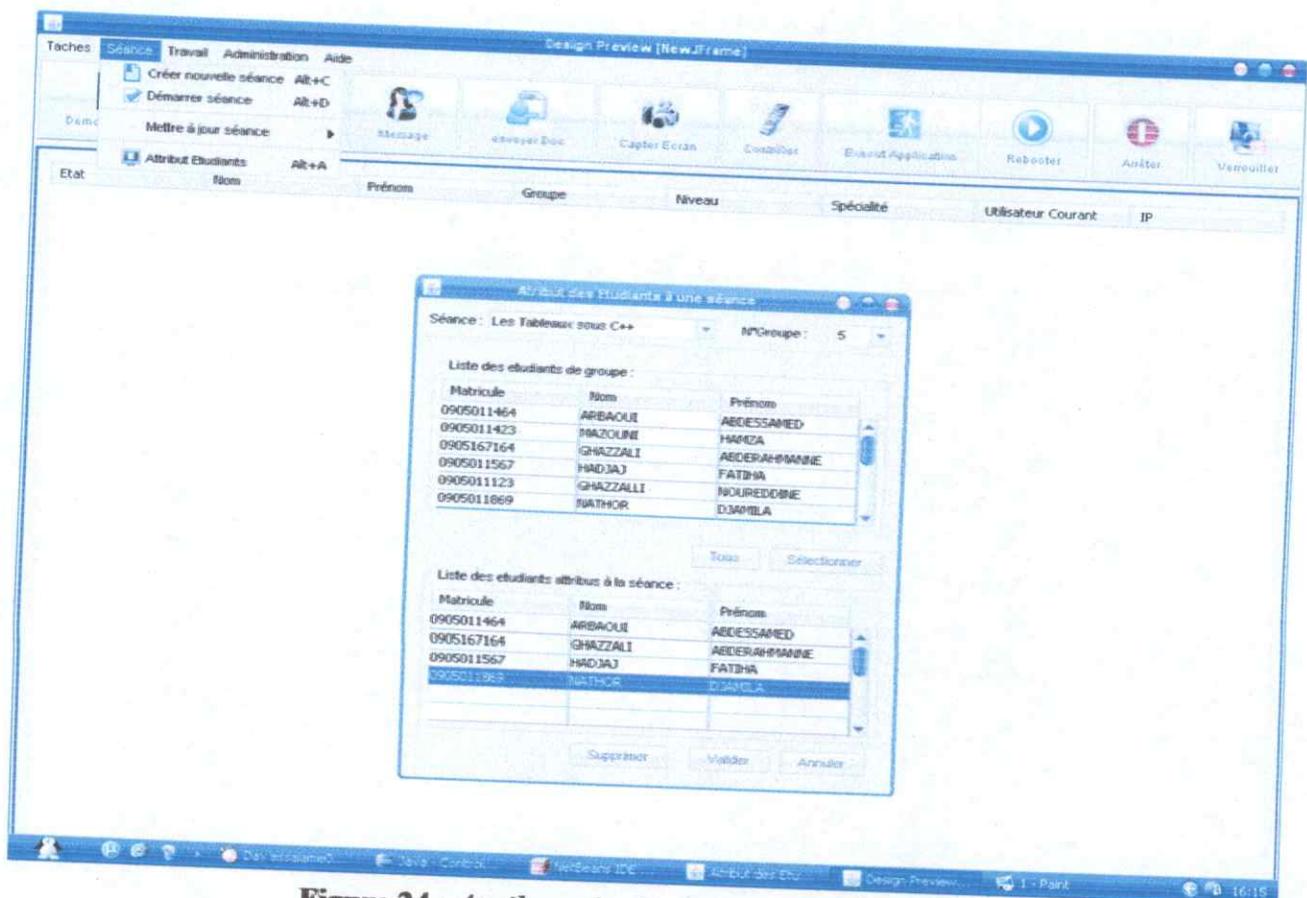


Figure 24 : Attribuer des étudiants à une séance

→ Pour attribuer un travail à une séance : aller au menu travail puis nouveau travail
 La fenêtre de la (figure 25) est affichée, saisir les données puis ajouter.

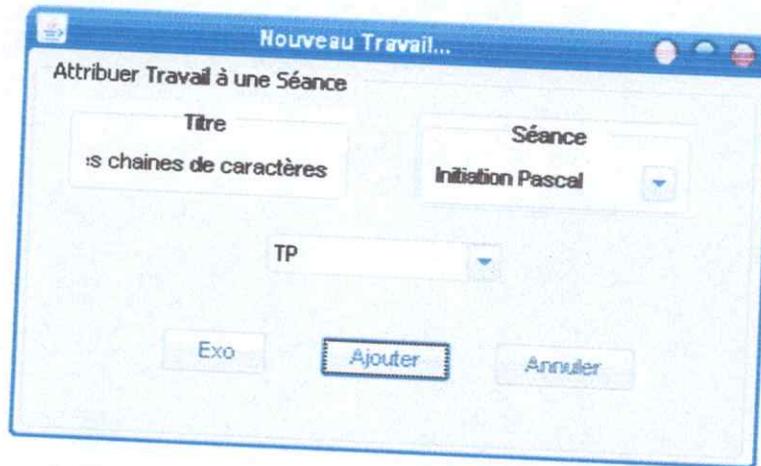


Figure 25 : Attribuer un travail à une séance

La figure 26 pour définir les exercices soit de l'archive ou bien pour ajouter des nouveaux exercices

La figure 27 pour voir les corrections des exercices et les notes de chaque exo.

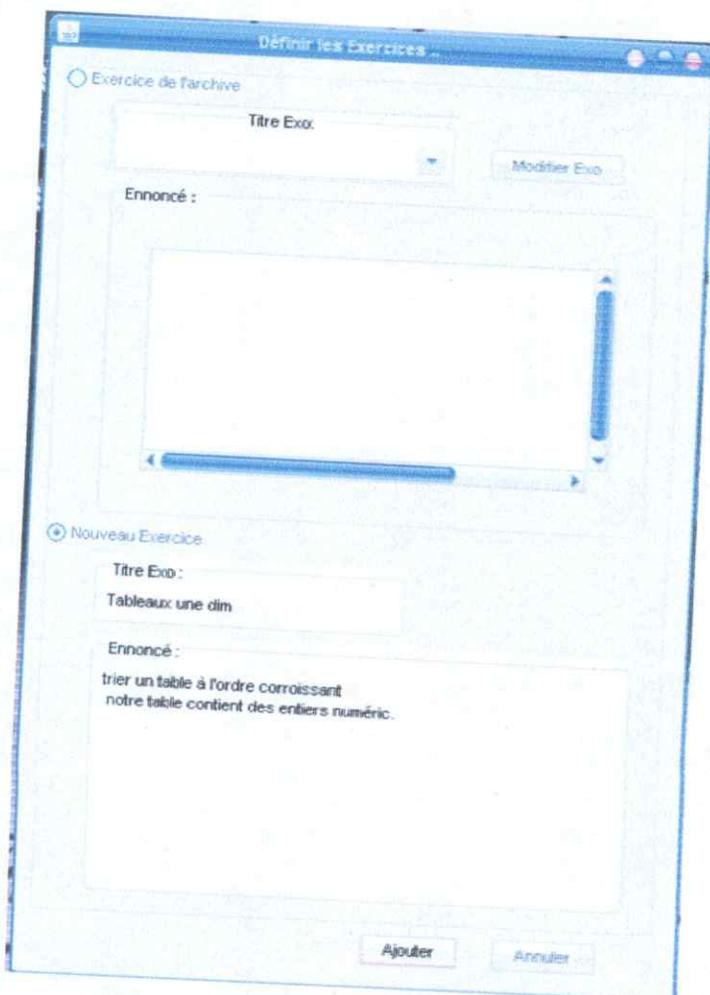


Figure 26 : Définir les exercices

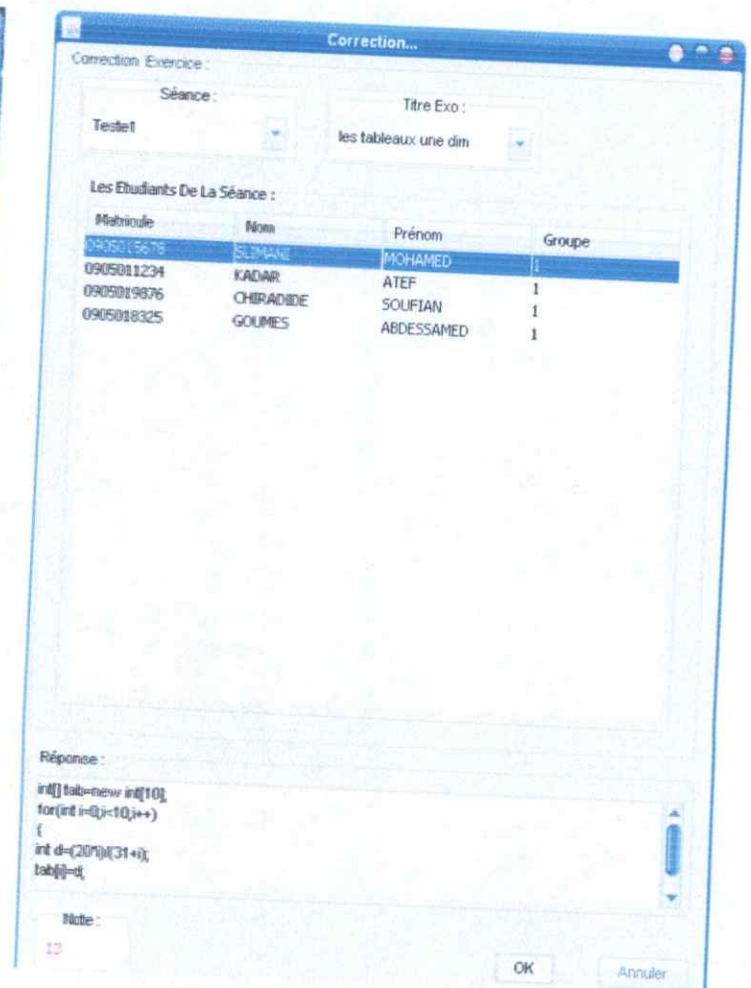
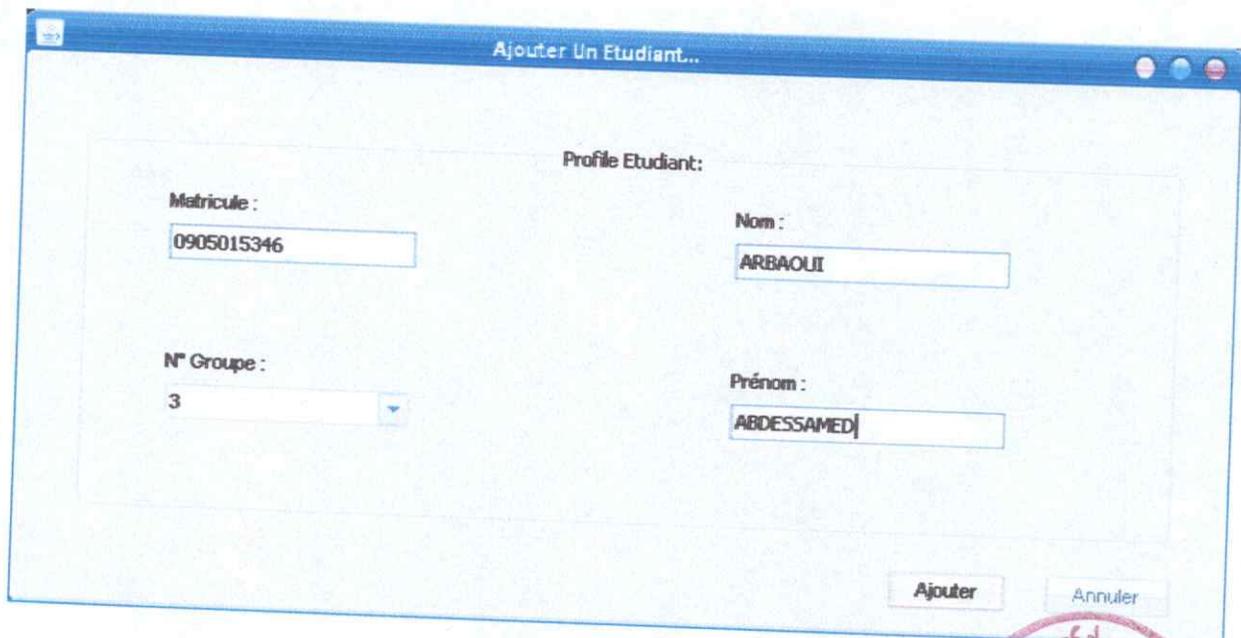


Figure 27 : Réponse d'un exercice

→ L'Administrateur a le droit de modifier, ajouter et supprimer des étudiants, des enseignants et des modules, et voila quelques exemples de ces opérations

**** Ajouter Etudiant**



Ajouter Un Etudiant...

Profile Etudiant:

Matricule : 0905015346

Nom : ARBAOUI

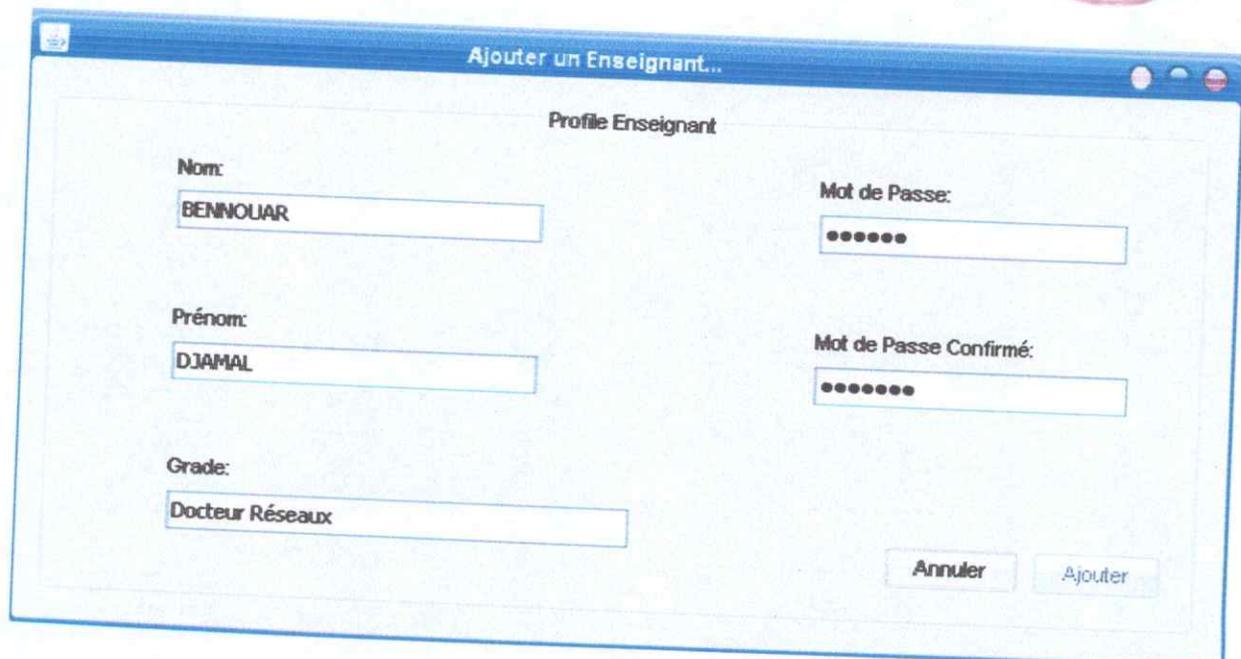
N° Groupe : 3

Prénom : ABDESSAMED

Ajouter Annuler

Figure 28 : Ajouter étudiant

**** Ajouter Etudiant**



Ajouter un Enseignant...

Profile Enseignant

Nom: BENNOUAR

Prénom: DJAMAL

Grade: Docteur Réseaux

Mot de Passe:

Mot de Passe Confirmé:

Annuler Ajouter

Figure 29 : Ajouter enseignant

→ Pour envoyer et recevoir un message à un étudiant :

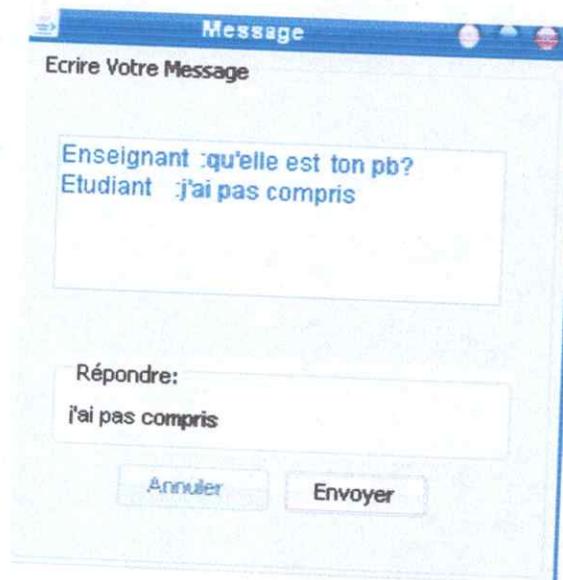


Figure 30 : Envoyer et recevoir des messages

→ Gestion des documents : on peut supprimer, copier ou voir des documents

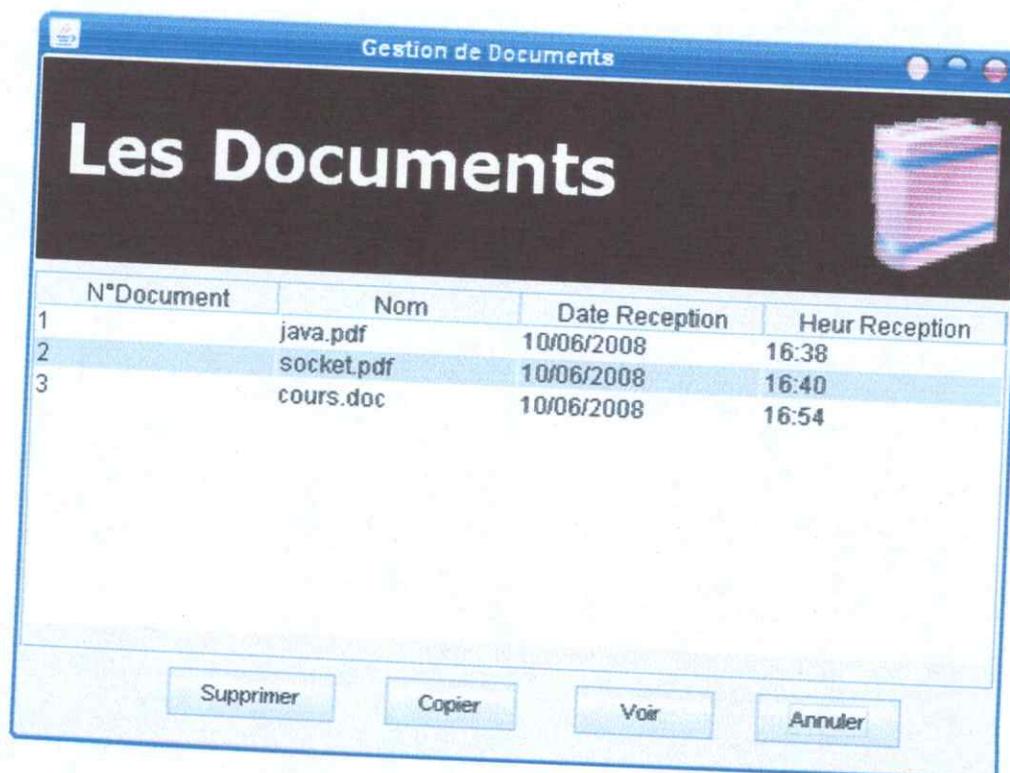


Figure 31 : Gestion des documents

2. Interface Client (Etudiant) :

Les touches sont activées après l'identification pour connecter au serveur

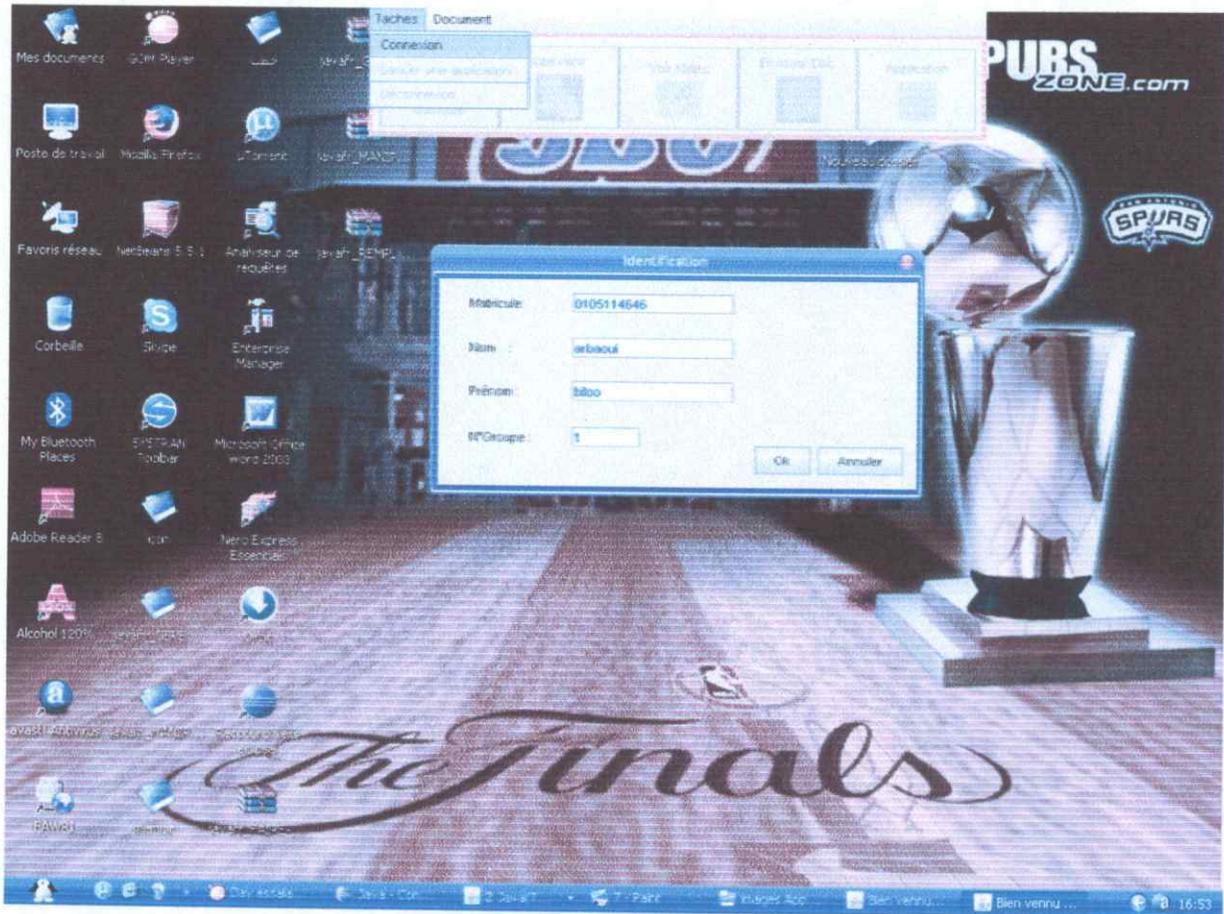


Figure 33 : Interface de l'étudiant avant la connexion

La barre de travail d'étudiant :

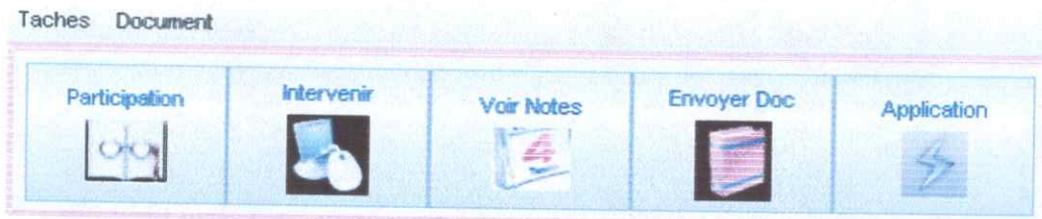


Figure 34 : la barre de travail de l'étudiant

1. **Menu :** Taches Document
2. **Participation :** participer à la séance en cours.
3. **Intervenir :** demande d'intervention pour poser une question.
4. **Voir Notes :** permet à l'étudiant de voir ses notes de TP ou de Test.
5. **Envoyer Doc :** envoyer un document.
6. **Application :** exécuter une application autorisée par l'enseignant

Conclusion :

Dans ce chapitre nous, avons décrit l'environnement de développement c'est-à-dire le langage de programmation qu'on à utilisé en commençant par le langage Java puis les environnements NetBeans et eclipse, puis on a présenté le SQL Server et le modèle MVC qui garantit d'avoir une application d'un niveau réellement élevé de réutilisation du code et d'organisation, et on a terminé par une présentation de notre application.

Conclusion Générale

CONCLUSION GENERALE

Devant la nécessité d'intégration des technologies, à tous les niveaux de l'éducation, de nombreux discours voient dans les activités collaboratives une évolution incontournable, notamment dans les formations universitaires. L'objectif est donc d'augmenter l'efficacité du travail collectif et d'améliorer la production en mettant à profit la technologie comme support pour travailler ensemble à des tâches, simultanément ou non, au même endroit ou dans des lieux différents.

Au niveau des universités, former des étudiants érudits à l'aide d'outils de formation traditionnels est aujourd'hui un défi. Les tableaux et les projecteurs ne conviennent pas aux sujets pour lesquels un exercice pratique représente la meilleure option.

Le travail effectué dans ce projet représente la conception et la réalisation d'un outil pour le contrôle de déroulement d'une séance de TP en informatique. Il nous a permis essentiellement d'approfondir nos connaissances concernant les systèmes collaboratifs, les collecticiels, l'architecture et le fonctionnement des réseaux informatiques. Il nous a permis également d'appliquer les principes et les méthodes théoriques acquises dans notre cursus de formation d'ingénieur, en l'occurrence la POO, la compilation, les bases de données, la programmation système, et aussi de nous familiariser avec un langage de programmation puissant et ouvert, à savoir le Java.

Afin d'améliorer et d'enrichir ce présent travail, nous proposons comme perspective avoir d'autres interfaces destinées à être utilisées sur d'autres dispositifs tels que PDA ou SMART Phone ...etc.

BIBLIOGRAPHIE

1. [DIL 99] DILLENBOURG P. (Edited by) Collaborative-learning: Cognitive and Computational
2. [EGR91] Clarence A. Ellis, Simon J. Gibbs, and Gail L. Rein. Groupware: some issues and experiences. *Comm. ACM*, 34(1):38--58, January 1991.
3. [LEC 99] LECLERCQ D. «Amphithéâtre Electronique Pour Animer De Grands Auditoires Selon Six Paradigmes D'apprentissage/Enseignement » 16 ième colloque de l'association internationale de pédagogie universitaire, Montréal, p:567-578.1999.
4. [II86] : Travail Collaboratif Distribué (M1 GEII et M1 RVSI)
5. [KOC 01] KCHMANNT, «Dewey Contribution To A Standard Problem Based Learning Practice ».proceeding of the first european conference on computer-suported collaborative learning, marche 22-24, maastricht, p: 356-363,2001.
6. [RAT 03] Ratto M, Shapiro R.B, Truong T.M, Griswold W, «The Activeclass Project:
7. Experiment in Encouraging Classroom Participation ».Proceedings CSCL 2003, kluwer, p: 343-352, juin 2003.
8. Courbon J.-C.; Groupware et Intranet; Paris; InterEditions; 1997
9. DESANCTIS, G. ET GALLUPE, B. « Group Decision Support Systems : a new frontier », *Data Base*, vol. 16, No. 2, pp 3-10, 1985.
10. CIBORRA, CLAUDIO U.« Groupware & teamwork : invisible aid or technical hindrance », Claudio Ciborra, 1996.
11. Service informatique Utilisation du logiciel Lanschool Version 6.5 MANON LABRECQUE
12. François Marie Colonna L'architecture Client – Serveur Octobre 2002
13. Client/Serveur à 3 niveaux Études de cas René J. Chevance CNAM Paris 1999-2000.
14. 3-Tier Client/Server at Work Jeri Edwards with Deborah DeVoe John Wiley & Sons (1997).

15. J. Delmas – www.prtice.info
- iTALC - Guide de prise en main rapide et iTALC - Mise en place dans un réseau
16. Architecture Client / Serveur http://www2.ac-lyon.fr/serv_ress/reseau/telechargement/doc_formation/config_client_w95_98.pdf
17. Les protocoles Peer-to-Peer, leur utilisation et leur détection:
<http://2003.jres.org/actes/paper.70.pdf>
18. Protocoles de communication en Peer-to-Peer :
<http://www-adele.imag.fr/~donsez/ujf/easrr0405/p2p/p2p.ppt>
http://www.indexel.net/1_6_2074__3_/3/15/1/Le_Peer_to_Peer__un_modele_totalement_decentralise.htm
http://www.indexel.net/1_20_3852___/Peer-to-peer_pour_les_applications_professionnelles_aussi.htm
19. Etude et utilisation des technologies P2P:
<http://schuler.developpez.com/articles/p2p/>