

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

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



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

Sujet :

**Systeme distribué et hétérogène pour
la gestion de la scolarité**

Présenté par : SOUAD BENSAAD
RYM AMRANI

Promoteur : M. DJAMAL BENNOUAR

MIG -004-174

Résumé

Le sujet de la gestion de la scolarité a été abordé plusieurs fois, étant un sujet important dans le domaine de l'enseignement. L'objectif est de concevoir un Système de Gestion de la Scolarité (SGS) qui facilite aussi bien la tâche de l'administration que celle des enseignants. L'idée a été de réaliser une application web, qui engloberait les principales fonctions de la scolarité et qui serait exploitable par internet. Cette application doit convenir à tout genre d'établissements d'enseignement, du primaire à l'université, et qui doit donc être paramétrable dès l'installation. Elle est destinée à être utilisée par quatre types d'utilisateurs : la direction, les enseignants, les élèves et les parents d'élève. Chacun ayant un espace approprié et sécurisé par un mot de passe.

Mots-clés : gestion de la scolarité, enseignement, internet, ...

ملخص

إن موضوع (إدارة التعليم) تطرق إليه عدة مرات، كونه موضوعا مهما في مجال التعليم. الهدف هو تحقيق (نظام لإدارة التعليم) يقوم بتسهيل كلا من عمل الإدارة وعمل الأساتذة. كانت الفكرة بإنجاز برنامج مبني على الشبكة، يشمل المهام الرئيسية للتعليم و يستعمل عبر الإنترنت. يجب على هذا البرنامج أن يكون ملائما لجميع المؤسسات التعليمية، بدءا بالمدرسة الابتدائية و حتى الجامعة، وأن يكون قابلا للبرمجة عند التثبيت. هذا البرنامج معد للاستعمال من طرف أربعة فئات: الإدارة، الأساتذة، التلاميذ وأولياء التلاميذ، لكل منهم فضاء ملائم محمي بكلمة سر.

الكلمات الأساسية : إدارة التعليم، التعليم، الإنترنت،

Abstract

The topic dealing with school management has been discussed several times, being an important subject in the field of teaching. The objective is to design a School Management System (SMS) which would make both the administration's and the teachers' job easier. The idea has been to create a web application that would include the school main functions and would be processible through Internet. This application must suit all kinds of schools, from elementary schools to universities and hence must be parameterizable as soon as the installation is made. It is designed to be used by four types of users: management administration, teachers, students and their parents. Each has her/his appropriate space secured by a password.

Key-words : school management, teaching, internet...

Remerciements

Nous remercions, premièrement et dernièrement, Allah, le seigneur Tout Puissant, de nous avoir donné la force de terminer ce projet.

Nous tenons à remercier notre promoteur Monsieur DJAMEL BENNOUAR, pour sa précieuse aide et son soutien tout au long de notre travail.

Nous remercions également nos deux chères familles, et spécialement nos mamans, pour leurs encouragements, leur patience et leur compréhension durant toute cette année.

Un merci spécial pour M. AMRANI HATEM, pour toute son aide, ses conseils et ses encouragements depuis le début. Merci d'avoir toujours été à nos côtés.

Un énorme merci pour Mme BENKOUAR AMEL, pour nous avoir soigneusement corrigé ce mémoire, pour tous ses conseils, ses remarques et ses chaleureux encouragements.

Un grand merci à Mlle BOURANANE AICHA pour ses précieux conseils et ses encouragements incessants.

Nous remercions aussi Mlle ACHOUR AMEL pour nous avoir fait profiter de son expérience, ainsi que BENSaad HABIB pour sa compréhensibilité et son encouragement, TEMAZOUT YUCEF, TEMAZOUT FREHA, BEKKAR MADJID, MOUALED MALIKA, MOUALED DJILALI et SAIGHI BOUAOUINA HADJILA pour leur énorme soutien et leur aide.

Sans oublier de remercier tous les professeurs du département d'informatique à l'université de Blida, pour la formation qu'ils nous ont donnée au cours de ces années d'études, sans laquelle nous ne serons pas arrivées à ce résultat.

Enfin, merci à tous ceux qui ont contribué de près ou de loin à ce projet. Soyez-en tous remerciés du fond du cœur.

Table des matières

Résumé	2
Remerciements	3
Introduction	6
Chapitre 1: Présentation du sujet	7
1.1 Objectif	8
1.2 Problématique	10
Chapitre 2: Méthodologie	11
2.1 Constat général	12
2.2 Les méthodes traditionnelles	13
2.3 La méthode proposée	14
Chapitre 3: Analyse des besoins	16
3.1 Diagramme de cas d'utilisation général	18
3.2 Diagramme de cas d'utilisation détaillé	19
3.2.1 L'administrateur du système	19
3.2.2 La direction	20
3.2.3 Les enseignants	23
3.2.4 Les élèves	24
3.2.5 Les parents d'élève	25
Chapitre 4: Conception	26
4.1 Architecture de l'application	28
4.2 Architecture de déploiement	28
4.3 Description des cas d'utilisation	29
4.3.1 Installation du système	29
4.3.2 Authentification des utilisateurs	31
4.3.3 Gestion de scolarité	33

4.3.4	Impression des états	56
4.3.5	Communication	64
4.3.5.1	Affichage du parcours	64
4.3.5.2	Publication	67
4.3.5.3	Messagerie	70
4.4	Diagramme de classe	73
4.5	Description du diagramme de classe	74
Chapitre 5:	Réalisation	76
5.1	Architecture trois-tiers	77
5.2	Environnement de développement	79
5.2.1	Choix du langage de programmation	79
5.2.2	Choix de l'éditeur de site web	79
5.2.3	Choix du SGBD	80
5.3	L'implémentation du diagramme de classes	83
5.4	La mise en œuvre du système	86
5.5	Configuration du système	87
5.6	Les modules de l'application	90
5.6.1	Module « Installation du système »	90
5.6.2	Module « Authentification des utilisateurs »	93
5.6.3	Module « Gestion de scolarité »	98
5.6.5	Module « Impression des états »	105
5.6.6	Module « Communication »	107
Conclusion et perspectives		113
Bibliographie		114

Introduction

L'Internet est devenu un des moyens de communication les plus utilisés dans le monde. Il permet également de partager et d'échanger des données, des programmes, des documents et autres entre utilisateurs du monde entier. La planète devient ainsi un tout petit village..

Le partage est en grande partie l'un des atouts porteurs du net. Il permet ainsi aux gérants des établissements de bénéficier et faire bénéficier les autres des services offerts par l'Internet tout en économisant de l'argent, étant donné qu'il suffit d'un seul serveur pour que tout le monde (selon les privilèges) accède à l'information. Ces techniques apportent beaucoup de réponses et de solutions aux problèmes classiques. En effet, pour accéder à une mise à jour, par exemple, un réseau d'entreprises aurait déplacé une navette humaine pour pouvoir copier dans chaque poste les compléments. Grâce à l'internet, puisque chaque terminal est sensé avoir accès au net, la mise à jour se fait par un simple « clic ». Ceci est sans doute l'intérêt majeur des applications basées sur Internet (web-based applications). Ces applications sont exécutées sur un navigateur Internet. Ainsi aucune installation n'est requise. De plus, une telle application peut être consultée en local (sur la même machine) ou à distance (via internet), ce qui lui donne une meilleure accessibilité tout en économisant du temps et de l'argent.

Grâce à la hiérarchisation des serveurs, il est possible, à partir du serveur central, de faire des copies de bases de données des serveurs auxiliaires, ce qui renforce la sécurité de l'information. En cas de panne de l'un des serveurs, l'information peut être retrouvée.

Le progrès technologique dote Internet d'outils de plus en plus puissants, partageables, mobiles, pourtant moins coûteux, nécessitant moins l'apport humain à grande échelle. Les applications basées sur Internet profitent directement de ces propriétés.

Chapitre1

Présentation du sujet

Présentation du sujet

1.1 Objectif

Notre sujet a pour objectif de réaliser un système de gestion de la scolarité pour toutes sortes d'établissements scolaires. Des systèmes de ce genre ont été réalisés auparavant. Cependant, il leur manquait d'autres aspects nécessaires. Ce que nous allons faire dans ce projet, en plus de la réalisation du système gestion de scolarité de base, nous allons lui ajouter les aspects manquants et que nous avons trouvés importants pour ce genre de système. Ainsi nous allons :

- Améliorer la saisie par l'ajout de photos identifiant l'étudiant, les enseignants et d'autres personnes reconnues par le système.

- Introduire les niveaux de hiérarchie entre utilisateurs reflétant la structure de l'institution et les droits affectés à chacun dans cette institution.

- Notion de blocage des modifications : le haut niveau de la hiérarchie peut bloquer la hiérarchie inférieure pour la modification d'une note.

- Le généraliser pour qu'il puisse s'adapter, dès l'installation à toute sorte de scolarité, du primaire à l'universitaire. Ce logiciel sera ainsi indépendant du type de l'institution (paramétrage à outrance, dont la quasi-totalité serait définie à l'installation).

- L'étendre par les caractéristiques suivantes :

- La distribution : celle-ci doit permettre de faire communiquer, via internet, un nombre important de systèmes installés. La distribution se fait selon deux schémas :

Le premier est un schéma hiérarchique reflétant l'organisation de l'institution. Ainsi, à titre d'exemple, pour une université nous aurons le serveur central de l'université, les serveurs des facultés et les serveurs des départements. Dans ce cas, la distribution consiste à répliquer les données locales pour chaque niveau afin de rendre les traitements plus rapides et efficaces et de permettre aussi d'assurer de manière indirecte une sécurité accrue de l'information.

Dans cette organisation deux types de serveurs sont observables. Les serveurs maîtres et les serveurs esclaves. Chaque serveurs est maître dans la hiérarchie ou il est le sommet (chaque nœud de l'arbre représentant la hiérarchie des serveurs est maître dans le contexte de son sous arbre. Ce même serveur est un serveur esclave de son ascendant propre si ce dernier existe.

Le deuxième schéma est un schéma de serveurs, représentant chacun une institution indépendante de l'autre. Ainsi, deux écoles sont considérées comme institutions indépendantes. De même, deux facultés, dans le contexte d'une même université, peuvent être regardées comme institutions indépendantes.

- La détermination du statut de chaque serveur (autonome, maître, esclave ou offline) se fait à l'installation du système. Un serveur esclave doit obligatoirement s'enregistrer au niveau d'un serveur maître.

- La détermination du parcours pédagogique complet d'un étudiant, du primaire à l'Université. Le problème qui se posera alors serait d'identifier cet étudiant de manière unique à travers le réseau et être capable de le retrouver. Chaque système appartient à une administration bien précise (école primaire, Moyen, lycée, université), et dans chaque système l'étudiant doit posséder une identification locale et une identification globale. Cette identification devrait permettre d'optimiser le temps de recherche par la détermination du serveur exact à interroger et qui posséderait l'information voulue. A titre d'exemple, dans l'identification pourrait apparaître le code de l'institution ou de toutes les institutions par lesquels il est passé.

- La coordination des réseaux de serveurs : La codification d'institutions impose la présence d'une autorité centrale qui affecte ces codes aux institutions. Ainsi à l'installation de chaque copie du logiciel un contact avec un serveur central serait nécessaire pour intégrer la copie au système.

- La sécurité : Une politique d'authentification entre serveurs maîtres et esclaves doit être définie. Pour des raisons de sécurité, lorsqu'un esclave se connecte à son maître, un système d'authentification serait mis en œuvre par le maître pour les futures communications. Il est clair que les

informations de sécurité des échanges peuvent être renouvelées de temps en temps (paramètre fixé à l'installation du maître). Un autre aspect lié à la sécurité concerne les informations en base de données. Celles-ci doivent être cryptées en base de données à base d'une clé définie par la source d'information (Enseignant).

1.2 Problématique

De nombreuses raisons ont rendu nécessaire l'automatisation de la gestion de la scolarité. Les plus importantes sont :

- La perte de temps liée à la lenteur des tâches traditionnelles effectuées manuellement, comme :
 - Calculer les moyennes.
 - Remplir les relevés de notes.
 - Etablir les attestations et les certificats de scolarité.
- Les erreurs faites lors du calcul manuel des moyennes.
- Imposer aux enseignants de se déplacer pour rendre les notes et aux élèves pour consulter leurs notes.
- Les retards pour donner les relevés de notes et les certificats de scolarité aux élèves.
- Difficulté de faire la recherche dans des archives en papier.
- La difficulté de recueillir toutes les informations qui concernent l'élève et de savoir tous les établissements par lesquels il est passé.

Chapitre2

Méthodologie

Méthodologie

Le choix de la méthodologie est une étape primordiale pour le développement d'un projet. Ce choix dépend directement de la nature du projet, de ses objectifs et surtout des contraintes qu'il faut respecter. Nous allons ci-dessous faire un constat général du projet.

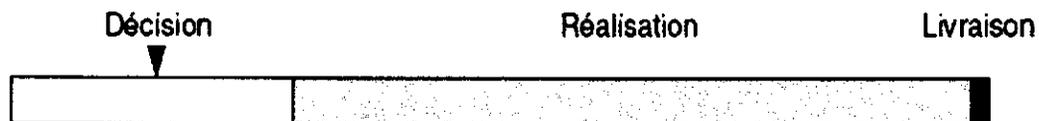
2.1 Constat général

L'objectif est un logiciel local, qui est :

- destiné à être réalisé par deux étudiantes sans aucune expérience,
- non destiné obligatoirement à être terminé par un autre groupe,
- sans contrainte budgétaire,
- avec contrainte de la nécessité de mise en place, dans les plus brefs délais, d'un prototype fonctionnel,
- avec un niveau d'étudiants en termes de maîtrise des outils de programmation,
- avec une petite expérience en termes de programmation Web,
- nécessitant une période d'apprentissage des outils,
- nécessitant une période pour la compréhension du problème,
- et nécessitant un temps pour la détermination progressive des vrais besoins.

2.2 Les méthodes traditionnelles

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



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

Malheureusement, les équipes qui évoluent dans un environnement changeant ou complexe savent à quel point il est difficile de s'en tenir aux décisions initiales. Le client réalise que ses besoins ont changé, ou bien l'équipe découvre en phase d'implémentation des erreurs de spécification ou de conception qui compromettent les plans de développement. Le changement s'impose donc tôt ou tard, mais voilà que cette organisation suppose l'absence de changement, ce qui peut compromettre la rentabilité du projet.

Mais puisque le changement est une composante incontournable de tout projet de développement logiciel, pourquoi ne pas l'accepter ? N'existe-t-il pas un moyen pour que les équipes de développement n'opposent plus de rigidité excessive aux demandes de leur maîtrise d'ouvrage ?

2.3 La méthode proposée

La méthodologie à suivre est basée sur la programmation intensive, itérative et progressive (incrémentale) centrée sur les tests.

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

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

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

Dans cette méthode :

- Le client pilote lui-même le projet, et ce de très près grâce à des cycles itératifs extrêmement courts (1 ou 2 semaines). Le client dans notre cas est le promoteur principal du sujet, à savoir Mr Djamal Bennouar, CC au Dpt Informatique de l'USDB.
- L'équipe, formée de deux étudiantes en phase de préparation de leur mémoire de fin d'études, doit livrer, très tôt dans le projet une première version du logiciel, et les livraisons de nouvelles versions s'enchaînent ensuite à un rythme soutenu pour obtenir un feedback maximal sur l'avancement des développements.
- L'équipe s'organise elle-même pour atteindre ses objectifs, en favorisant une collaboration maximale entre ses membres.

- L'équipe doit mettre en place un ensemble de jeux d'essai ou des tests automatiques pour toutes les fonctionnalités qu'elle développe, ce qui devrait garantir au produit un niveau de robustesse très élevé.

- Les développeurs améliorent sans cesse la structure interne du logiciel pour que les évolutions y restent faciles et rapides.

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

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

- Le client jouit d'une très grande visibilité sur l'avancement du projet.
- Le client utilise le logiciel lui-même comme support de réflexion pour le choix des fonctionnalités à implémenter. Il peut en particulier intégrer très tôt les retours des utilisateurs pour orienter les développements en conséquence.
- La première mise en production du logiciel intervient très tôt dans le projet, ce qui avance d'autant le moment à partir duquel le client peut en tirer des bénéfices.
- L'ordre d'implémentation des fonctionnalités n'est pas guidé par des contraintes techniques, mais par les demandes du client. Celui-ci peut donc focaliser les efforts de l'équipe sur les fonctionnalités les plus importantes dès le début du projet, et ainsi optimiser l'utilisation de son budget.

Chapitre 3

Analyse des besoins

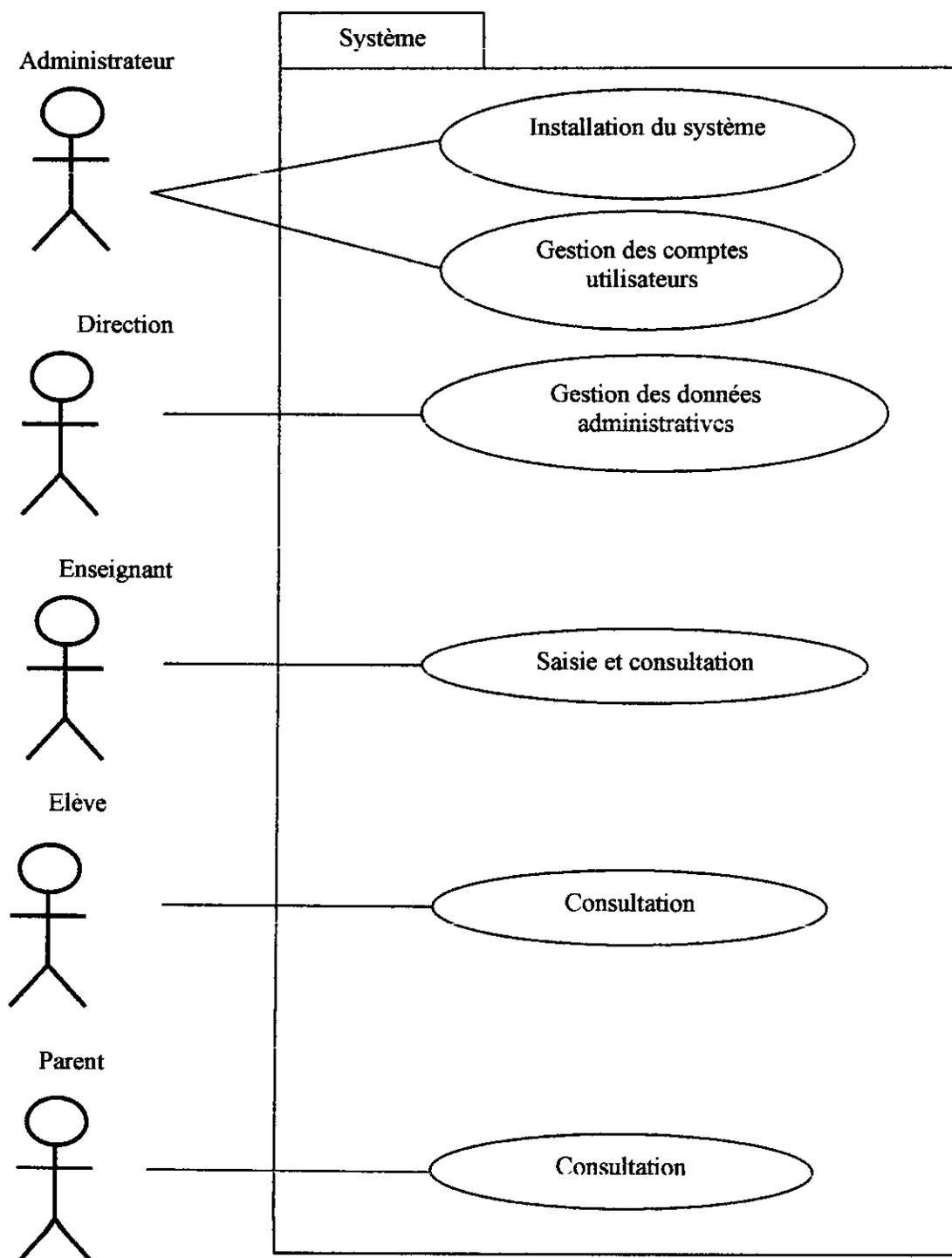
Analyse des besoins

L'analyse des besoins est une étape importante de la modélisation du système. Elle consiste à déterminer de manière explicite les besoins de l'utilisateur.

Nous avons utilisé la technique des cas d'utilisation proposée par UML. Cette technique va nous permettre de préciser le fonctionnement du système en décrivant les différentes manières pour l'utiliser. Les cas d'utilisations sont représentés graphiquement sur un schéma appelé : « diagramme de cas d'utilisation ».

Le diagramme de cas d'utilisation schématise les interactions permettant aux utilisateurs d'exploiter le système.

3.1 Diagramme de cas d'utilisation général



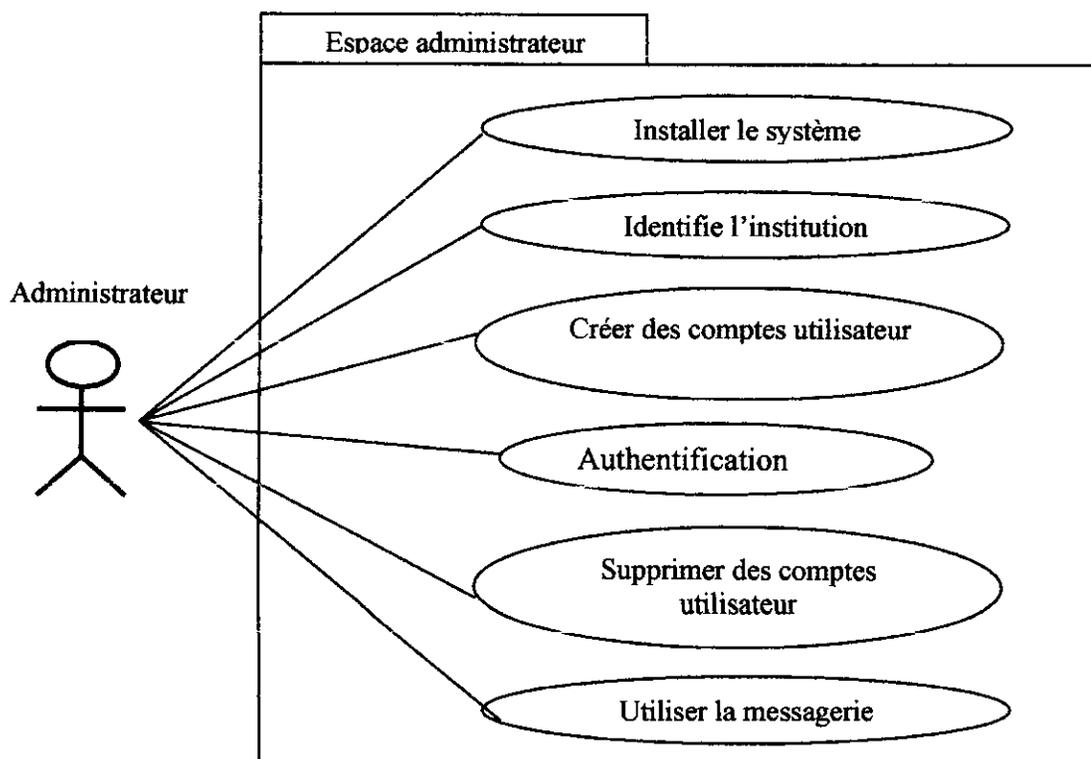
3.1 Diagramme de cas d'utilisation général

3.2 Diagramme de cas d'utilisation détaillé

3.2.1 L'administrateur du système

L'administrateur est le premier acteur du système, puisqu'il lui appartient de faire l'installation de ce système. Ses principales tâches sont :

- Installation : C'est la première tâche à effectuer dans le système. Pendant les étapes de l'installation, l'administrateur définit les éléments fondamentaux de l'établissement.
- Création de comptes : L'administrateur crée les comptes pour tous les utilisateurs du système. Les comptes sont protégés par mot de passe crypté en un seul sens.
- Suppression des comptes : La suppression des comptes utilisateur se fait aussi dans l'espace administrateur.
- Messagerie : Comme tout utilisateur du système, l'administrateur possède lui aussi une boîte d'envoi et une boîte de pour communiquer avec les autres utilisateurs.



3.2.1 Diagramme de cas d'utilisation de l'administrateur

3.2.2 La direction

Dans l'espace direction se trouvent les fonctions les plus importantes de la scolarité. Ces fonctions sont :

- Initialisation d'une année : Cette action consiste à faire toutes les modifications nécessaires pour avoir ce système tel qu'il était avant une ou plusieurs années. Quand la direction initialise une année, le système détecte l'année la plus ancienne puis une nouvelle base de données est créée. Cette base de données contient la même structure existante de l'établissement, tous les enseignants, et les élèves dans les niveaux où ils étaient l'année précédant l'année la plus ancienne. L'objectif d'une telle action est donc de faire gagner du temps saisi des enseignants et des élèves, et de ne saisir que leurs notes en cette année.

- Saisie du personnel administratif, inclure les photos dans la saisie et permettre la saisie par lot.

- Recrutement des enseignants : La direction se charge de saisir les enseignants, que ce soit un par un ou par groupe, ainsi que toutes leurs informations.

- Saisie des élèves avec leurs informations puis les affecter aux niveaux appropriés pour pouvoir retirer ensuite les certificats de scolarité de chaque élève.

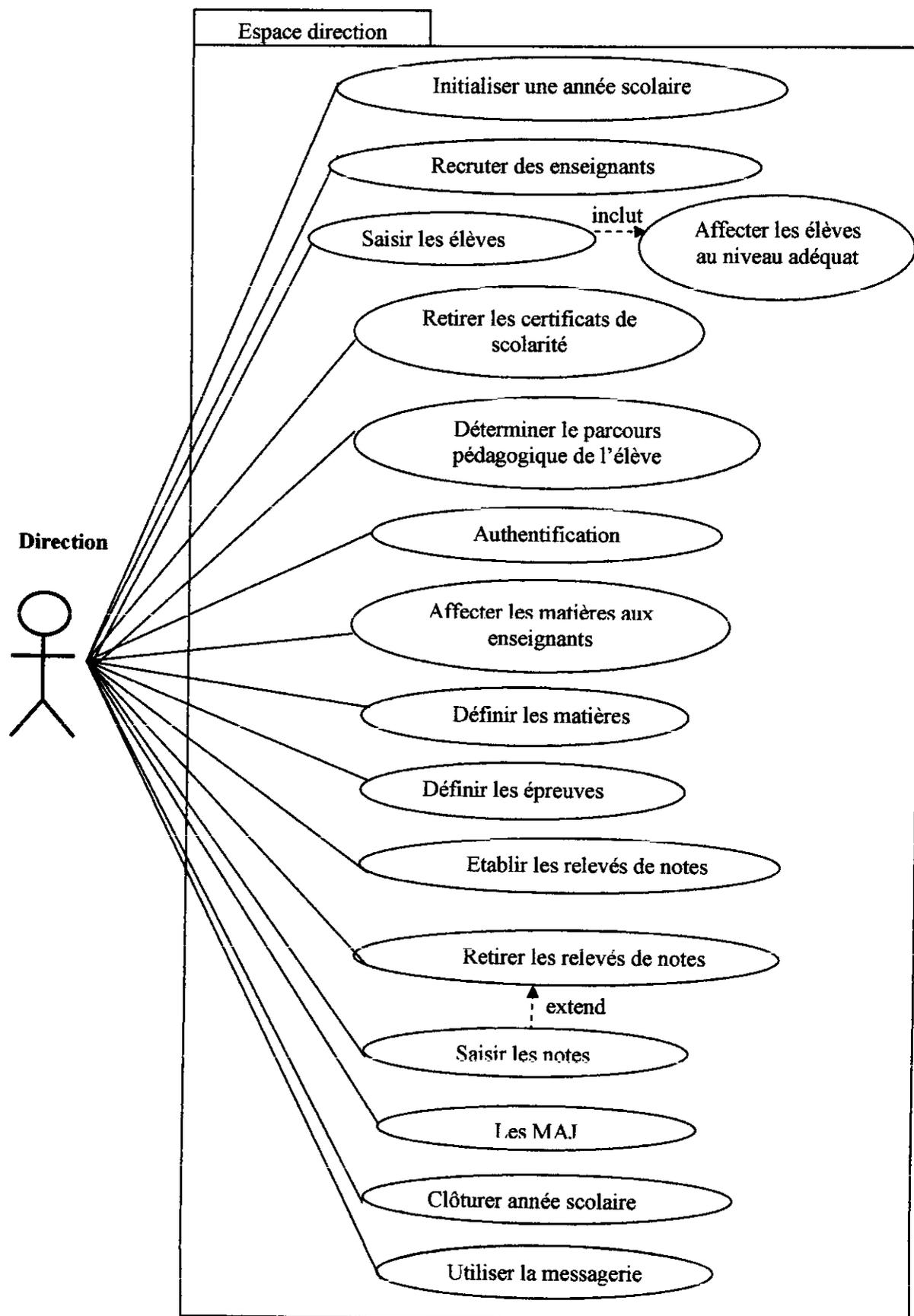
- Ajouter des matières et leur coefficient puis affecter les matières aux enseignants.

- Ajouter des épreuves avec leur poids. Le poids est utilisé dans la formule du calcul de la moyenne.

- Saisir les notes des enseignants qui ne possèdent pas de compte.

- Etablir les relevés de notes des élèves pour lesquels toutes les notes ont été saisies.

- Faire les différentes mises à jour, ajouter, modifier ou supprimer.
- Clôturer une année scolaire : Cette action entraîne la fermeture en modification de l'année courante, et l'ouverture d'une autre année, l'année supérieure, en faisant les passages nécessaires des élèves. Donc une nouvelle base de donnée sera créée et elle contiendra toutes les informations de l'ancienne, avec les modifications suite au passage ou au redoublement.
- Utiliser la messagerie : La direction pourra ainsi envoyer et recevoir des messages de tous les utilisateurs du système.
- Déterminer le parcours pédagogique complet de l'étudiant du primaire à l'université, en affichant toutes ses notes de tous les établissements par lesquels il est passé

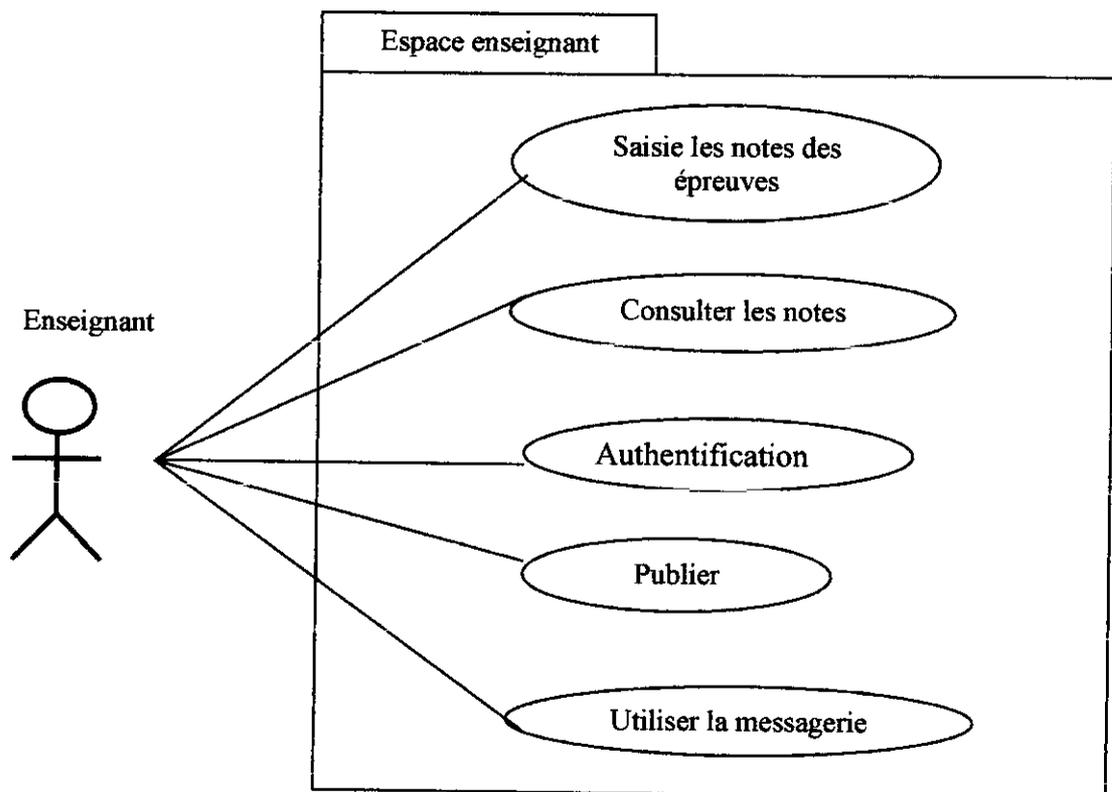


3.2.2 Diagramme de cas d'utilisation de la direction

3.2.3 Les enseignants

Un enseignant possède les tâches suivantes :

- Saisie des notes dans les matières qu'il enseigne.
- Voir les notes de ses élèves dans les autres matières.
- Autoriser les autres enseignants à consulter les notes de ses matières.
- Publier : Dans ce cas, l'enseignant travaille sur un ordinateur distant, il fait des modifications (saisie des notes par exemple) puis il se connecte au serveur de l'établissement pour lui envoyer les modifications.
- Utiliser la messagerie : Il peut envoyer et recevoir des messages de tout utilisateur du système.

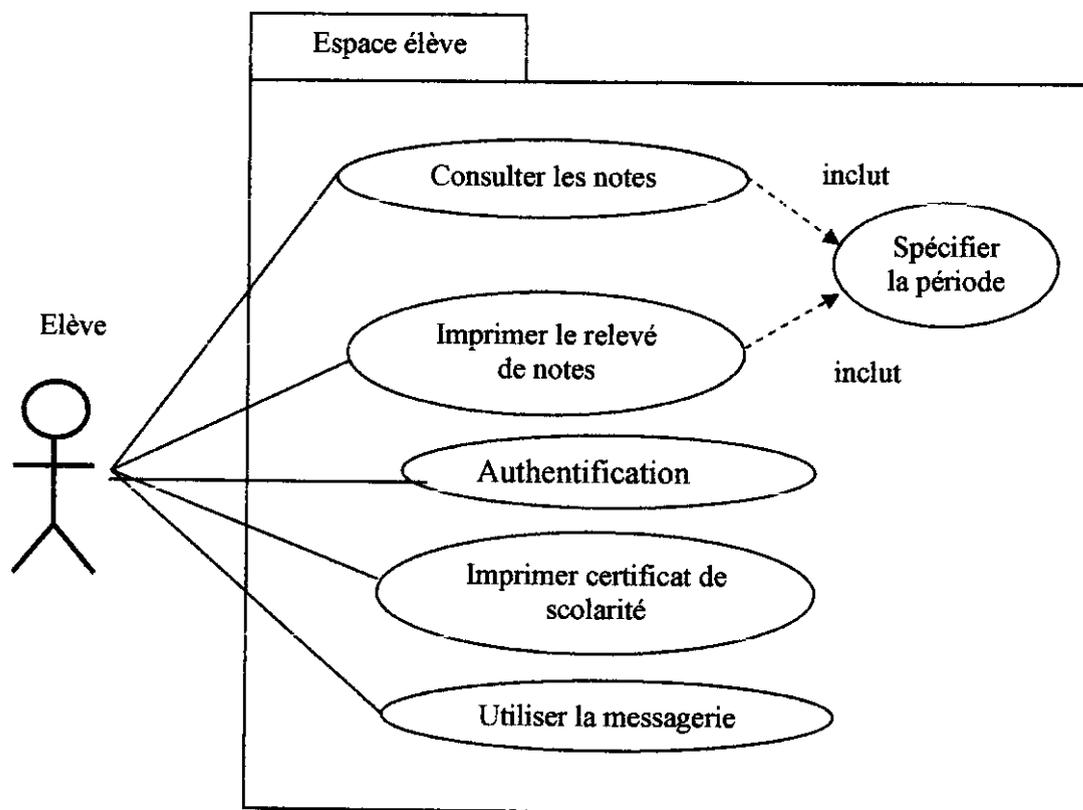


3.2.3 Diagramme de cas d'utilisation de l'enseignant

3.2.4 Les élèves

Un élève peut :

- Voir ses notes dans toutes les matières. Il ne peut pas voir les notes de ses camarades/ il ne peut voir que ses notes.
- Faire sortir un état : certificat de scolarité, relevé de note ou attestation de succès.
- Envoyer et recevoir des messages de tous les utilisateurs.

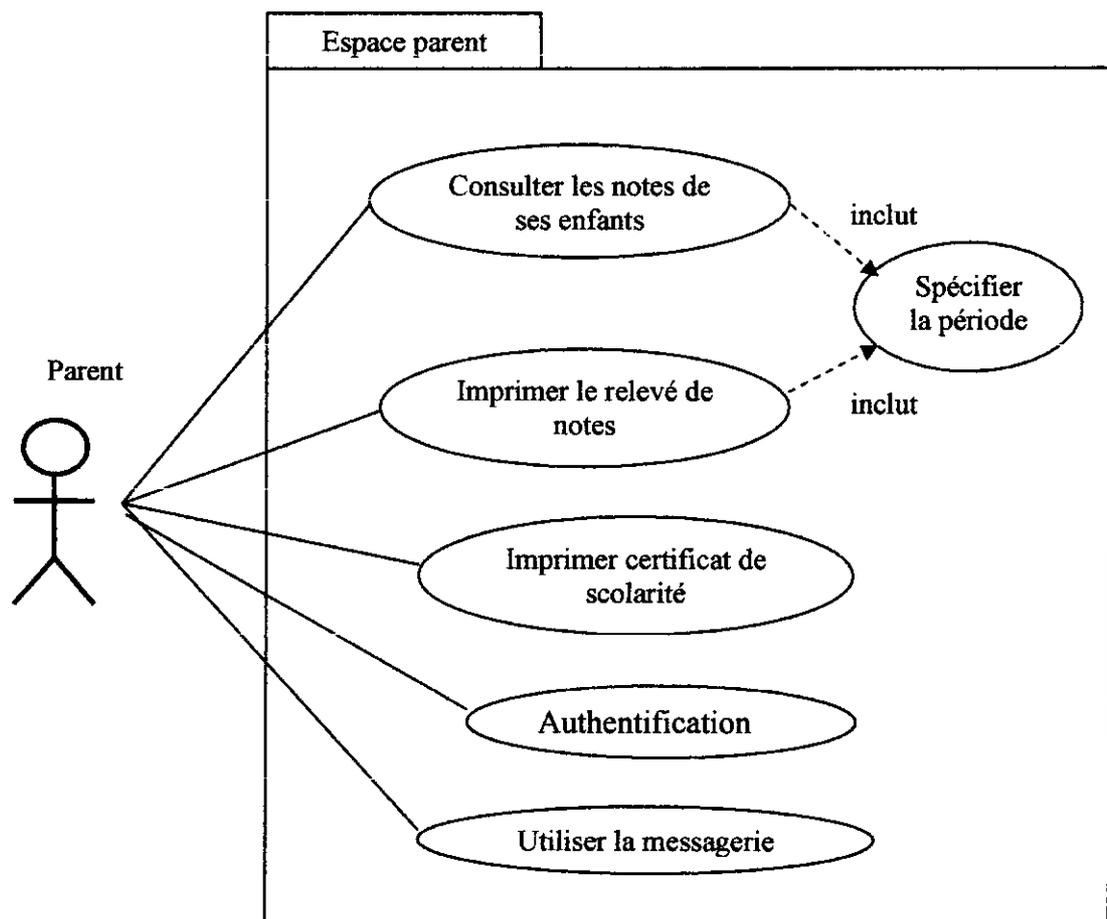


3.2.4 Diagramme de cas d'utilisation de l'élève

3.2.5 Les parents d'élève

Un parent d'élève/le tuteur peut :

- Voir les notes des enfants dont il est le tuteur. Les parents doivent créer leur compte avant, et préciser leurs enfants ? .
- Faire sortir un état : certificat de scolarité, relevé de note ou attestation de succès.
- Envoyer et recevoir des messages de tous les utilisateurs.



3.2.5 Diagramme de cas d'utilisation du parent d'élève

Chapitre 4

Conception

Conception

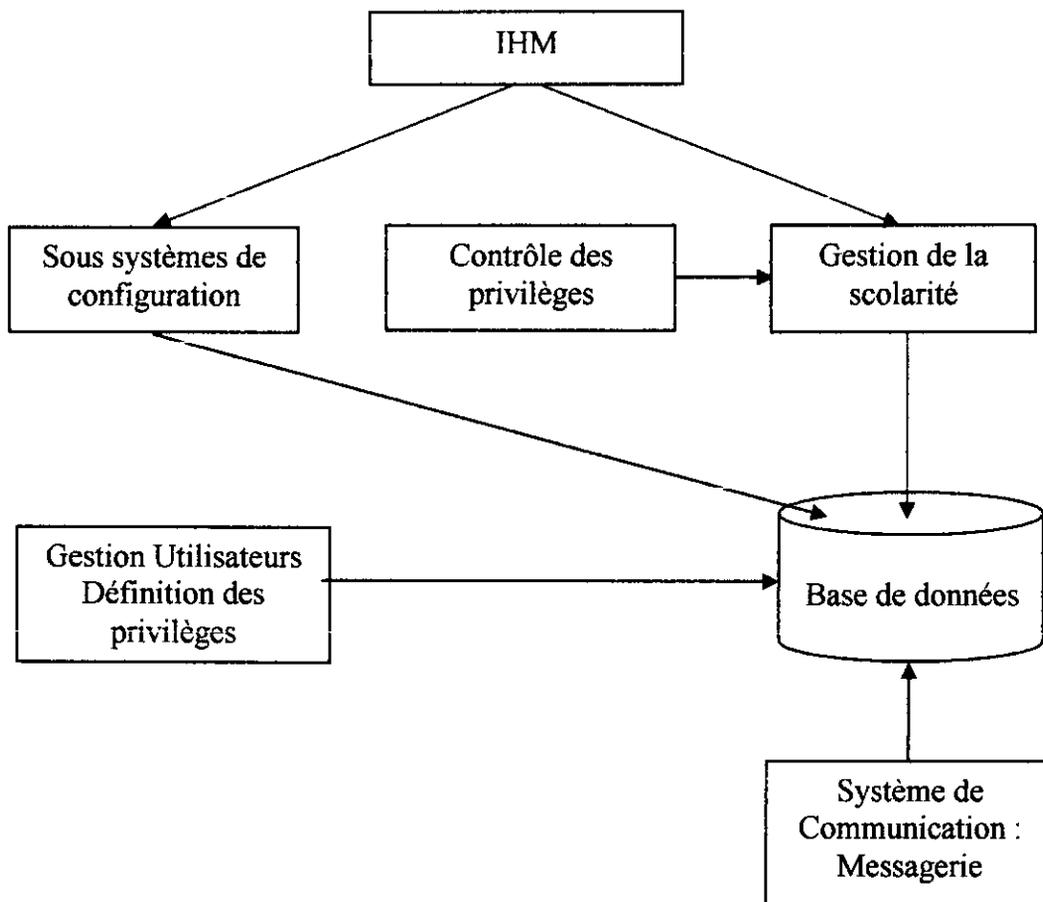
Dans ce chapitre, nous allons donner la description des différents cas d'utilisation du système. Pour chaque utilisateur, nous détaillons tous ses cas d'utilisation. Chaque cas d'utilisation est décrit textuellement de façon détaillée, mais donne également lieu à un diagramme de séquence simple représentant graphiquement la chronologie des interactions entre les acteurs et le système vu comme une boîte noire, dans le cadre du scénario nominal.

Un avantage important de cette technique consiste en la possibilité de découper le travail suivant les différents cas d'utilisation, plutôt que de vouloir tout traiter d'un bloc.

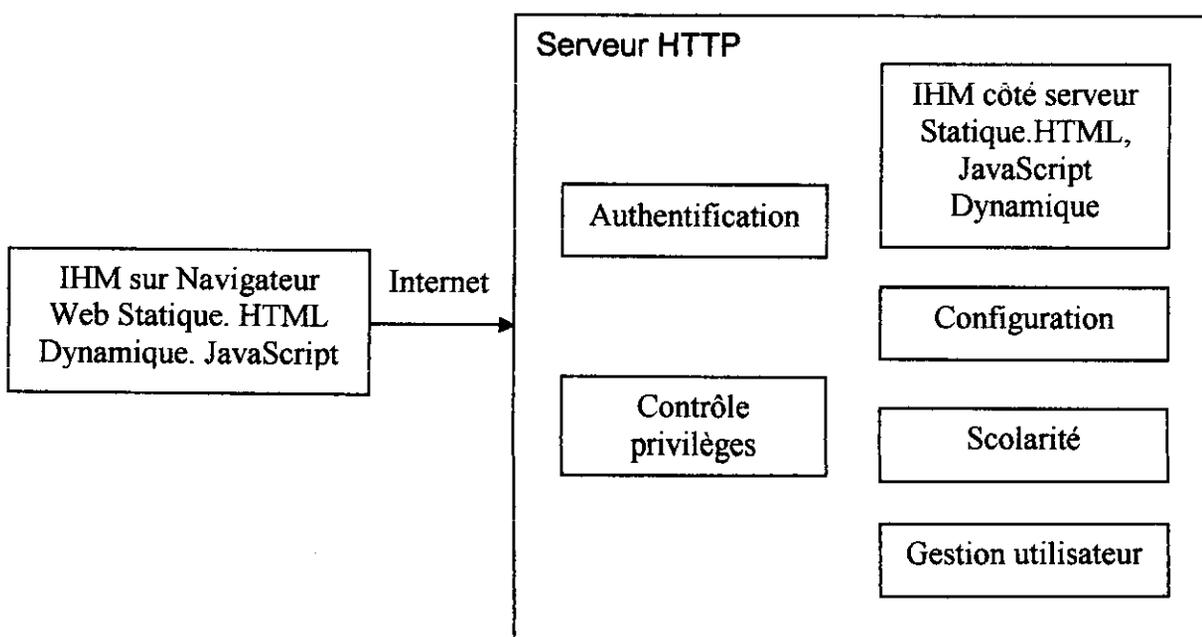
Notez que nous n'utilisons pas tous les types de diagrammes proposés par UML, mais seulement ceux que nous avons trouvés importants, en insistant particulièrement sur les diagrammes de séquence et les diagrammes d'activité. Cette limitation volontaire permet de réduire le temps d'apprentissage de la modélisation avec UML, tout en restant largement suffisante.

4.1 Architecture de l'application

Voici les principales composantes du système et leurs liens d'interaction :



4.2 Architecture de déploiement



4.3 Description des cas d'utilisation

4.3.1 Installation du système

Cas d'utilisation « Installer le système »

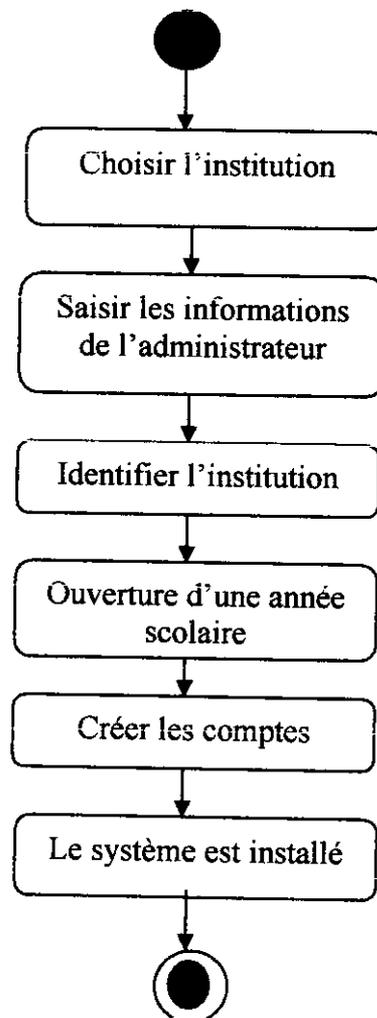
Généralité

Nom fiche de cas d'utilisation	etape1.php ; etape2.php ; etape2_pr.php; etape3_pr.php; etape4_pr.php; etape5_pr.php; créer_bdd.php
Cas d'utilisation	Installer le système
Acteurs	Administrateur
But	Installer le système « initialiser une année scolaire à l'institution choisie »
Description	L'administrateur commence l'installation du système par le choix de l'institution, il s'identifie et saisit les informations de l'institution et la finit par la création de compte pour la direction
Pré conditions	Aucune condition n'est imposée
Post conditions	L'institution choisie est créée avec deux comptes : un pour l'administrateur et l'autre pour la direction
Exception	Annulation : Pour la création de compte de la direction, si l'administrateur tape le même login que le sien, Le Système décèle une erreur de saisie et offre une nouvelle possibilité de saisie en affichant un message d'erreur

Scénario Nominal « Installer le système »

N°enchaînement	Actions acteur et Actions système
1	L'administrateur choisit l'institution
2	L'administrateur s'identifie.
3	L'administrateur identifie l'institution choisie
4	L'administrateur initialise une année scolaire
5	L'administrateur crée des comptes pour les premiers utilisateurs

Diagramme d'activité « Installer le système »



4.3.2 Authentification des utilisateurs

Cas d'utilisation « Authentification des utilisateurs »

Généralité

Nom fiche de cas d'utilisation	Accueil.php
Cas d'utilisation	Authentification des utilisateurs
Acteurs	Administrateur, direction, enseignant, élève, parent
But	Accéder à l'espace approprié pour l'utilisateur
Description	Le système est dans l'état : « Authentification » L'utilisateur tente d'accéder à son espace en remplissant un formulaire adéquat qui contient son login et mot de passe.
Pré conditions	Aucune condition n'est requise
Post conditions	L'utilisateur est dans son espace, il peut effectuer les tâches qui lui sont permises
Exception	Annulation : si l'utilisateur tape un login et un mot de passe qui ne correspondent pas, le système affiche un message et reviens à l'écran « page d'accueil »

Scénario Nominal

N° enchaînement	Actions acteur et Actions système
1	L'utilisateur demande de s'authentifier.
2	Le système affiche un formulaire d'authentification
3	L'utilisateur remplit le formulaire pour accéder à son espace.

Diagramme de séquence «Authentification des utilisateurs»

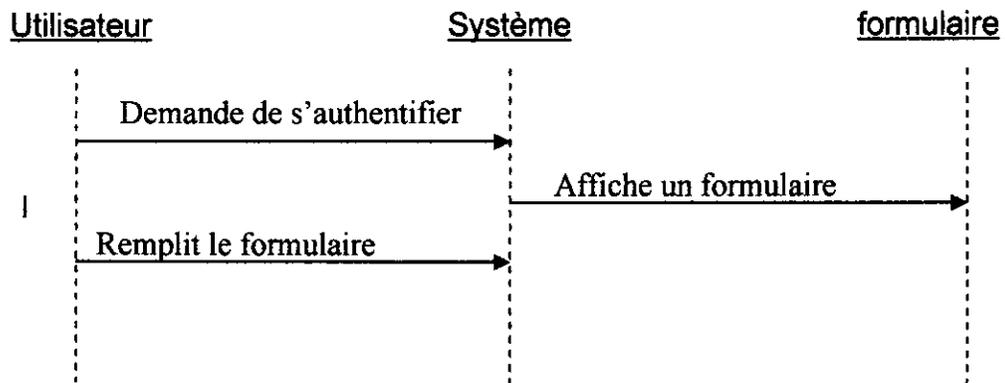
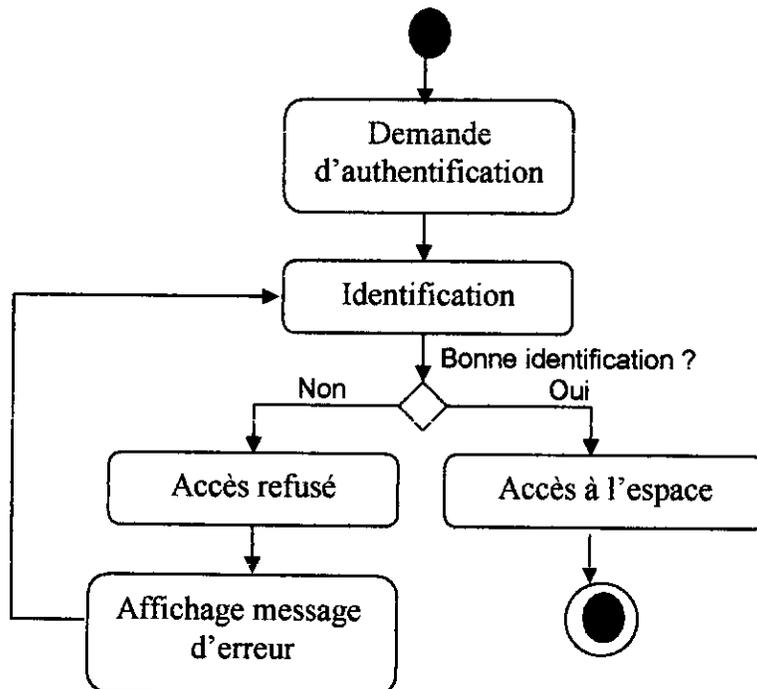


Diagramme d'activité «Authentification des utilisateurs»



4.3.3 Gestion de scolarité

Cette partie contient les cas d'utilisation suivants :

- Saisir les élèves.
- Recruter les enseignants.
- Affecter des matières et classes aux enseignants.
- Saisir les notes.
- Etablir les moyennes annuelles
- Initialiser une année scolaire.
- Clôturer une année scolaire.
- Faire les MAJ.
- Afficher le parcours pédagogique.

Cas d'utilisation « Saisir les élèves »

Généralité

Nom fiche de cas d'utilisation	Saisir_ele.php, remplir_ele.php
Cas d'utilisation	Saisir les élèves.
Acteurs	Direction
But	Inscrire de nouveaux élèves dans l'établissement
Description	Le système est dans l'état : « Ajouter élèves » La direction inscrit de nouveaux élèves dans l'établissement et les affecte à différents niveaux puis remplit un formulaire pour valider l'inscription
Pré conditions	La direction est identifiée.
Post conditions	De nouveaux élèves sont enregistrés, pouvoir retirer les certificats de scolarité
Exception	Annulation : si la direction saisit un matricule qui a déjà été saisi auparavant, le système affiche un message et revient à l'écran avec une liste des matricules déjà saisis et toutes les informations correspondantes

Scénario Nominal « Saisir les élèves »

N° enchaînement	Actions acteur et Actions système
1	La direction s'identifie pour accéder au système.
2	Le système accepte l'accès de la direction
3	La direction demande au système de saisir un ou plusieurs élèves
4	Le système affiche tous les niveaux et toutes leurs classes.
5	La direction choisit le niveau et la classe de des élèves.
6	Le système affiche un formulaire vide
7	L'agent remplit le formulaire, puis valide
8	Le système contrôle les informations saisies, s'il y a une erreur il affiche un message
9	Le système confirme l'enregistrement des élèves

Diagramme de séquence « Saisie des élèves »

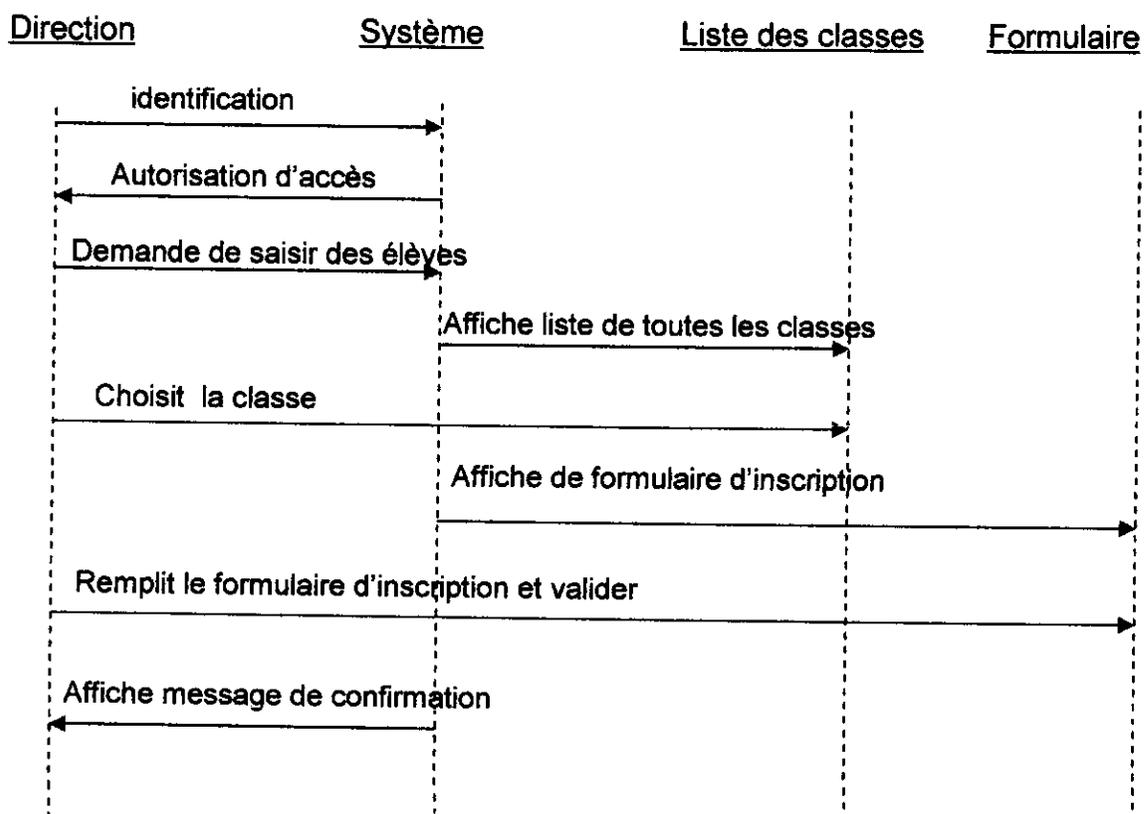
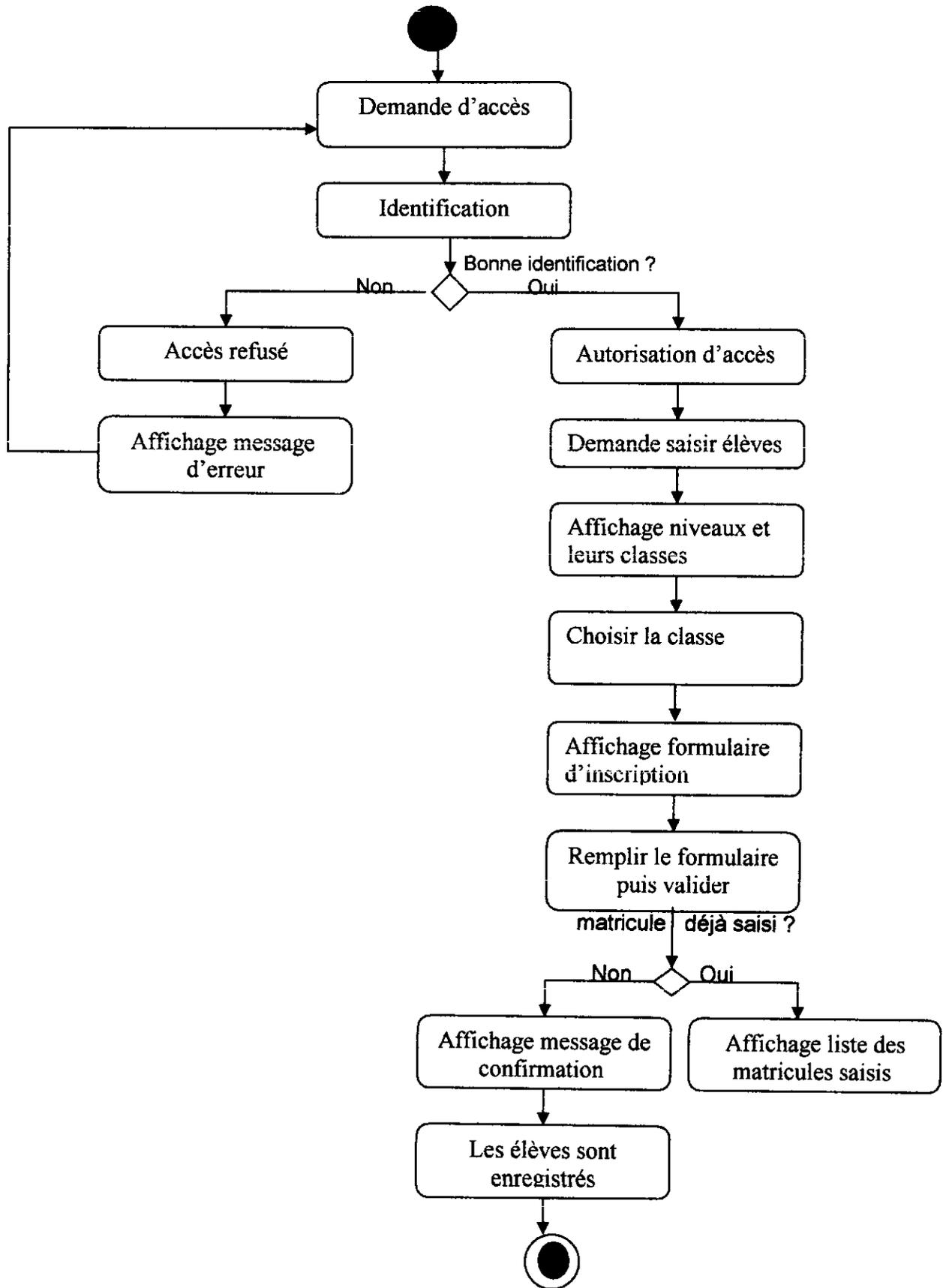


Diagramme d'activité « Saisir des élèves »



Cas d'utilisation « recruter des enseignants »

Généralité

Nom fiche de cas d'utilisation	recruter_ens.php
Cas d'utilisation	Recruter des enseignants
Acteurs	Direction
But	Recruter de nouveaux enseignants
Description	Le système est dans l'état : « Ajouter enseignants» La direction ajoute de nouveaux enseignants dans l'établissement, elle pour par la suite leur affecter des classes et des matières
Pré conditions	La direction est identifiée
Post conditions	Les enseignants sont enregistrés
Exception	

Scénario Nominal « recruter des enseignants »

N° d'enchaînement	Actions acteur et Actions système
1	La direction s'identifie pour accéder au système.
2	Le système accepte l'accès de la direction
3	La direction demande de recruter de nouveaux enseignants.
4	Le système affiche un formulaire d'inscription vide.
5	La direction remplit le formulaire de recrutement de l'enseignant, puis valide.
6	Le système affiche un message de confirmation de recrutement.

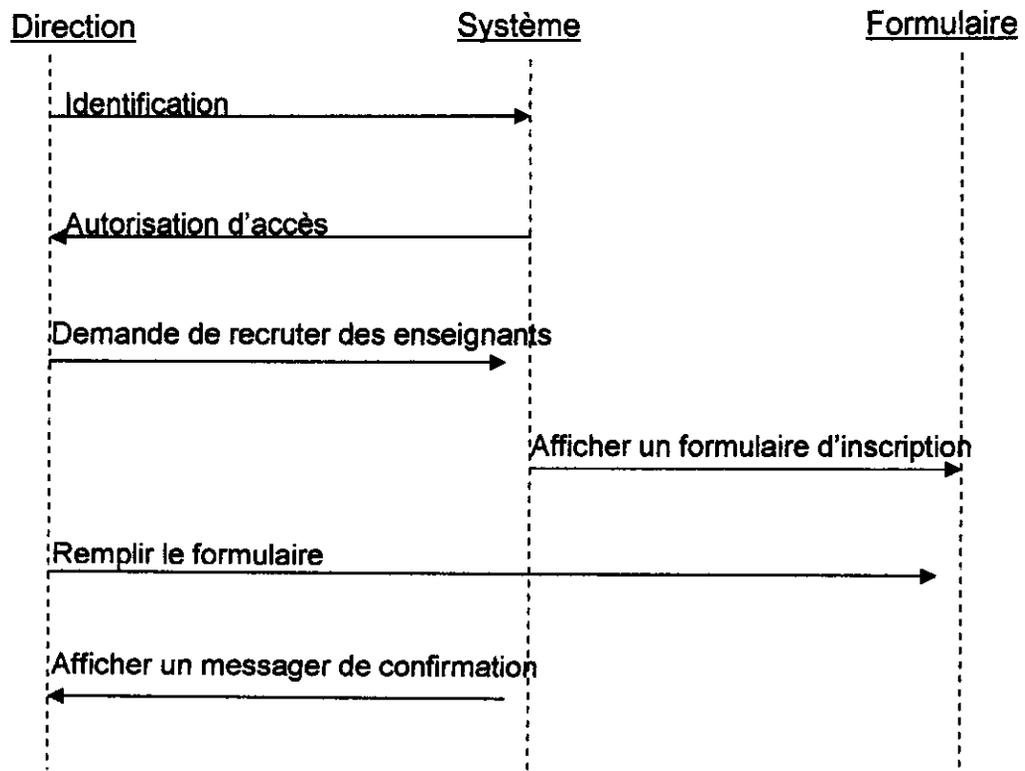
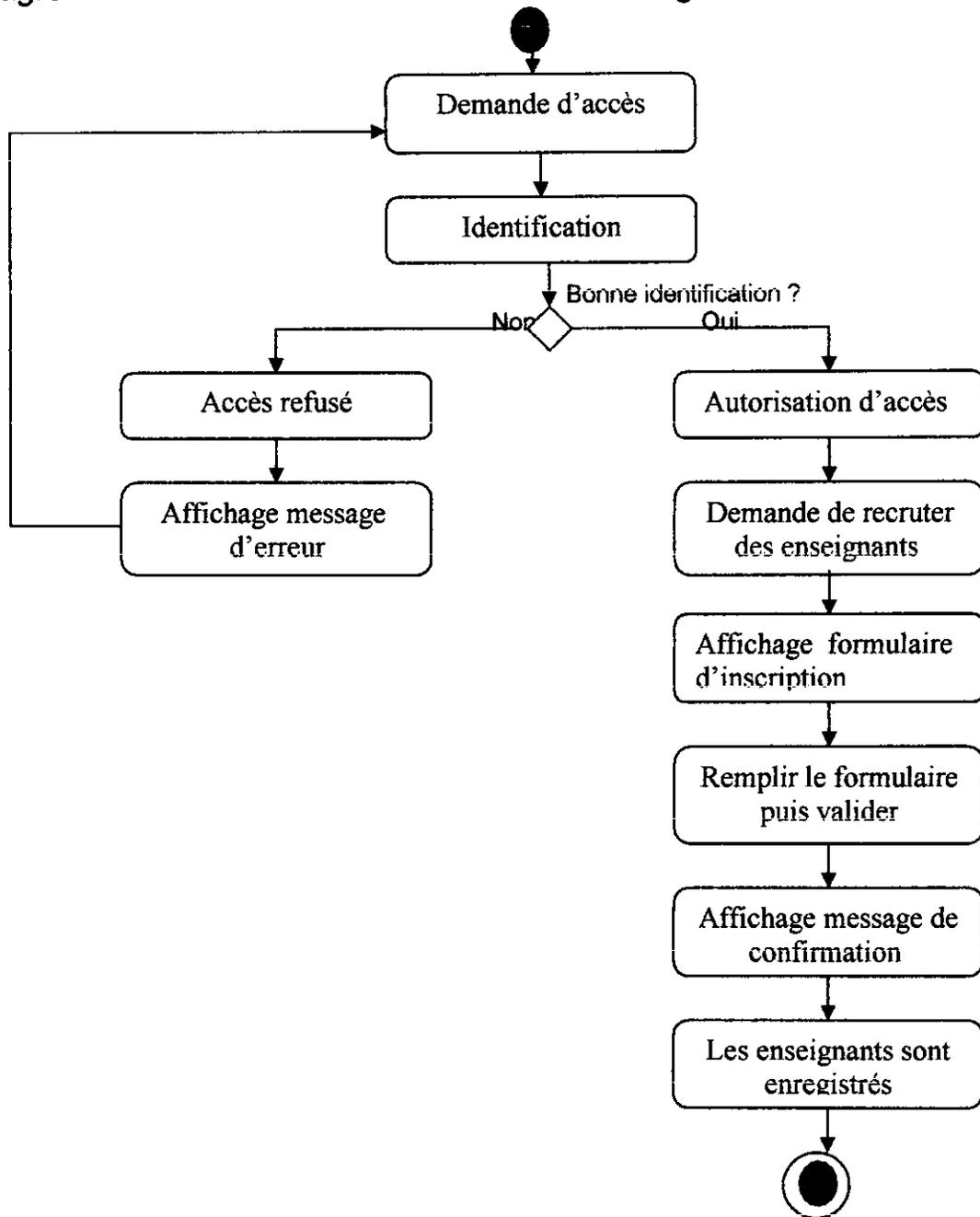
Diagramme de séquence « recrutement des enseignants »

Diagramme d'activité « recrutement des enseignants »



Cas d'utilisation « Affecter les classes et les matières aux enseignants »

Généralité

Nom fiche de cas d'utilisation	Affec_ens_cla.php ; Affec_ens_mat.php
Cas d'utilisation	Affecter les classes et les matières aux enseignants
Acteurs	Direction
But	Affecter à l'enseignant des matières dans des classes précises
Description	Le système est dans l'état : « Affecter les matières et classes aux enseignants » La direction affecte des matières et des classes qui sont enregistrées dans l'établissement à des enseignants enregistrés pour que ces derniers puissent saisir des notes de ces matières
Pré conditions	La direction est identifiée, l'enseignant est enregistré, les matières et les classes sont enregistrées
Post conditions	L'enseignant pourra enseigner ces matières dans ces classes, et saisir les notes
Exception	

Scénario Nominal

N° d'enchaînement	Actions acteur et Actions système
1	La direction s'identifie pour accéder au système.
2	Le système accepte l'accès de la direction
3	La direction demande d'affecter des matières et des classes à l'enseignant
4	Le système affiche toutes les classes de l'institution.
5	La direction choisit une classe
6	Le système affiche la liste des matières du niveau de la classe choisie
7	La direction choisit les matières à affecter à cet enseignant
8	Le système accepte l'affectation.

Diagramme de séquence « Affecter des matières et classes aux enseignants »

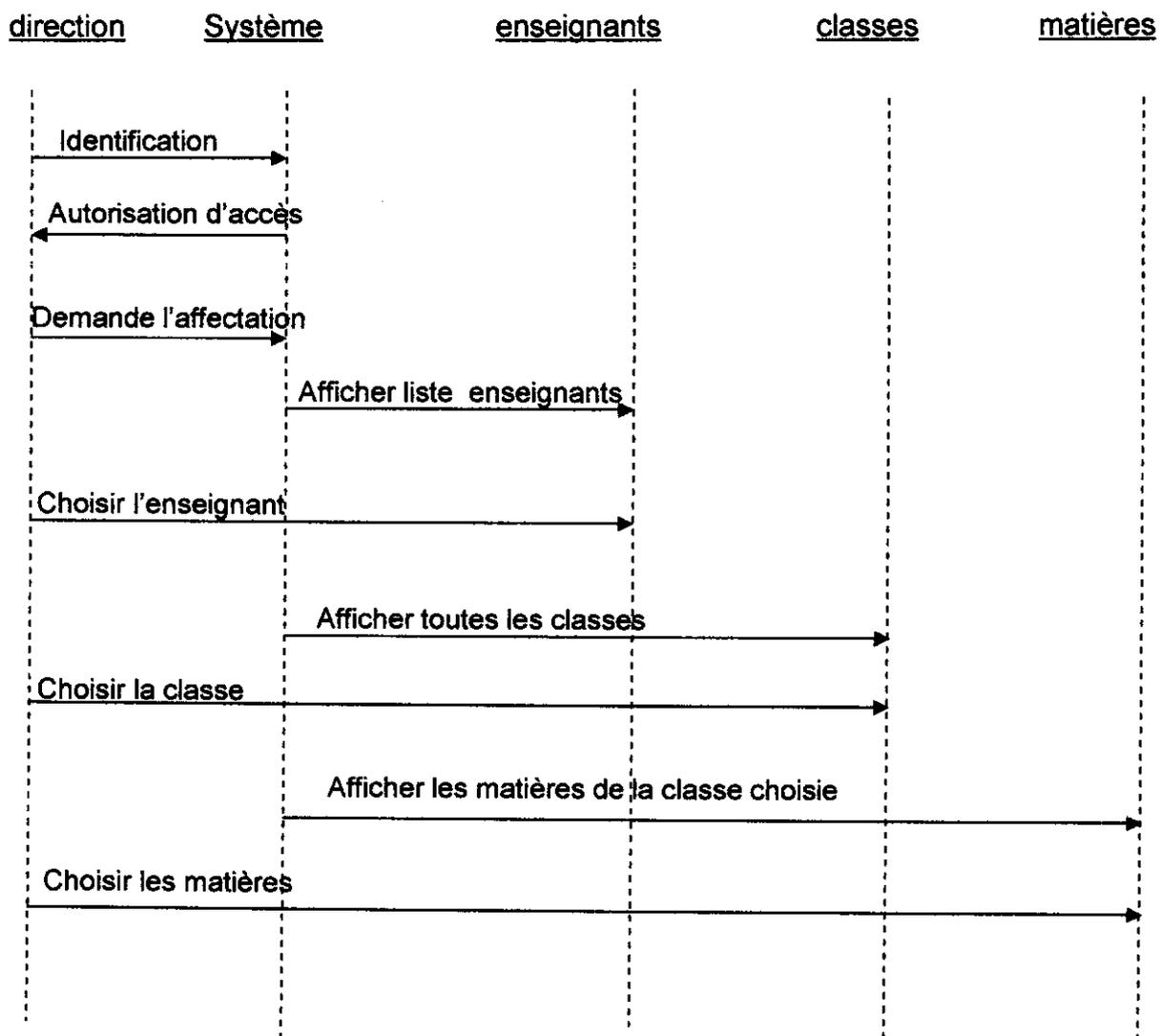
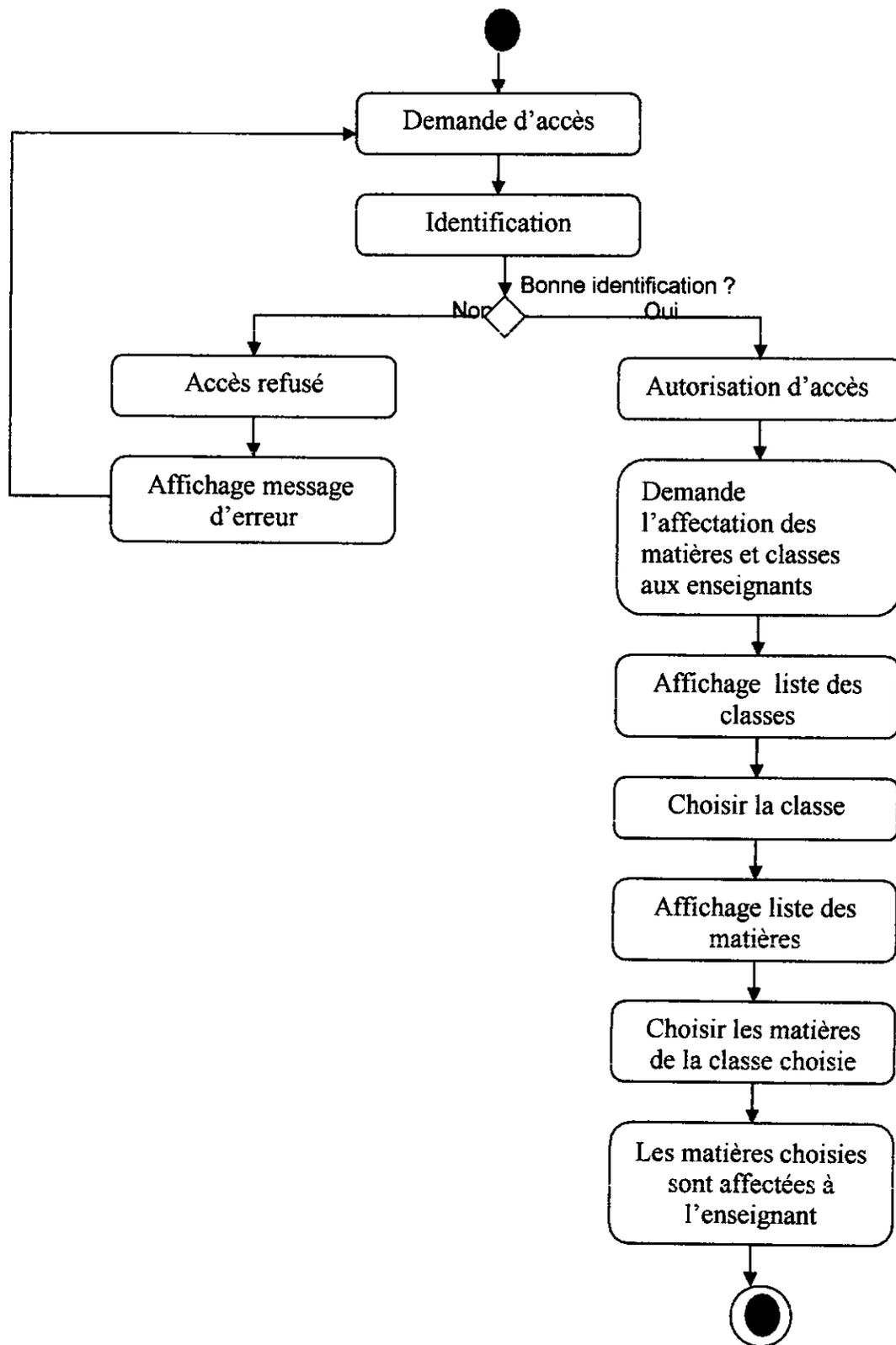


Diagramme d'activité « Affecter des matières et classes aux enseignants »



Cas d'utilisation « Saisir les notes »

Généralité

Nom fiche de cas d'utilisation	Saisi_note.php
Cas d'utilisation	Saisir les notes.
Acteurs	Direction
But	Saisir les notes des enseignants qui n'ont pas pu le faire
Description	Le système est dans l'état : « Saisir les notes » La direction sélectionne les épreuves des matières qui n'ont pas été saisies par les enseignants et saisit les notes
Pré conditions	La direction est identifiée, les notes des matières n'ont pas été saisies par l'enseignant.
Post conditions	Les notes des élèves sont saisies.
Exception	

Scénario Nominal

N° d'enchaînement	Actions acteur et Actions système
1	La direction s'identifie pour accéder au système.
2	Le système accepte l'accès de la direction
3	La direction demande de saisir des notes
4	Le système affiche toutes les classes et les matières de l'institution
5	La direction choisit la classe et la matière
6	Le système affiche les épreuves qui n'ont pas été saisie
7	La direction choisit l'épreuve
8	Le système affiche un formulaire pour saisir les notes de la matière choisie

Diagramme de séquence « Saisir les notes »

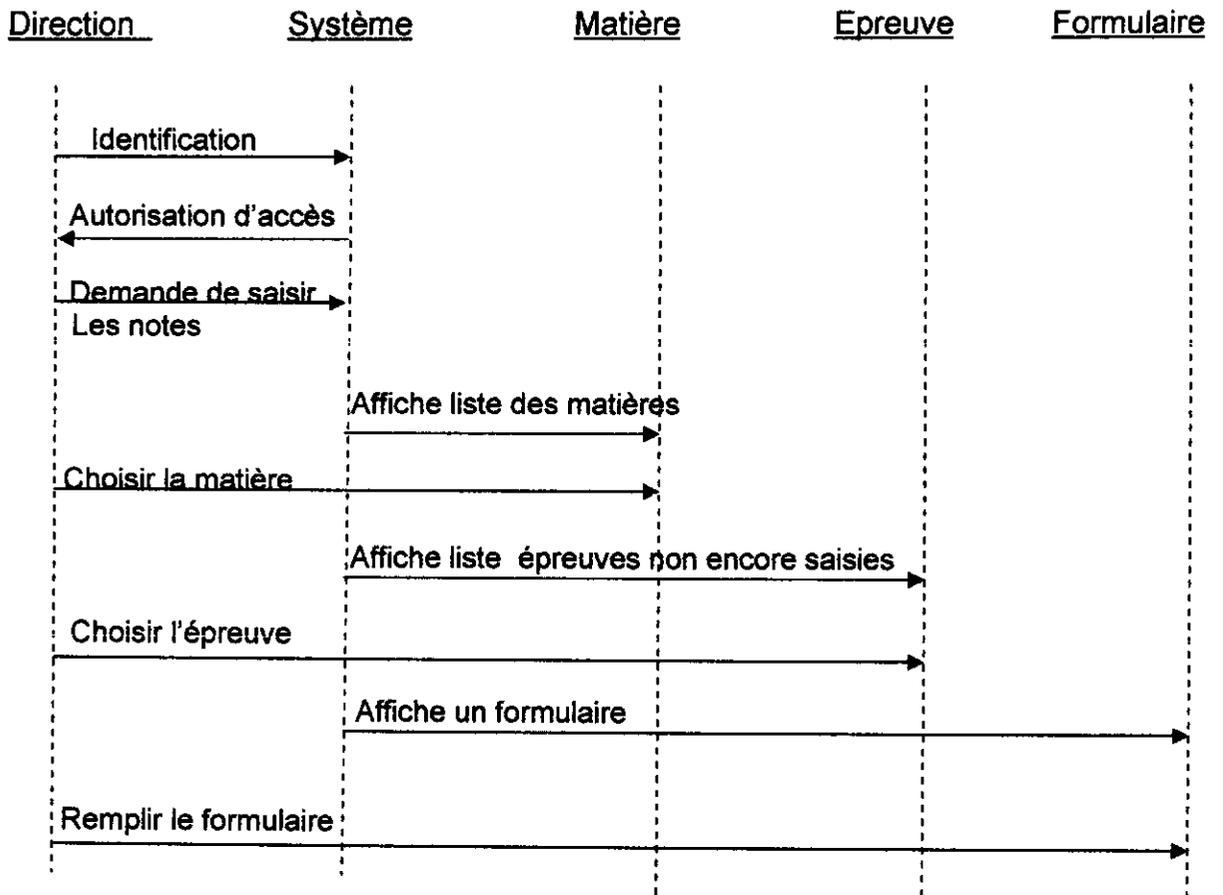
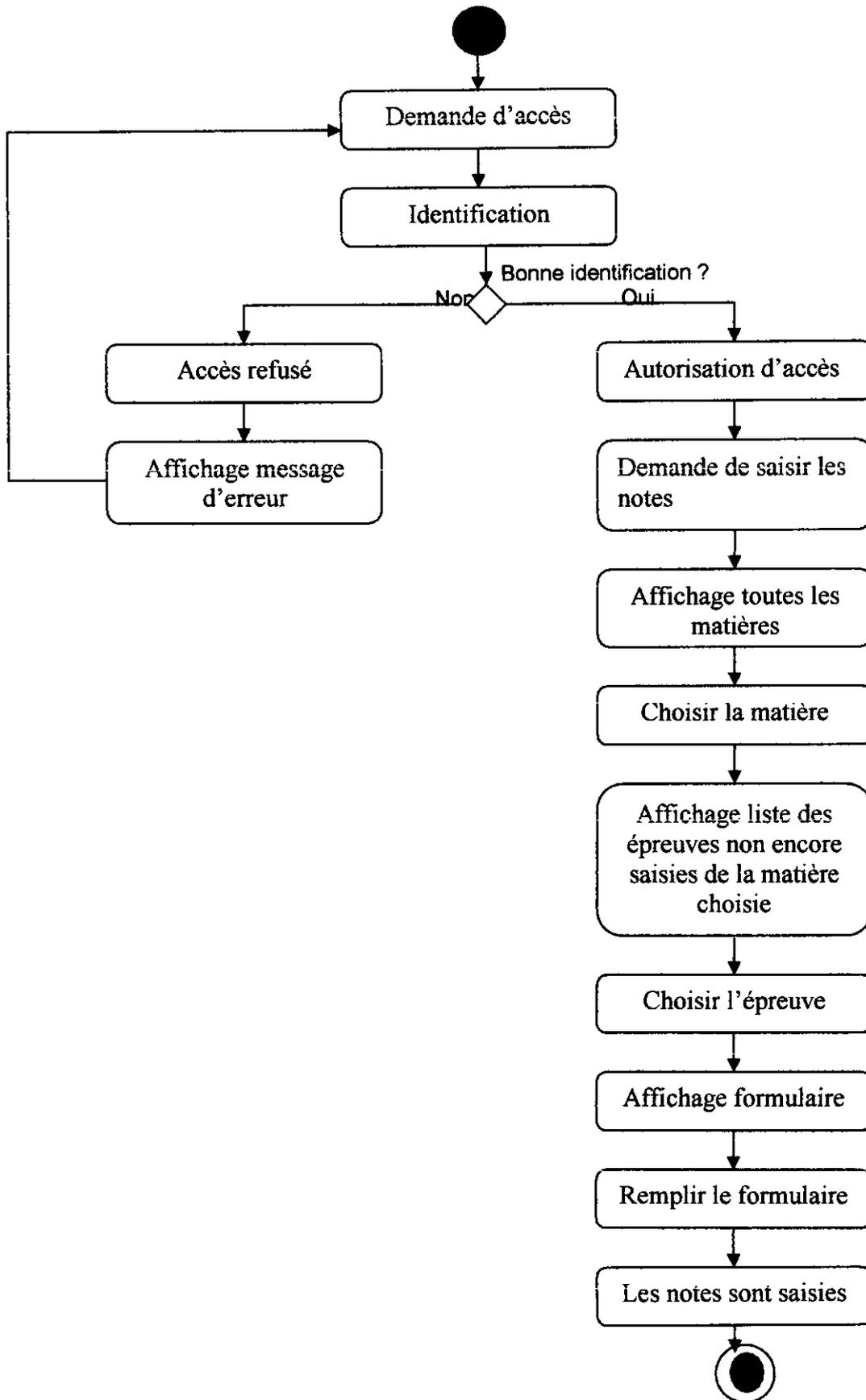


Diagramme d'activité « Saisir les notes »



Cas d'utilisation « Etablir les moyennes annuelles »

Généralité

Nom fiche de cas d'utilisation	relev_note0.php
Cas d'utilisation	Etablir les relevés des notes.
Acteurs	Direction
But	Etablir le passage des élèves
Description	Le système est dans l'état : « Etablir les moyennes annuelles » La direction peut établir les moyennes annuelles si toutes les notes sont saisies
Pré conditions	La direction est identifiée, les notes de toutes les épreuves de toutes les épreuves sont saisies.
Post conditions	pouvoir retirer les bulletins des notes de la dernière période.
Exception	Annulation : si les notes des épreuves des matières de la classe choisie ne sont pas toutes saisies, le système affiche un message et revient à l'écran « espace direction »

Scénario Nominal « Etablir les moyennes annuelles »

N° d'enchaînement	Actions acteur et Actions système
1	La direction s'identifie pour accéder au système.
2	Le système accepte l'accès de la direction
3	La direction demande d'établir les moyennes annuelles
4	Le système affiche toutes les classes
5	La direction choisit la classe
6	Le système affiche tous les élèves de la classe choisie ainsi que leur moyenne annuelle
7	La direction établit le passage des élèves au niveau supérieur
8	La direction confirme que ces résultats sont finaux

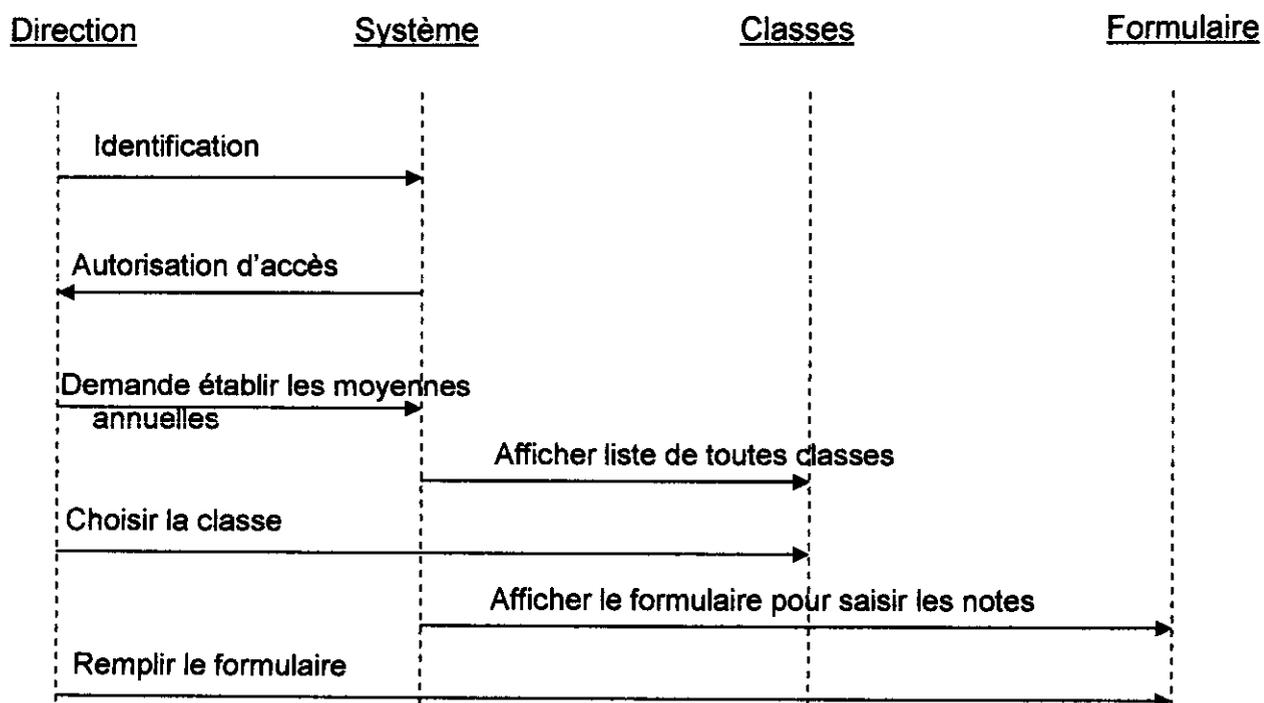
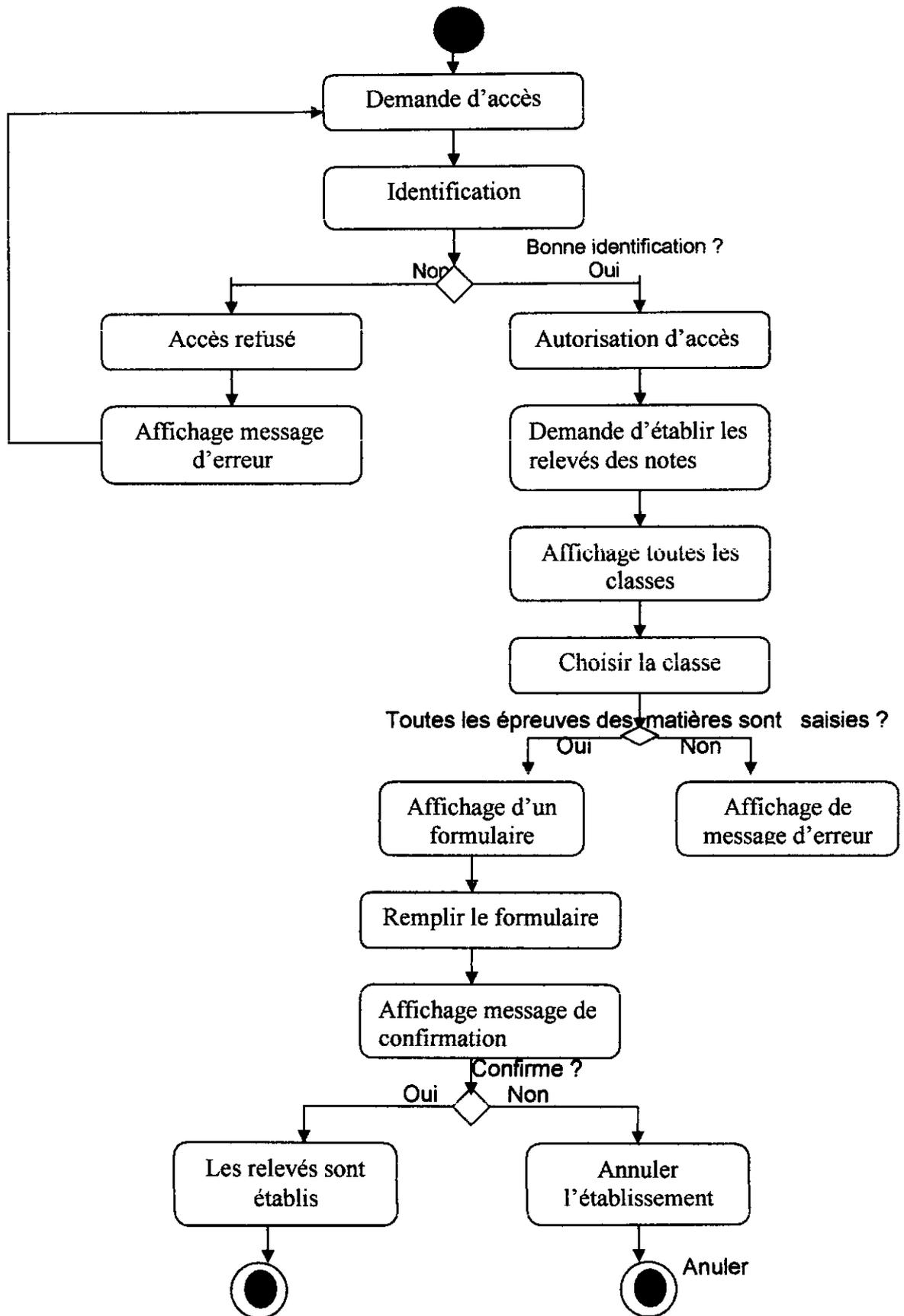
Diagramme de séquence « Etablir les moyennes annuelles »

Diagramme d'activité « Etablir les moyennes annuelles »



Cas d'utilisation « Initialiser une année scolaire »

Généralité

Nom fiche de cas d'utilisation	Init_an_scol.php
Cas d'utilisation	Initialiser une année scolaire
Acteurs	Direction
But	
Description	Le système est dans l'état : « Initialiser une année scolaire », Le système lui fournit la possibilité d'initialiser une année qui vient juste avant l'année précédente déjà initialisée la direction confirme d'initialiser une année scolaire.
Pré conditions	La direction est identifiée
Post conditions	Une année ; qui vient juste avant l'année précédente initialisée dans cette école ; est initialisée
Exception	Annulation : Si la direction annule l'initialisation; Le système revient à l'écran « page direction »

Scénario Nominal « Initialiser une année scolaire »

N° d'enchaînement	Actions acteur et Actions système
1	La direction s'identifie pour accéder au système.
2	Le système accepte l'accès de la direction
3	La direction demande d'initialiser une année scolaire
4	Le système affiche un message de confirmation
5	La direction confirme l'initialisation de l'année scolaire

Diagramme de séquence « Initialiser une année scolaire »

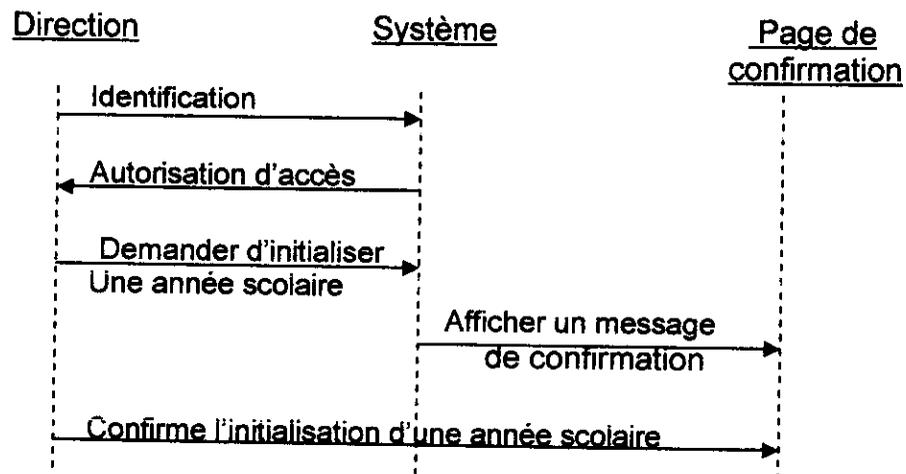
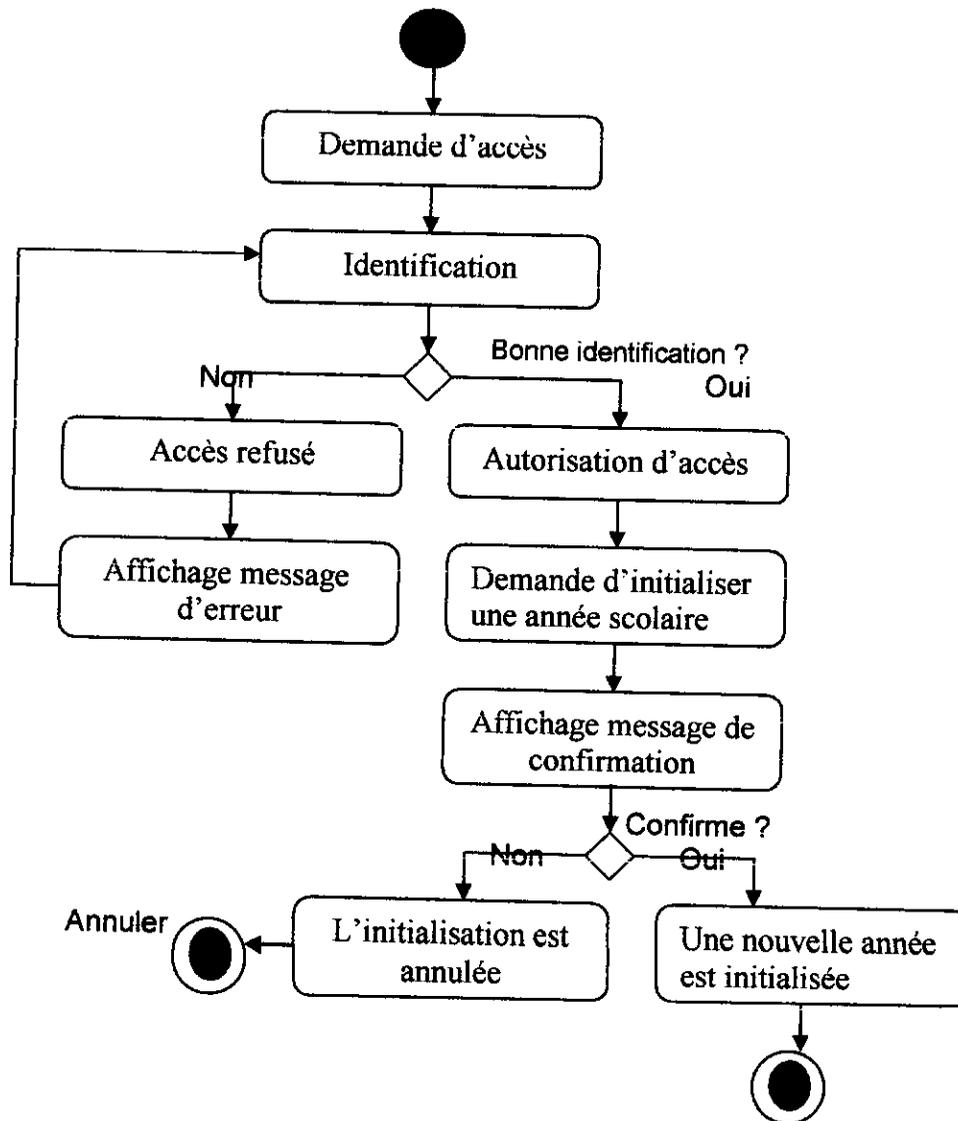
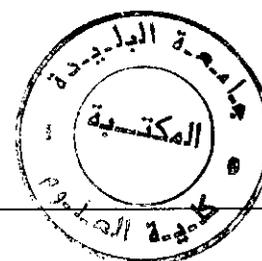


Diagramme d'activité « Initialiser une année scolaire »



Cas d'utilisation « Clôturer une année scolaire »



Généralité

Nom fiche de cas d'utilisation	clo_an_scol.php
Cas d'utilisation	Clôturer une année scolaire
Acteurs	Direction
But	Clôturer l'année et ne permettre d'y accéder qu'en mode consultation
Description	Le système est dans l'état : « Clôture d'une année scolaire », la direction confirme la clôture d'une année scolaire pour empêcher les modifications possibles dans cette année. On pourra y accéder en mode consultation par conséquent une année qu'il la poursuit est initialisée tout en reprenant la structure de l'établissement existante et les utilisateurs valides et réalisant le passage ou le redoublement automatique des élèves
Pré conditions	La direction est identifiée, les moyennes annuelles établies
Post conditions	Empêcher de faire des traitements autres que les consultations
Exception	Annulation : Si la direction annule l'initialisation; Le système revient à l'écran « page direction » Si l'établissement des moyennes annuelles n'a pas été réalisé, le système affiche un message et revient à l'écran « page direction »

Scénario Nominal « Clôturer une année scolaire »

N° d'enchaînement	Actions acteur et Actions système
1	La direction s'identifie pour accéder au système.
2	Le système accepte l'accès de la direction
3	La direction demande de clôturer une année scolaire
4	Le système affiche un message de confirmation
5	La direction confirme la clôture de l'année scolaire

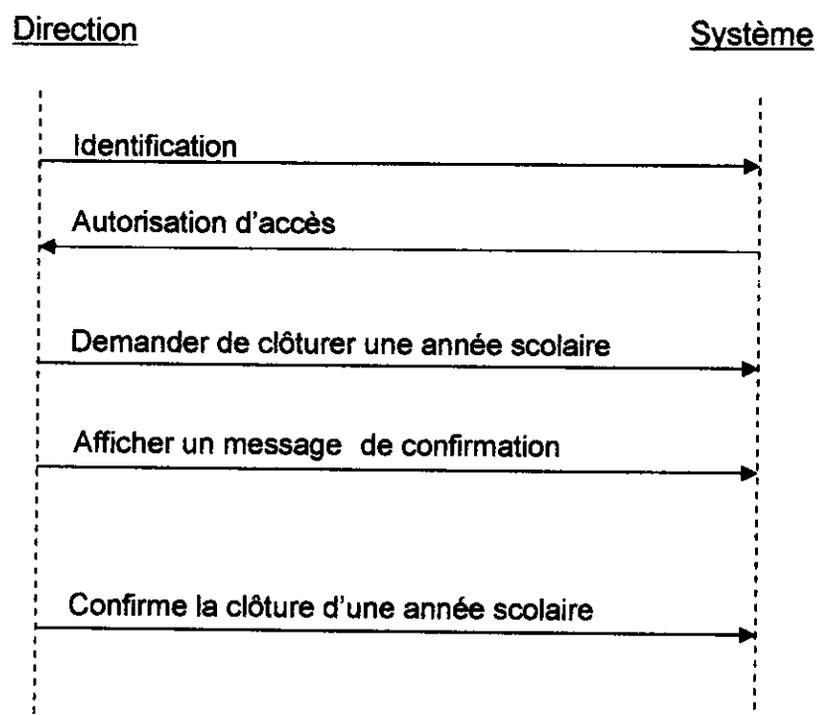
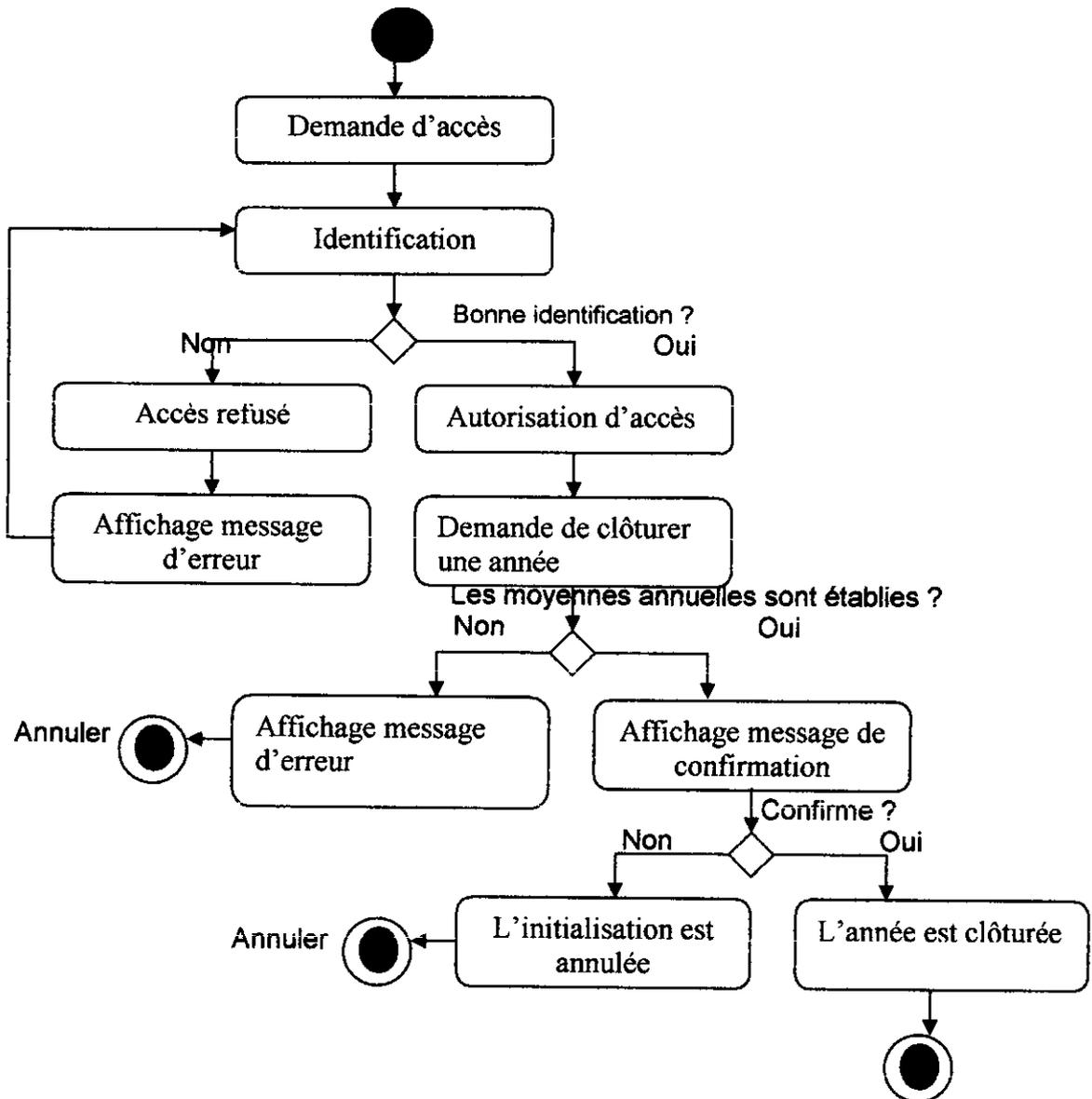
Diagramme de séquence « Clôturer une année scolaire »

Diagramme d'activité « Clôturer une année scolaire »



Cas d'utilisation « Consulter les notes »

Généralité

Nom fiche de cas d'utilisation	affi_note_dir.php
Cas d'utilisation	Consultation les notes
Acteurs	Direction
But	Consulter les notes de toutes les matières et de toutes les périodes
Description	Le système est dans « Afficher notes » La direction peut à tout moment consulter les notes de toutes les matières de n'importe quelle classe
Pré conditions	La direction est identifiée
Post conditions	Les notes sont affichées
Exception	Aucune

Scénario Nominal

N° d'enchaînement	Actions acteur et Actions système
1	La direction s'identifie pour accéder au système.
2	Le système accepte l'accès de la direction
3	La direction demande de consulter les notes
4	Le système affiche une liste de toutes les classes et leurs matières
5	La direction choisit une classe et une matière
6	Le système affiche les notes de la matière choisie de toutes les épreuves

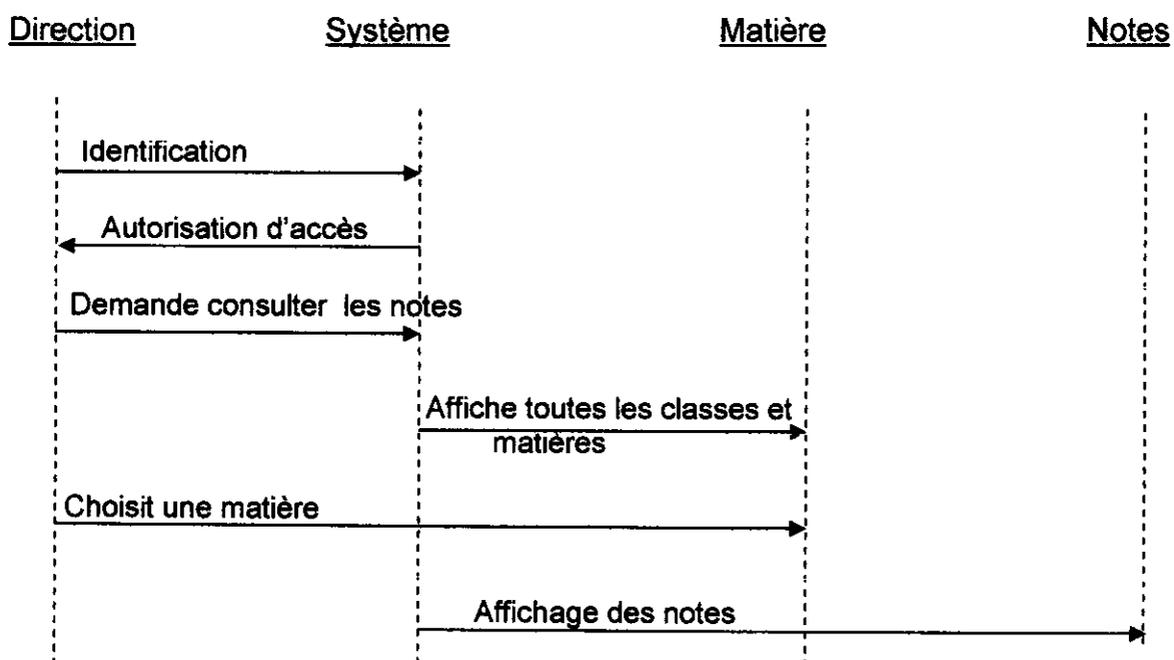
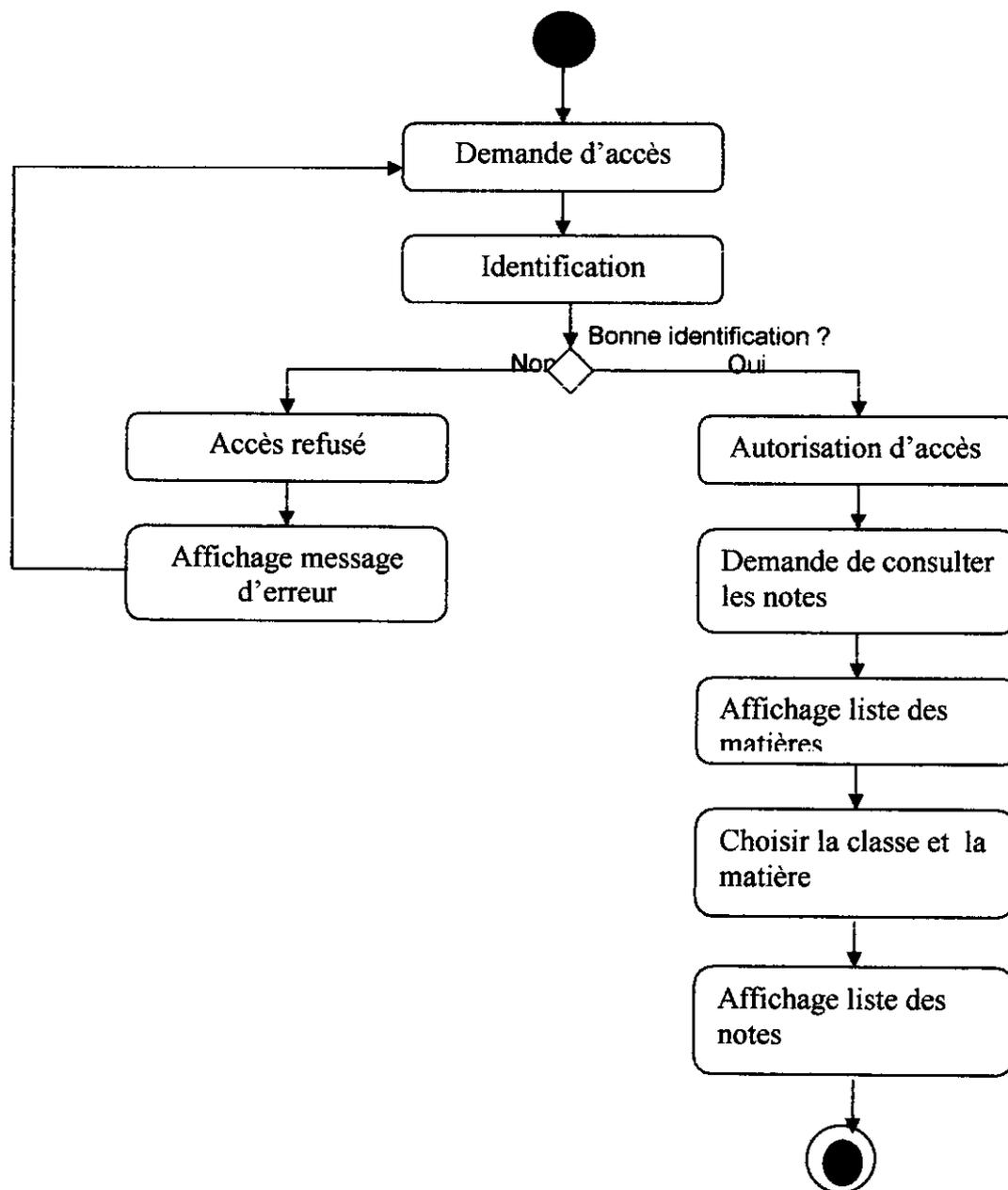
Diagramme de séquence « Consulter les notes »

Diagramme d'activité « Consulter les notes »



4.3.4 Impression des états

Cas d'utilisation « Retirer les relevés des notes »

Généralité

Nom fiche de cas d'utilisation	Retir_bul.php
Cas d'utilisation	Retirer les relevés des notes
Acteurs	Elève
But	Imprimer les relevés des notes de l'élève
Description	Le système est dans l'état : « Retirer mes relevés de notes », L'élève peut choisir la période pour imprimer son relevé des notes de cette période mais il faut que toutes les notes de toutes les épreuves de ses matières soient saisies dans cette période
Pré conditions	L'élève est identifié, les notes de toutes les épreuves de toutes les matières de la période choisie sont saisies
Post conditions	Le relevé est affiché
Exception	Annulation : S'il y a une épreuve ou plus d'une matière du niveau de l'élève et dans la période choisie qui n'ont pas été saisies ; Le système affiche un message et revient à l'écran « page élève » Si la période choisie est la dernière période et si les moyennes annuelles n'ont pas été établies ; Le système affiche un message et revient à l'écran « page élève »

Scénario Nominal « Retirer les relevés des notes »

N° d'enchaînement	Actions acteur et Actions système
1	Un élève demande l'accès au système.
2	Le système accepte l'accès de l'élève
3	L'élève demande de retirer le relevé des notes et choisit une période
6	Le système affiche le relevé des notes de l'élève de la période choisie
7	L'élève imprime le relevé des notes

Diagramme de séquence « Retirer les relevés des notes »

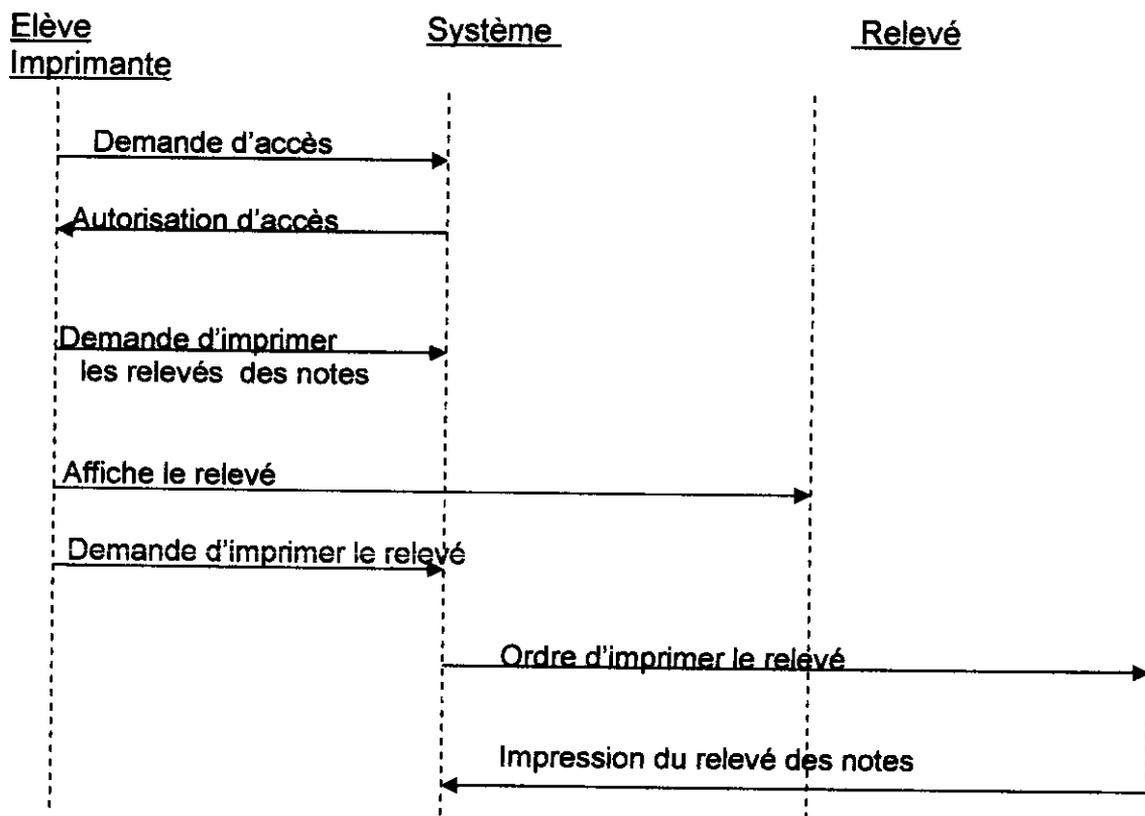
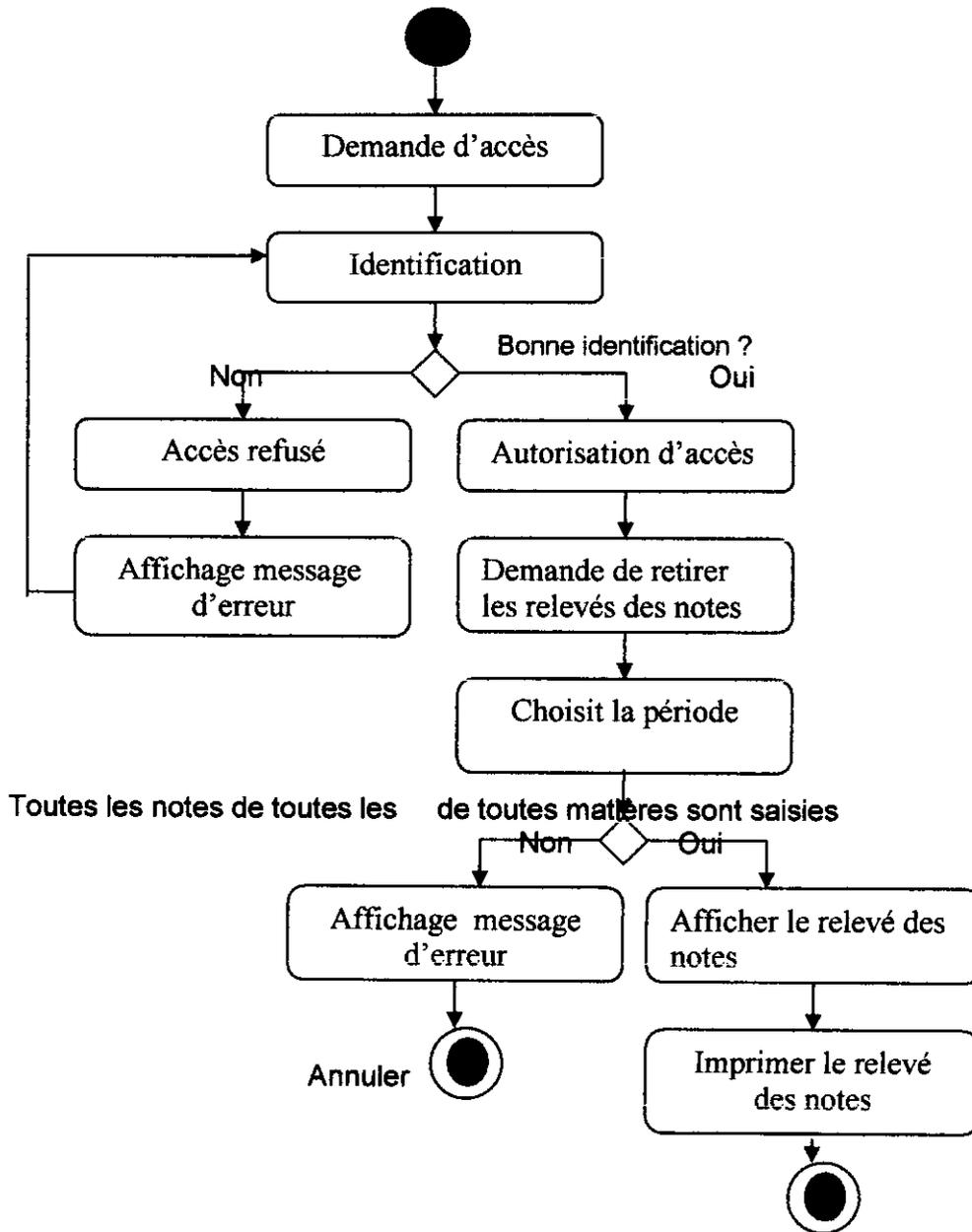


Diagramme d'activité « Retirer les relevés des notes »



Cas d'utilisation « Retirer le certificat de scolarité »

Généralité

Nom fiche de cas d'utilisation	Retir_certif_scol.php
Cas d'utilisation	Retirer le certificat de scolarité
Acteurs	Elève
But	Imprimer le certificat de scolarité
Description	Le système est dans l'état : « Retirer certificat de scolarité », L'élève peut à tout moment imprimer son certificat de scolarité
Pré conditions	l'élève est identifié
Post conditions	Le certificat de scolarité est affiché
Exception	

Scénario Nominal « Retirer les relevés des notes »

N° d'enchaînement	Actions acteur et Actions système
1	Un élève s'identifie pour accéder au système
2	Le système accepte l'accès de l'élève
3	L'élève demande de retirer son certificat de scolarité
6	Le système affiche le certificat de scolarité
7	L'élève imprime le certificat de scolarité

Diagramme de séquence « Retirer le certificat de scolarité »

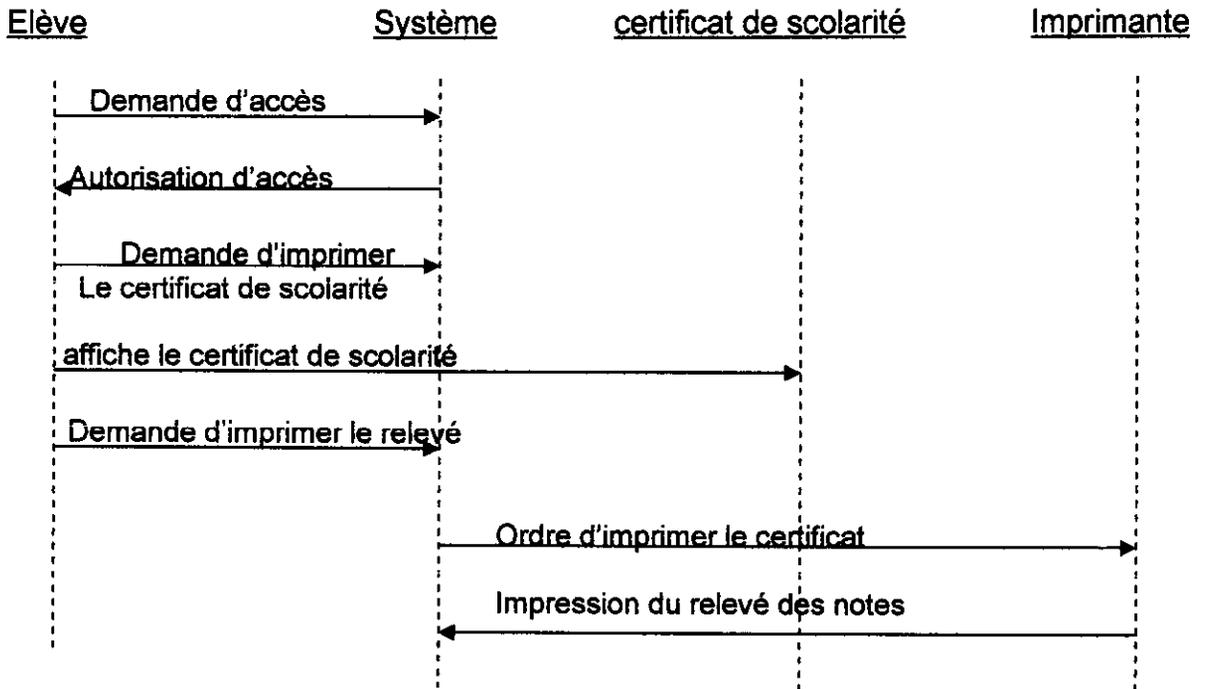
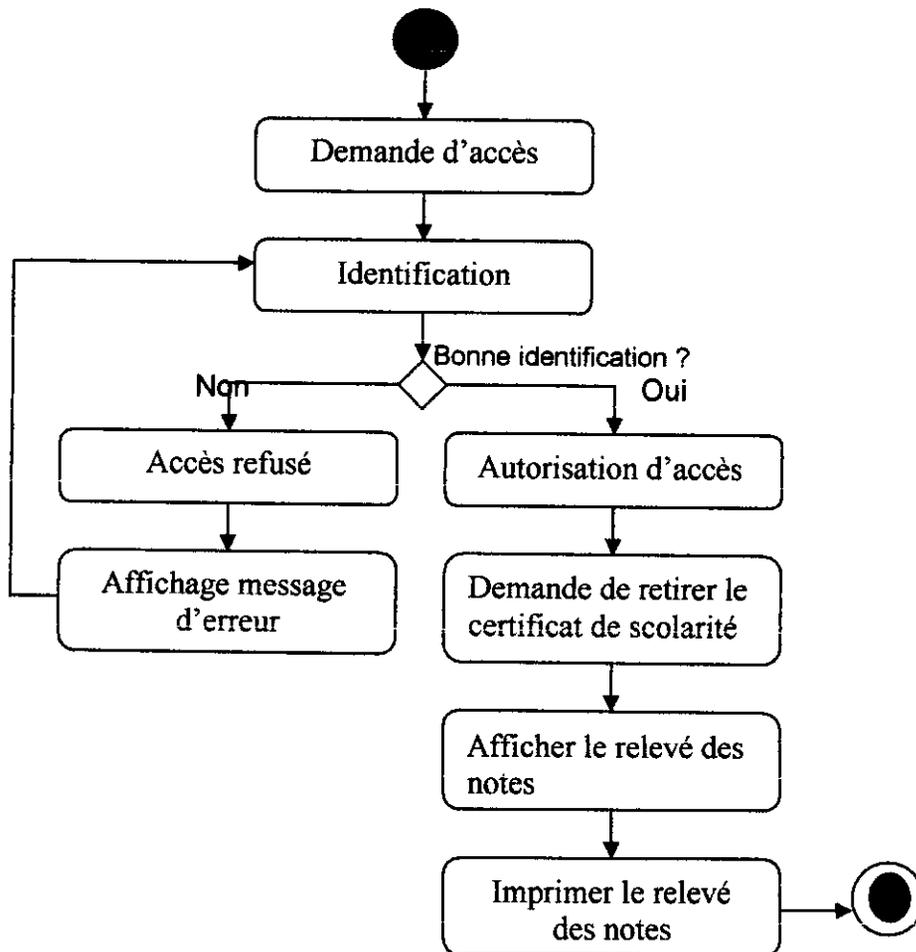


Diagramme d'activité « Retirer le certificat de scolarité »



Cas d'utilisation « Retirer le certificat de scolarité »

Généralité

Nom fiche de cas d'utilisation	Retir_certif_scol.php
Cas d'utilisation	Retirer le certificat de scolarité
Acteurs	Elève
But	Imprimer le certificat de scolarité
Description	Le système est dans l'état : « Retirer certificat de scolarité », L'élève peut à tout moment imprimer son certificat de scolarité
Pré conditions	l'élève est identifié
Post conditions	Le certificat de scolarité est affiché
Exception	Aucune

Scénario Nominal « Retirer les relevés des notes »

N° d'enchaînement	Actions acteur et Actions système
1	Un élève s'identifie pour accéder au système.
2	Le système accepte l'accès de l'élève
3	L'élève demande de retirer son certificat de scolarité
6	Le système affiche le certificat de scolarité
7	L'élève imprime le certificat de scolarité

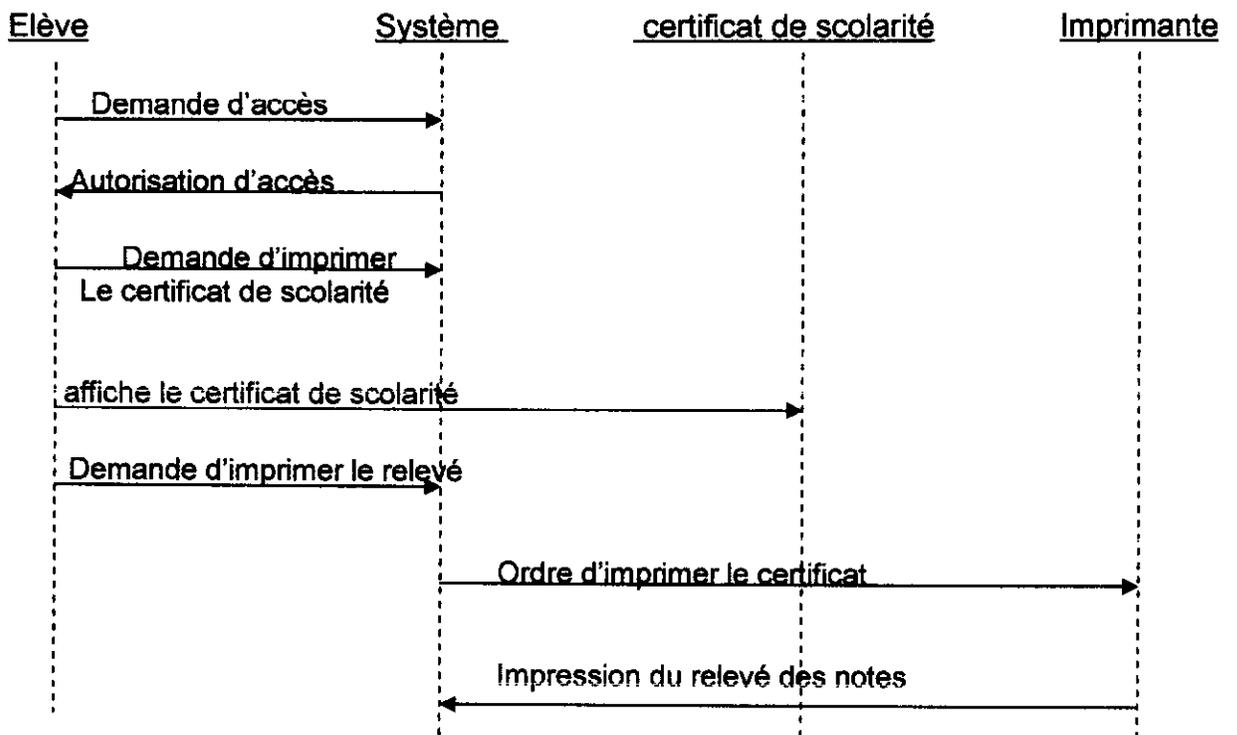
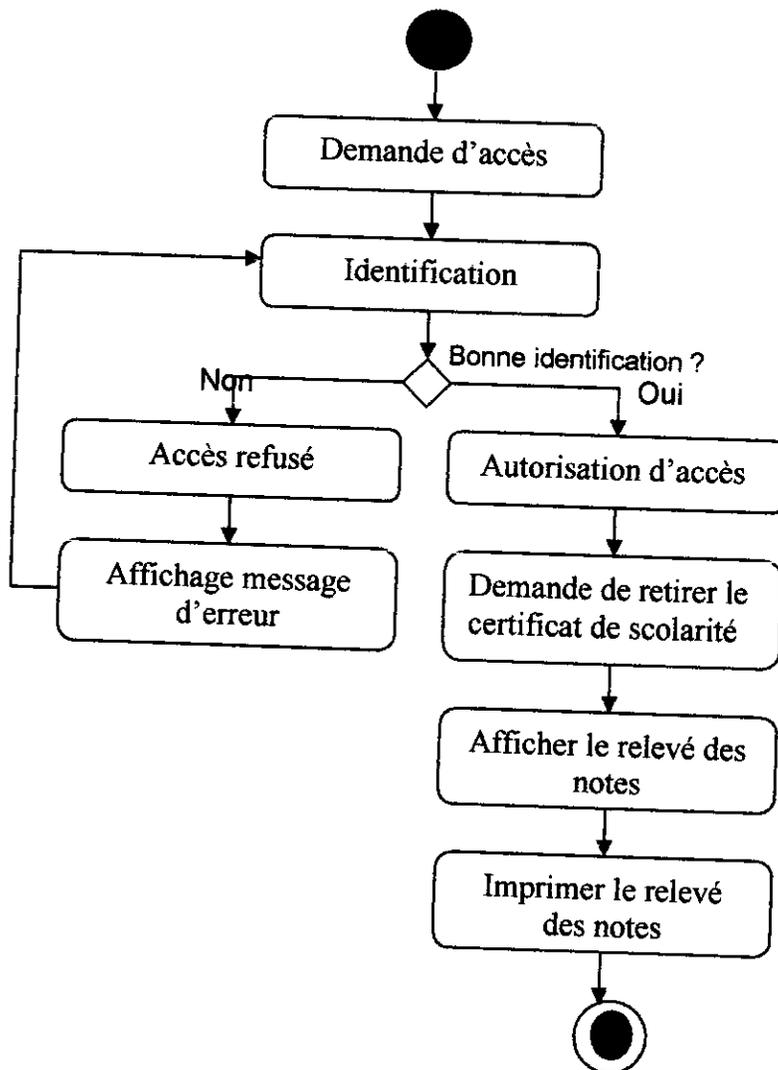
Diagramme de séquence « Retirer le certificat de scolarité »

Diagramme d'activité « Retirer le certificat de scolarité »



4.3.5 Communication

4.3.5.1 Affichage du parcours

Cas d'utilisation « Afficher le parcours pédagogique »

Généralité

Nom fiche de cas d'utilisation	affich_par.php
Cas d'utilisation	Afficher le parcours pédagogique
Acteurs	Direction
But	Affichage de toutes les notes de l'élève de tous les établissements par lesquels il est passé
Description	Le système est dans l'état : « Afficher parcours » La direction demande d'afficher le parcours d'un élève, le système se connecte au serveur central et déduit d'après l'identification globale de l'élève tous les établissements par lesquels il est passé et affiche toutes les notes de ces établissements
Pré conditions	La direction est identifiée, la connexion est établie au serveur central
Post conditions	Le parcours est affiché
Exception	

Scénario Nominal « Afficher le parcours pédagogique »

N° d'enchaînement	Actions acteur et Actions système
1	La direction s'identifie pour accéder au système.
2	Le système accepte l'accès de la direction
3	La direction demande d'afficher le parcours
4	Le système affiche un formulaire pour saisir le matricule de l'élève
5	La direction remplit le formulaire
6	Le système se connecte au serveur central
7	Le système affiche le parcours

Diagramme de séquence « Afficher le parcours pédagogique »

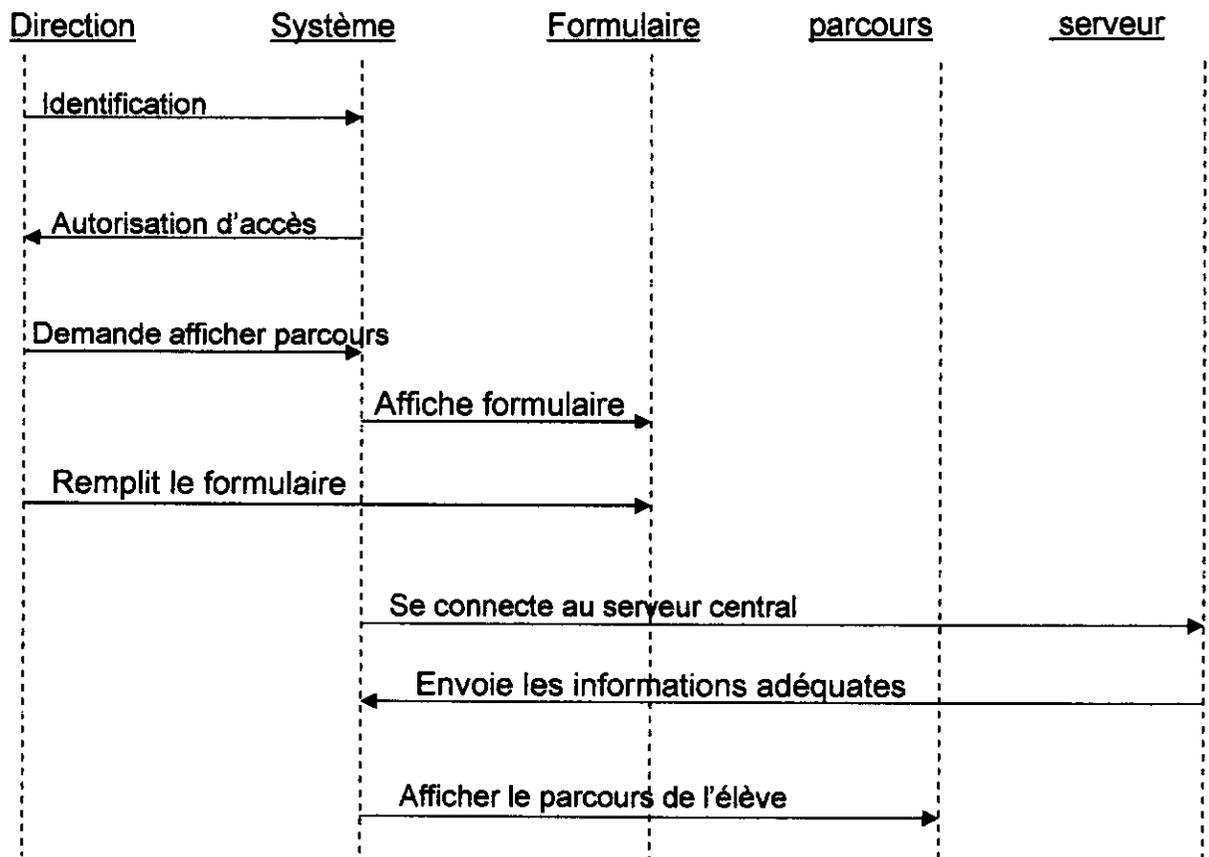
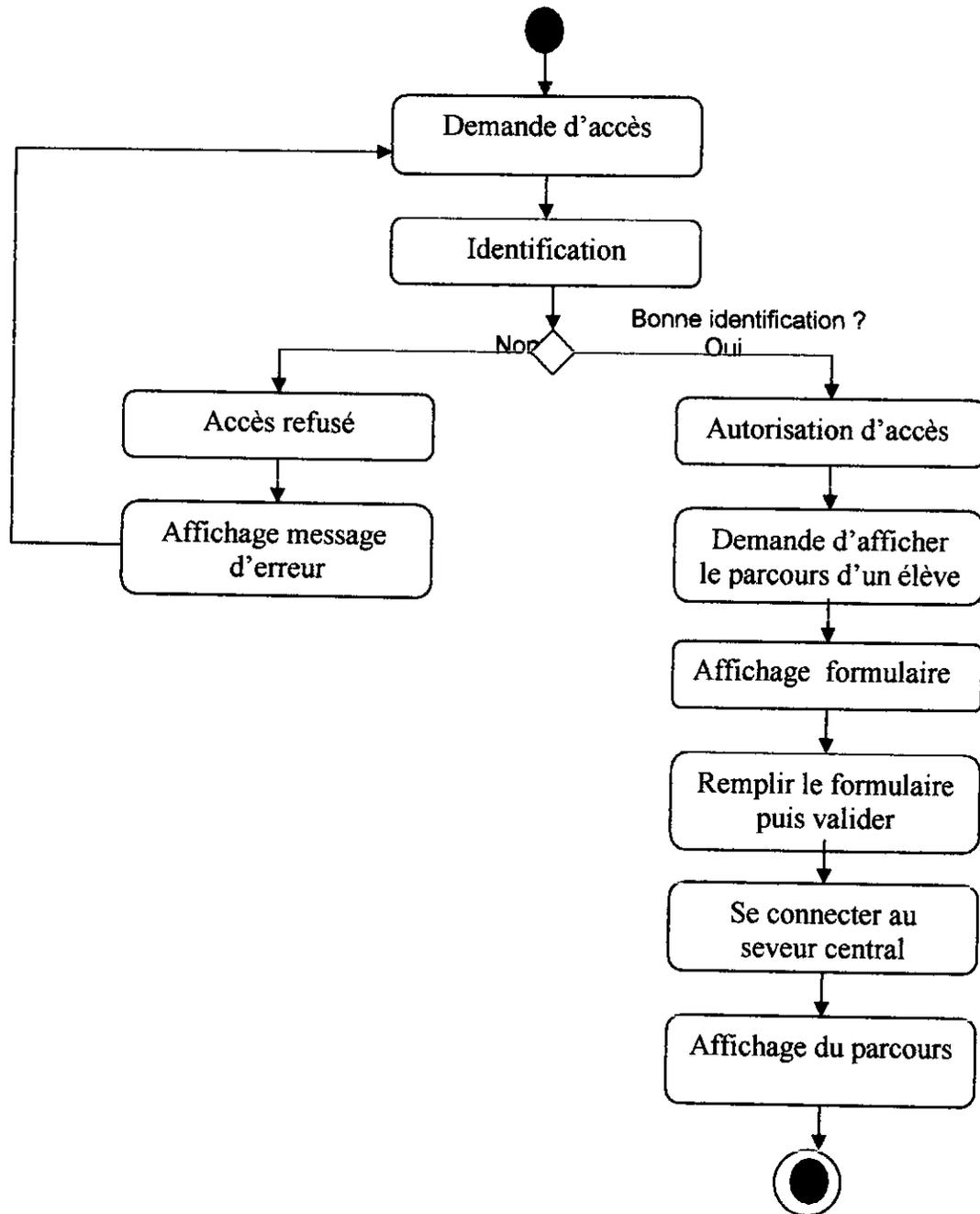


Diagramme d'activité « Afficher le parcours pédagogique »



4.3.5.2 Publication

Cas d'utilisation « Publier »

Généralité

Nom fiche de cas d'utilisation	publier.php
Cas d'utilisation	Publier
Acteurs	Enseignant
But	Mettre à jour ses modifications pendant qu'il est off ligne dans le serveur central
Description	Le système est dans « Publier » L'enseignant peut saisir ses notes tout en étant off ligne et quand il se connecte il pourrait envoyer ses modifications à son serveur et recevoir les informations qui one été saisies pendant qu'il était off ligne
Pré conditions	L'enseignant est identifié, connecté à Internet
Post conditions	
Exception	

Scénario Nominal

N° d'enchaînement	Actions acteur et Actions système
1	Un enseignant demande l'accès au système.
2	Le système accepte l'accès de l'enseignant.
3	L'enseignant demande de Publier
4	Le système essaie de se connecter au serveur de l'enseignant, puis effectue les mises à jour nécessaires

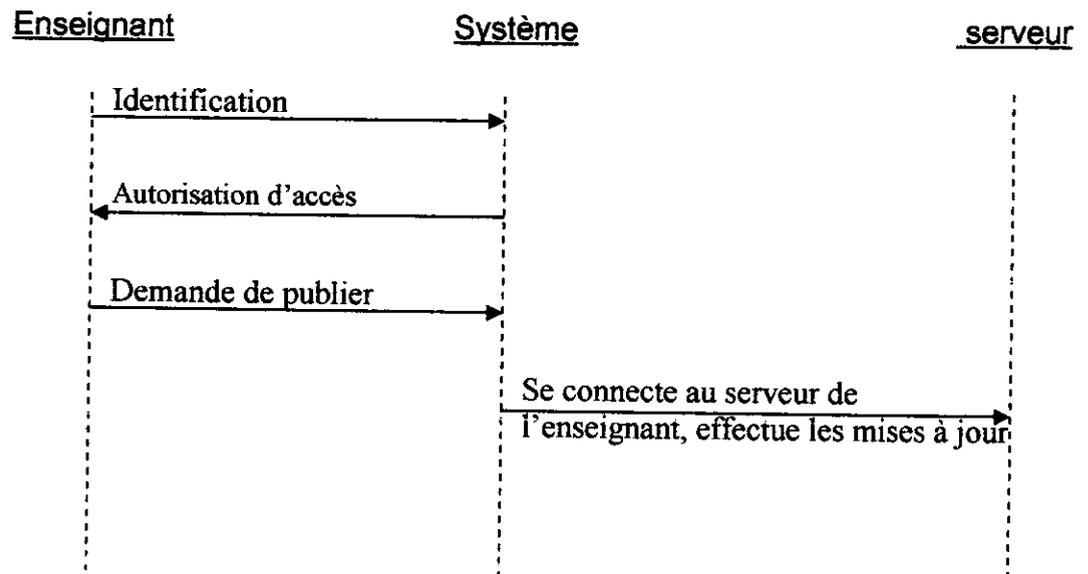
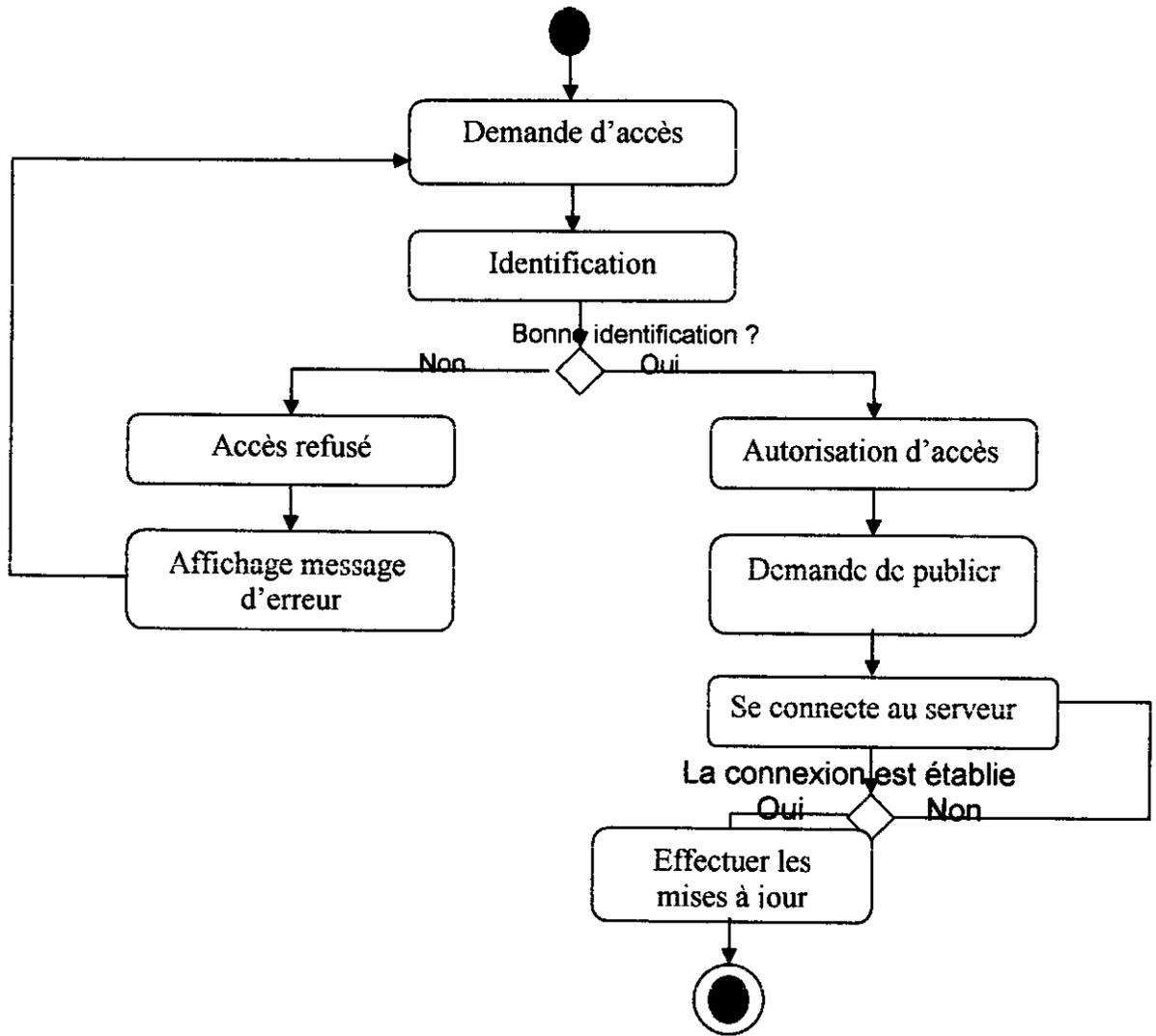
Diagramme de séquence « Publier »

Diagramme d'activité « Publier »



4.3.5.3 Messagerie

Cas d'utilisation « messagerie »

Généralité

Nom fiche de cas d'utilisation	messagerie.php
Cas d'utilisation	Utiliser la messagerie
Acteurs	utilisateur
But	Permettre a l'utilisateur de consulter, rédiger des messages
Description	<p>Le système est dans l'état : « Messagerie »</p> <p>L'utilisateur peut à tout moment rédiger un message, Le système lui affiche un formulaire nécessaire puis l'utilisateur valide et envoie le message</p> <p>L'utilisateur peut consulter ses messages envoyés et reçus,</p> <p>Le système affiche tous les messages</p> <p>L'utilisateur peut répondre à un message reçu</p> <p>Le système fourni un formulaire adéquat</p>
Pré conditions	L'utilisateur est identifié
Post conditions	
Exception	

Scénario Nominal

N° enchaînement	Actions acteur et Actions système
1	Un utilisateur demande l'accès au système
2	Le système vérifie l'existence de l'utilisateur

écriture d'un message

3	l'utilisateur demande d'écrire message
4	Le système affiche le formulaire d'écriture
5	L'utilisateur écrit message et valide

Consulter les messages reçus

3	l'utilisateur demande de consulter tous ses messages
4	Le système affiche tous les messages
5	L'utilisateur demande de lire l'un des messages
6	Le système affiche le message et l'émetteur du message

Supprimer un message

3	l'utilisateur demande de supprimer un message
4	Le système supprime le message choisi

Diagramme de séquence « messagerie »

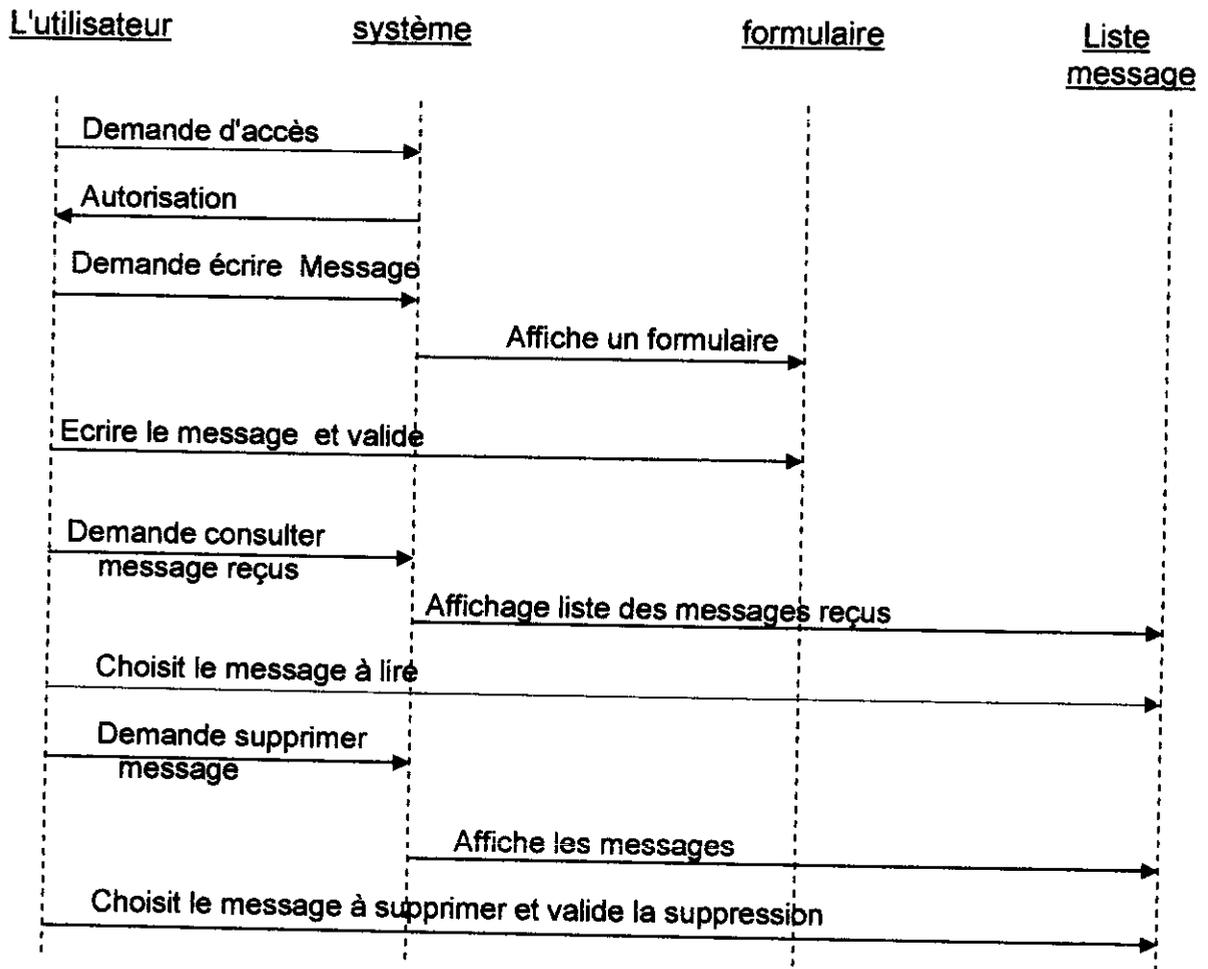
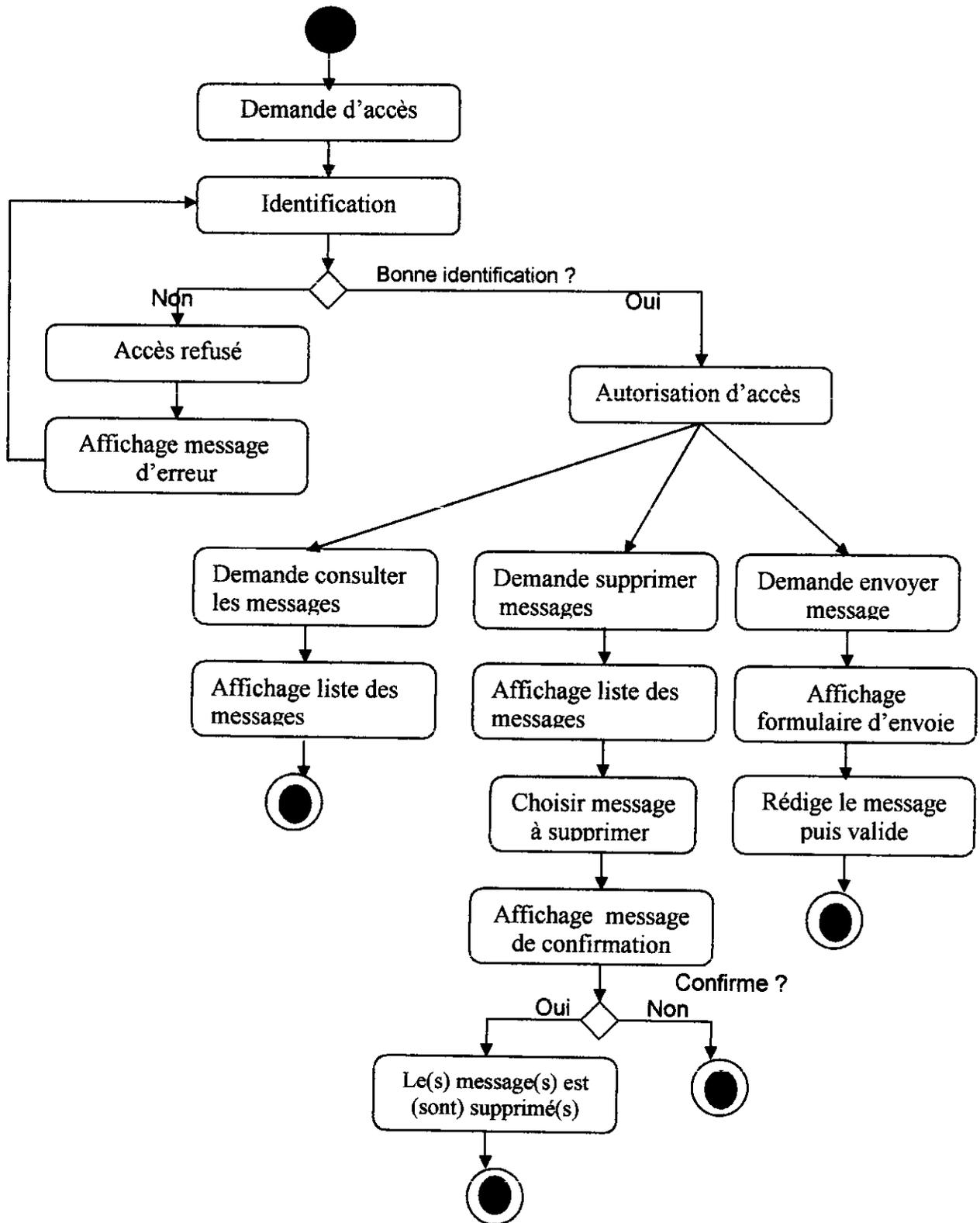
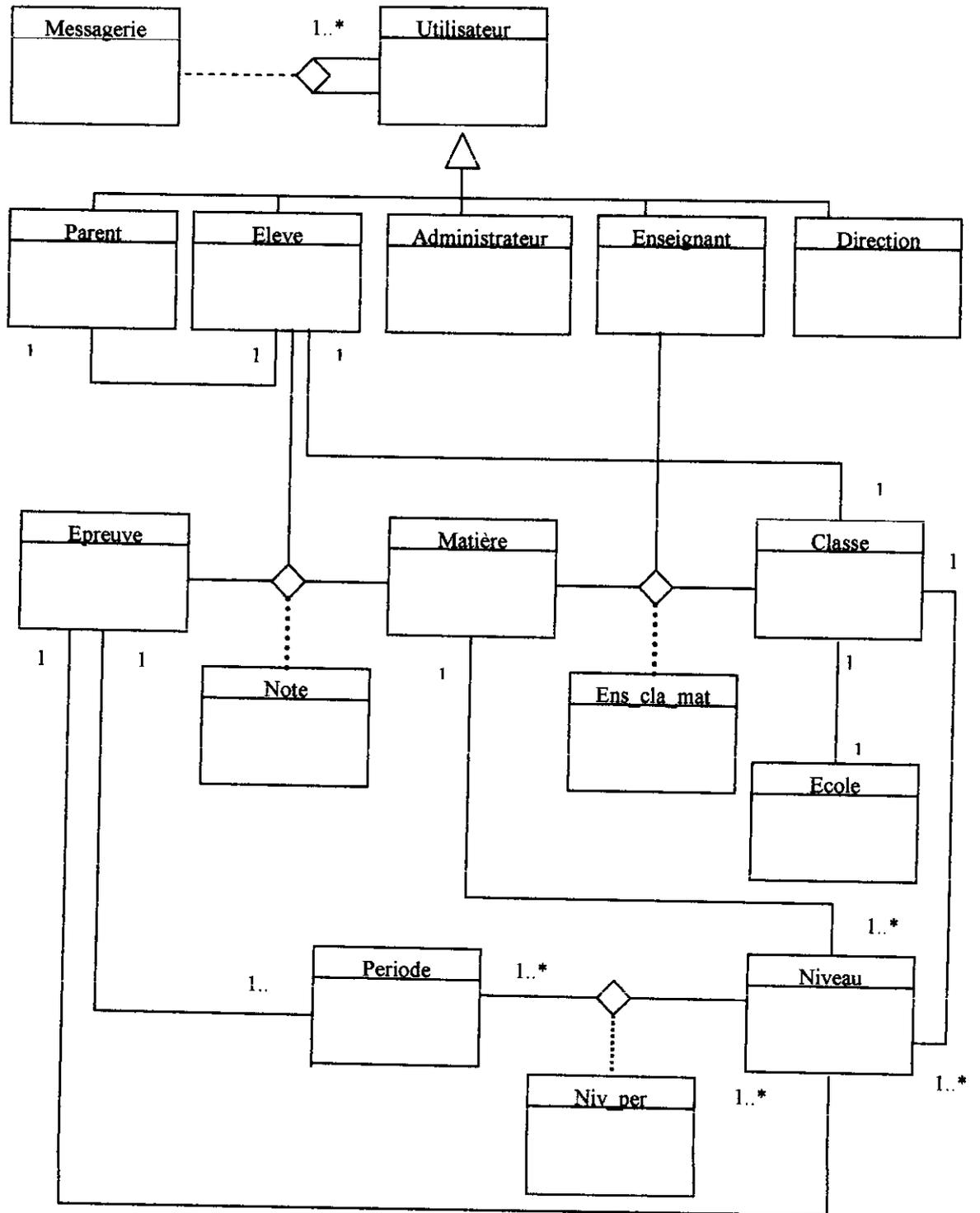


Diagramme d'activité « messagerie »



4.4 Diagramme de classe



4.5 Description du diagramme de classe

Nom Classe	Attributs de la classe	Désignation des attributs
Eleve	Elev_id nom prenom jour_nais mois_nais annee_nais Lieu_nai nationalite sexe adresse classe_id photo Img_taille etat Date_sup Nbr_db inscription Utili_id Ident_glob parent_id	Matricule local de l'élève Nom de l'élève prénom de l'élève Jour de naissance Mois de naissance Année de naissance de l'élève Lieu de naissance Nationalité Sexe de l'élève Adresse Code de sa classe Le chemin de sa photo Taille de sa photo Son état : supprimé ou pas Date de sa suppression Nombre de son doublement de l'année actuelle Inscrit pour la première fois (oui/non) Code utilisateur Identification globale Code parent
Enseignant	Ens_id nom prenom jour_nais mois_nais annee_nais sexe Adresse Num_tel E_mail specialite photo Img_taille autor Utili_id	Code enseignant Nom de l'enseignant prénom Jour de naissance Mois de naissance Année de naissance de l'élève Sexe de l'enseignant Adresse Numéro de téléphone Adresse email Sa spécialité Le chemin de sa photo Taille de la photo autorisation Code utilisateur
utilisateur	utili_id login mdp type_utili	Code utilisateur login mot de passe Type utilisateur
matière	code_mat lib_mat coef_mat niveau_id	Code matière Libellé matière Coefficient de la matière Code niveau

epreuve	epreuve_id lib_epreuve periode_id poix niveau_id	Code épreuve Libellé épreuve Période de l'épreuve Poix de l'épreuve Code niveau
niveau	niveau_id lib_niv Nbr_classes	Code niveau Libellé niveau Nombre de classes du niveau
classe	classe_id lib_cle nbr_elev niveau_id	Code classe Libellé classe Nombre élèves Code niveau
parent	parent_id nom prenom utili_id	Code parent Nom parent Prénom parent Code utilisateur
ecole	code_ecole nom_ecole annee code_wilaya cloture	Code école Nom école Année Code wilaya Année clôturée (oui/non)
periode	periode_id	Code période
ens_cla_mat	ens_id classe_id matiere_id	Code enseignant Code classe Code matière
note	Elev_id Matiere_id Epreuve_id Note date	Code élève Code matière Code épreuve Note Date
messagerie	id-exp id_dest id_mess date heure sujet texte lu type	Code expéditeur Code destinataire Code message Date du message Heure du message Sujet Texte message Lu (oui/non) type
niv_per	Niveau_id Periode_id	Code niveau Code période

Réalisation Chapitre 5

Réalisation

La réalisation joue un rôle essentiel dans le développement du système d'information. Elle résulte des spécifications décrites lors de la conception, et vise la production d'un outil capable d'effectuer les traitements automatisés.

Conformément à la solution informatique retenue, l'application sera développée avec le langage de programmation PHP, et le système de gestion de base de données MySQL.

5.1 Architecture trois-tiers

Le principe d'une architecture trois-tiers est relativement simple. Il consiste à séparer la réalisation de ces trois parties :

- **Stockage et accès aux données**

Il a pour but de conserver une quantité plus ou moins importante de données de façon structurée. Pour cette partie des systèmes très variés sont utilisés qui peuvent être des systèmes de fichiers, des mainframes, des systèmes de bases de données relationnelles, etc... Le point commun entre tous ces systèmes est qu'ils permettent le partage des données qu'ils contiennent via un réseau. La méthode d'accès à ces données dépendra du type d'organisation de ces données.

- **Logique applicative**

Cette logique constitue les traitements nécessaires sur l'information afin de la rendre exploitable par chaque utilisateur. Les utilisateurs peuvent avoir des besoins très variés et évolutifs. Il devient alors nécessaire de permettre l'évolution du système sans pour autant devoir tout reconstruire. Cette partie utilise les données pour les présenter de façon exploitable par l'utilisateur. Il convient donc de bien identifier les besoins des utilisateurs afin de réaliser une logique applicative utile tout en structurant les données utilisées.

- **Présentation**

La présentation est la partie la plus immédiatement visible pour l'utilisateur. Elle a donc une importance primordiale pour rendre attrayante l'utilisation de l'informatique.

La séparation de ces trois parties signifie qu'il est possible de déployer chaque partie sur un serveur indépendant, toutefois cela n'est pas obligatoire. La mise en place de ce type d'architecture permet dans tous les cas une plus grande évolutivité du système. Il est ainsi possible de commencer par déployer les deux serveurs sur la même machine, puis de déplacer le serveur applicatif sur une autre machine lorsque la charge devient excessive. Les éléments permettant la réalisation classique d'un système en architecture trois tiers sont les suivants:

- système de base de données relationnel (SGBDR) pour le stockage des données
- serveur applicatif pour la logique applicative
- navigateur web pour la présentation

Il est important de remarquer que l'essentiel du travail de développement sera implanté au niveau du serveur applicatif. Le SGBDR nécessitera un travail d'administration surtout dans le cas d'une quantité de données importante. Le travail de conception de la base de données sera la pierre angulaire du système. En effet l'ensemble du développement s'appuiera sur cette conception. Le navigateur web nécessitera la programmation de code spécifique permettant de gérer l'affichage par ce navigateur. Ce code sera placé sur le serveur applicatif pour permettre une mise à jour sans nécessiter de nouveaux déploiements.

5.2 Environnement de développement

5.2.1 Choix du langage de programmation

Notre choix du langage de programmation s'est porté sur le langage PHP et cela pour ces divers atouts :

- Le PHP est rapide : Compilé en tant que module Apache, les temps d'exécution sont très performants. Les scripts PHP sont exécutés par le serveur web, sans ressources supplémentaires.
- Le PHP est un langage facile à apprendre : Il reprend assez fidèlement la syntaxe du C, ce qui rend sa prise en main par un développeur qui connaît ce langage quasi immédiate.
- Le PHP gère très bien, depuis les origines, les requêtes SQL : On peut donc facilement écrire des programmes qui affichent des données extraites de bases SQL, ou qui stockent des données postées par un formulaire dans une table SQL. Le PHP sait communiquer avec presque tous les SGBD de la création (Oracle, MySql, DB2, Informix, Ingres, Postgresql, SQL Server, Access etc...).
- Aux instructions de bases du PHP, s'ajoutent une foule de fonctions, couvrant tous les besoins imaginables pour un webmaster.

5.2.2 Choix de l'éditeur de site web

Macromedia Dreamweaver est le choix des développeurs pour développer des sites et des applications web. Macromedia Dreamweaver offre une puissante combinaison d'outils visuels, de fonctions de développement d'applications et de possibilités de manipulation du code aux développeurs et aux designers de tous niveaux qui souhaitent créer rapidement des sites et des applications web. Avec ses fonctions de mise en page CSS et de programmation manuelle, Dreamweaver propose les outils dont les professionnels ont besoin, dans un environnement intégré et optimisé. Les développeurs peuvent utiliser Dreamweaver avec les technologies serveur de leur choix pour créer de puissantes applications Internet qui se connectent à

des bases de données, des flux de données en direct et des systèmes d'entreprise.

5.2.3 Choix du SGBD

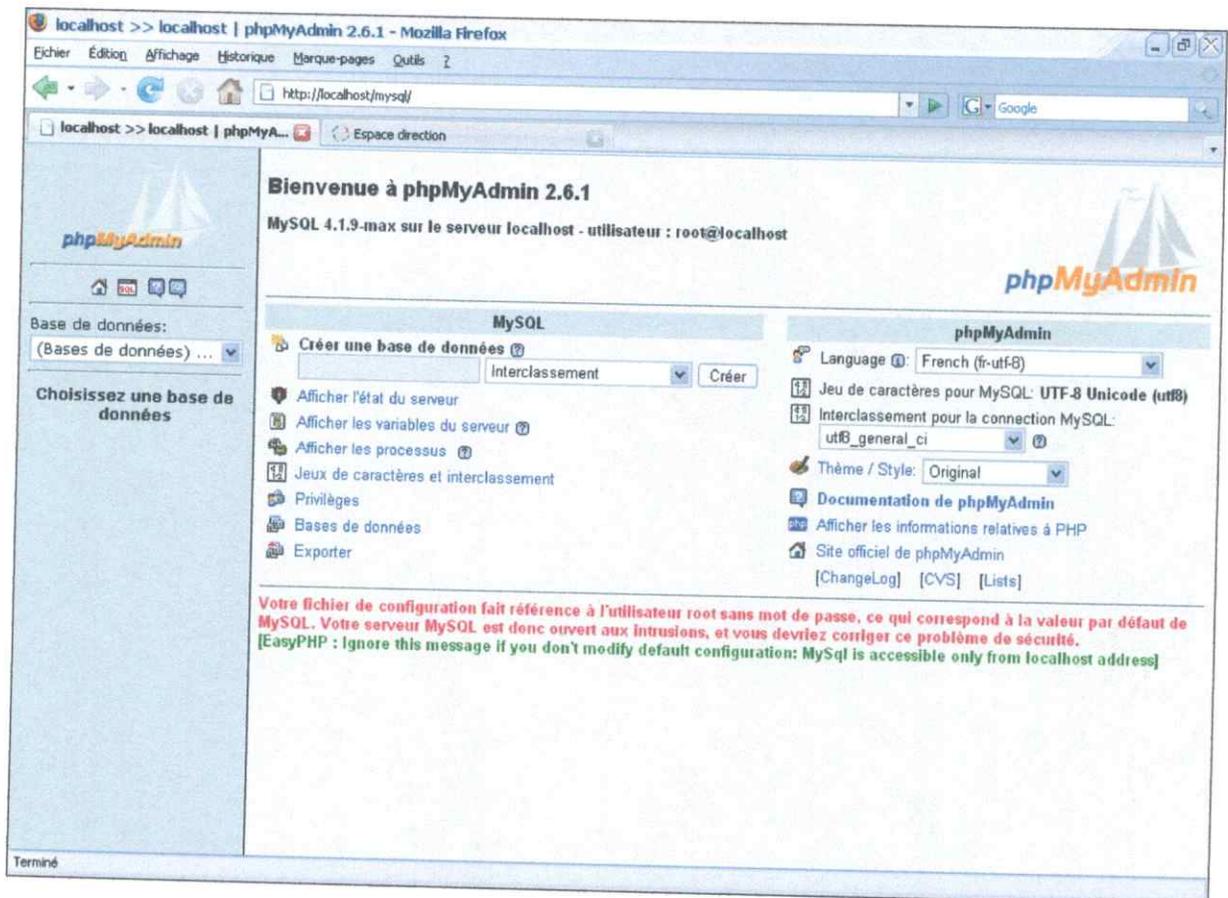
Un serveur de bases de données MySQL est très rapide, fiable et facile à utiliser, il fonctionne sous la plupart des systèmes d'exploitation comme Unix, Linux et Windows NT/2000/XP. Le logiciel MySQL a l'avantage d'être gratuit et hautement adapté au Web.

L'un des points forts de MySQL est qu'il est un SGBD de type relationnel, c'est à dire qu'il organise les données selon des tables comportant des champs à attributs simples et monovalués. Il offre ainsi une simplicité de structure des données, une simplicité des opérateurs, et l'indépendance entre les données et les applications.

Présentation du SGBD :

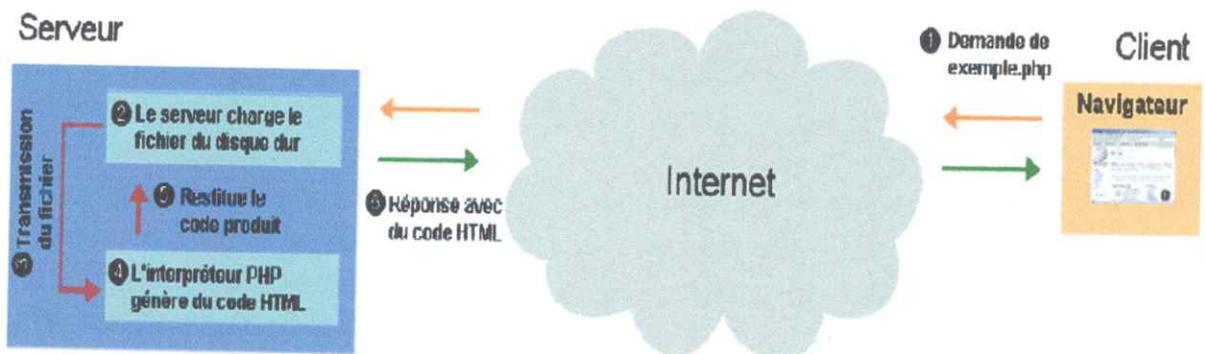
phpMyAdmin est un excellent outil d'administration pour gérer une base de données MySQL. Plutôt que de coder toutes les requêtes sql en langage php, phpMyAdmin permet de gérer les bases de données directement depuis le navigateur. créer les bases, y modifier les tables, configurer les champs, exécuter des requêtes SQL, etc.

L'outil phpMyAdmin 2.6.1 :



Développer un site en PHP/MySQL, permet de gagner un temps considérable. En travaillant directement dans le répertoire du site local, les modifications sont instantanées. Le résultat sera sensiblement le même que celui qu'on obtiendra chez l'hébergeur. Cela est pratique pour tester rapidement des requêtes sans détruire tout le travail. Cependant, un problème est fréquemment rencontré : tous les liens doivent être relatifs, pour toujours travailler localement et ne pas mélanger le site en ligne et sa version de test, en local. Débrancher le câble réseau et se balader sur le site en local constitue un bon test

EasyPHP est un paquetage contenant à la fois Apache, PHP et MySQL. Il permet d'installer automatiquement et facilement une plateforme permettant l'exploitation d'un site web en PHP qui éventuellement aurait besoin d'un accès à une base de données. De la sorte on se libère des complications d'une installation manuelle de chacune de composantes séparément



5.3 L'implémentation du diagramme de classes

Classe	Attributs	type	Désignation	Méthodes
Eleve	Elev_id nom prenom jour_nais mois_nais annee_nais Lieu_nai nationalite sexe adresse classe_id photo Img_taille etat Date_sup Nbr_db inscription Utili_id Ident_glob parent_id	Caractère Caractère Caractère Entier Caractère Entier Caractère Caractère Caractère Entier Caractère Entier Binaire Date Entier Binaire Entier Entier Entier	Matricule local de l'élève Nom de l'élève prénom de l'élève Jour de naissance Mois de naissance Année de naissance de l'élève Lieu de naissance Nationalité Sexe de l'élève Adresse Code de sa classe Le chemin de sa photo Taille de sa photo Son état : supprimé ou pas Date de sa suppression Nombre de son doublement de l'année actuelle Inscrit pour la première fois (oui/non) Code utilisateur Identification globale Code parent	Consulter note () Imprimer relevés notes () Imprimer certificate solarise () Messagerie ()
Enseignant	Ens_id nom prenom jour_nais mois_nais annee_nais sexe Adresse Num_tel E_mail specialite photo Img_taille autor Utili_id	Entier Caractère Caractère Entier Entier Entier Caractère Caractère Caractère Caractère Caractère Entier Binaire Entier	Code enseignant Nom de l'enseignant prénom Jour de naissance Mois de naissance Année de naissance de l'élève Sexe de l'enseignant Adresse Numéro de téléphone Adresse email Sa spécialité Le chemin de sa photo Taille de la photo autorisation Code utilisateur	Saisir note () Consulter ses notes ()
utilisateur	utili_id login mdp type_utili	Entier Caractère Caractère Caractère	Code utilisateur login mot de passe Type utilisateur	

matière	code_mat lib_mat coef_mat niveau_id	Caractère Caractère Entier Entier	Code matière Libellé matière Coefficient de la matière Code niveau	
epreuve	epreuve_id lib_epreuve periode_id poix niveau_id	Entier Caractère Entier Entier Entier	Code épreuve Libellé épreuve Période de l'épreuve Poix de l'épreuve Code niveau	
niveau	niveau_id lib_niv Nbr_classe	Entier Caractère Entier	Code niveau Libellé niveau Nombre de classes du niveau	
classe	classe_id lib_cle nbr_elev niveau_id	Entier Caractère Entier Entier	Code classe Libellé classe Nombre élèves Code niveau	
parent	parent_id nom prenom utili_id	Entier Caractère Caractère Entier	Code parent Nom parent Prénom parent Code utilisateur	Consulter notes eleve) Retirer certificat () Retirer releves de noesà ()
ecole	code_ecole nom_ecole annee code_wilaya cloture	Entier Caractère Caractère Entier Binaire	Code école Nom école Année Code wilaya Année clôturée (oui/non)	
periode	periode_id	Entier	Code période	
ens_cla_ma t	ens_id classe_id matiere_id	Entier Entier Entier	Code enseignant Code classe Code matière	
note	Elev_id Matiere_id Epreuve_id Note date	Entier Entier Entier Entier date	Code élève Code matière Code épreuve Note Date	Saisir note () Modifier note)
messagerie	id-exp id_dest	Entier Entier	Code expéditeur Code destinataire	Envoyer message ()

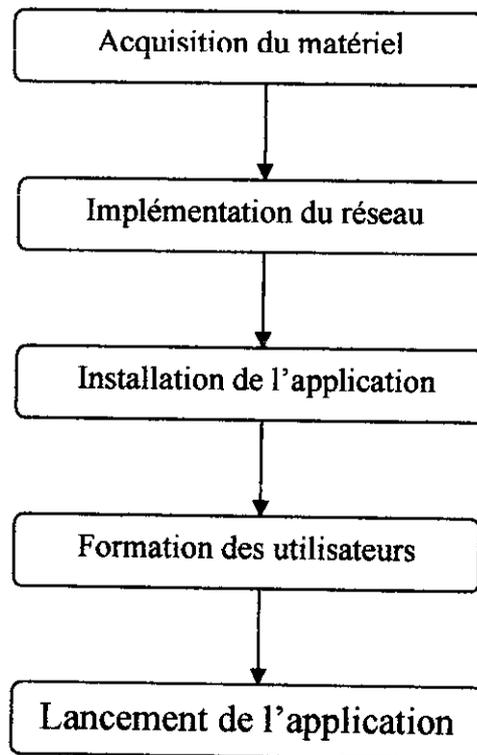
	id_mess date heure sujet texte lu type	Entier Date Entier Caractère Caractère Binaire Caractère	Code message Date du message Heure du message Sujet Texte message Lu (oui/non) type	Lire message () Répondre message ()
niv_per	Niveau_id Periode_id	Entier Entier	Code niveau Code période	

5.4 La mise en œuvre du système

Après la réalisation du logiciel et son test, on entame l'étape de mise en œuvre.

La mise en œuvre passe par les étapes suivantes :

- Acquisition du matériel : Serveur, Hub, RJ45, câble,...etc.
- Licence d'utilisation du système d'exploitation.
- La mise en place du réseau.
- Installation de l'application sur les postes concernés.
- Formation des utilisateurs.



Enchaînement des étapes de la mise en œuvre

5.5 Configuration du système

Cette application fonctionne sous Windows. Elle est exploitable à partir d'un simple navigateur internet. Nous allons ci-dessous donner les étapes pour configurer le système :

- Installer EasyPHP dans les machines 'serveur' et mettre l'application (le dossier : projet) dans le dossier www de EasyPHP.

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

Pour lancer l'application, il y a deux manières :

- Taper directement dans le navigateur l'adresse suivante <http://localhost/projet/> ou alors <http://127.0.0.1/projet/>
- Cliquez droit sur l'icône de EasyPHP puis sur « Web local ». Une liste s'affiche. Cliquez sur « projet ».

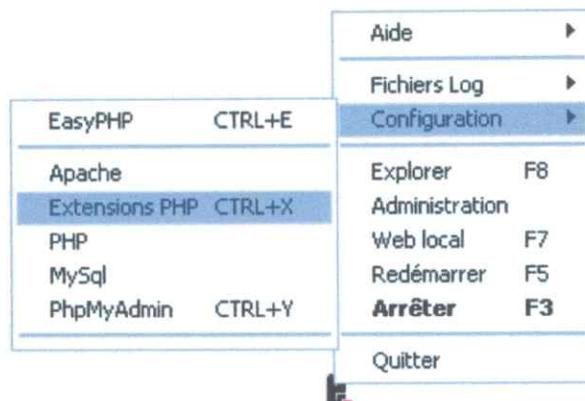
Cela est dans le cas où le travail se fait sur la même machine où EasyPHP est installé. Pour lancer l'application à distance, à partir d'une machine sur le réseau local ou sur internet, il faut remplacer « localhost » ou « 127.0.0.1 » par l'adresse de la machine où EasyPHP est installé.

Exemple :

- Dans le réseau local : <http://192.168.0.2/projet/>
Où « 192.168.0.2 » est l'adresse IP (dans le réseau) de la machine où EasyPHP est installé.
 - Sur internet : <http://87.119.92.164/projet/>
Où « 87.119.92.164 » est l'adresse IP (sur internet) de la machine où EasyPHP est installé.
- Cocher deux extensions nécessaires au bon fonctionnement de l'application, à savoir, l'extension gd et l'extension socket.
 - L'extension gd : Elle est nécessaire pour les fonctions de manipulation des images.
 - L'extension socket : Elle est nécessaire pour l'utilisation des sockets.

Ce qui suit montre comment activer ces deux extensions dans EasyPHP.

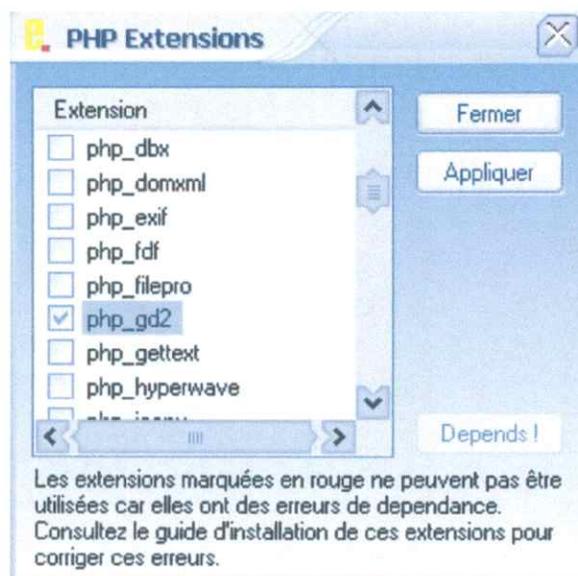
1 – Ouvrir les extensions PHP : clique droit de la souris sur l'icône de EasyPHP, pointer la souris sur « configuration » puis cliquer sur « Extension PHP ».



Ouvrir Extensions PHP

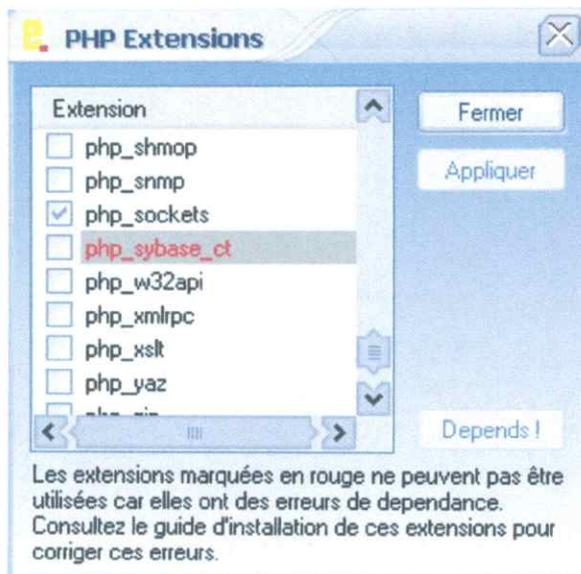
La fenêtre suivante s'ouvrira :

2 – Cocher l'élément « php_gd2 »



Extensions PHP

3 – Cocher l'élément « php_sockets » puis cliquer sur « Appliquer »



Extensions PHP

- Aucune autre contrainte n'est requise

5.6 Les modules de l'application

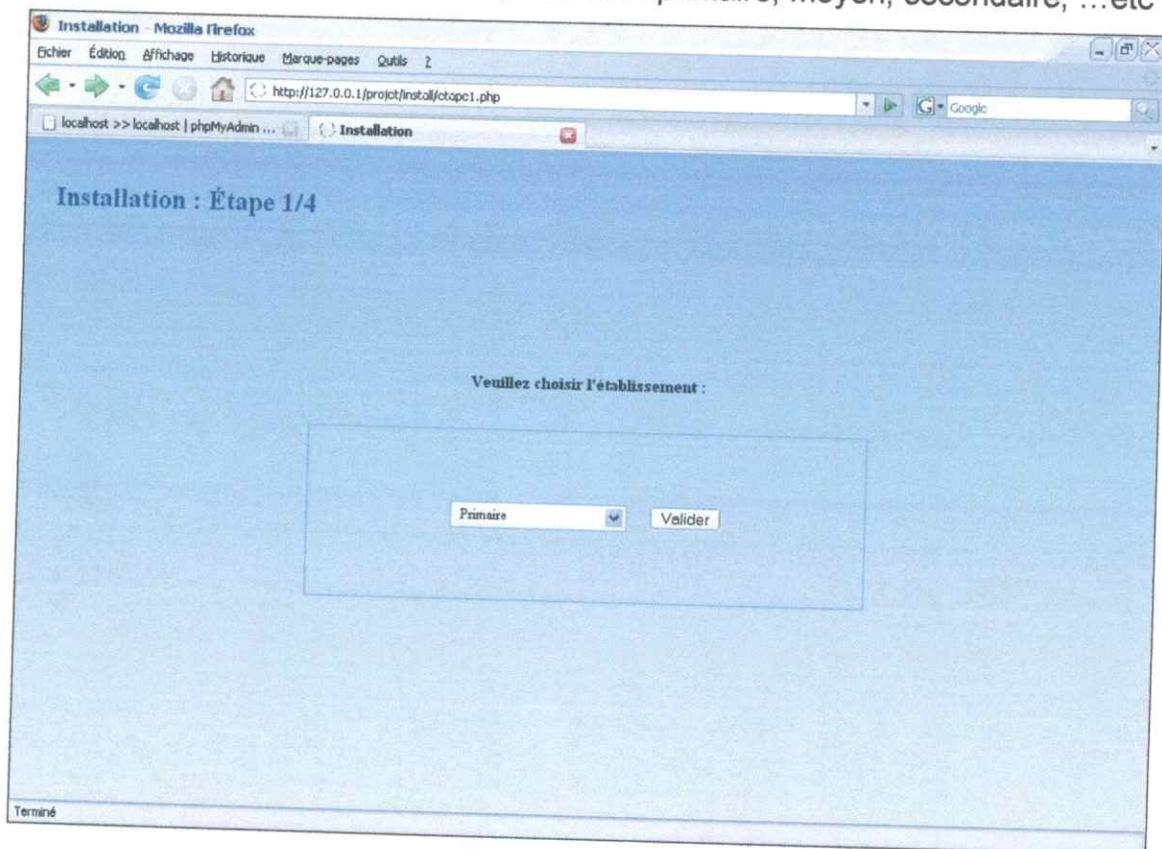
5.6.1 Module « Installation du système »

L'installation du système est l'étape où les principales informations de l'établissement sont définies. C'est l'administrateur qui s'occupe de l'installation.

Voici les étapes de l'installation :

Étape 1 :

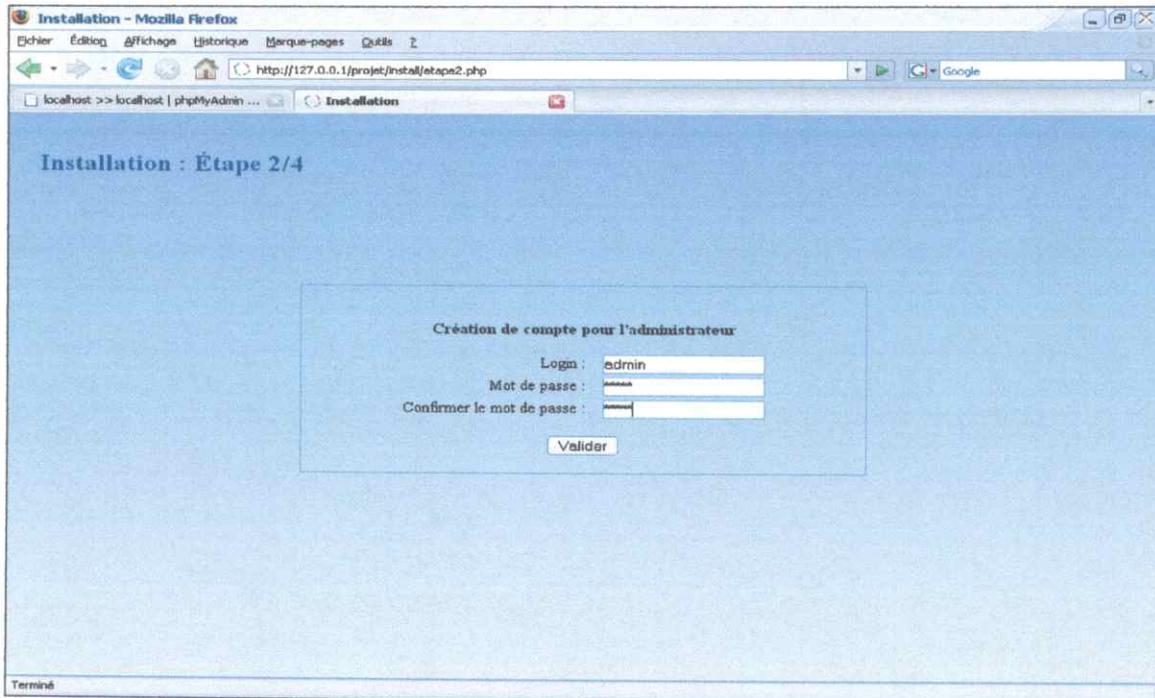
Choisir le type de l'établissement, à savoir : primaire, moyen, secondaire, ...etc



Installation : Etape 1/4

Étape 2 :

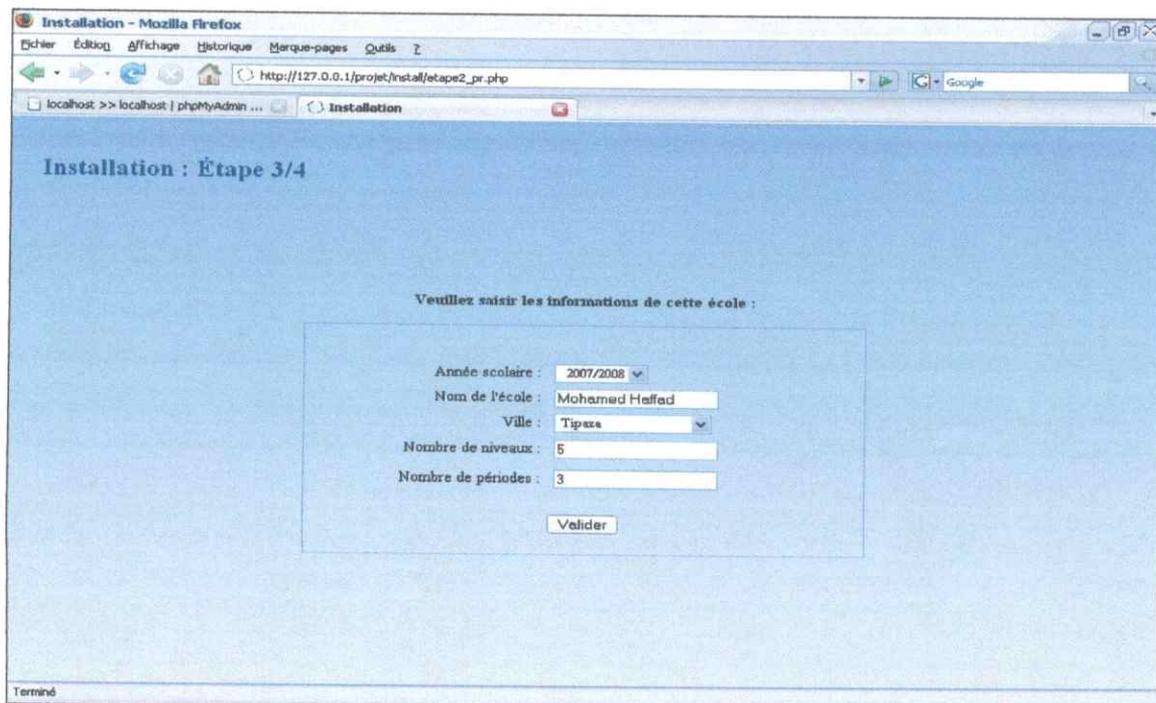
Création du compte de l'administrateur du système. L'administrateur doit choisir un login et un mot de passe qu'il va utiliser pour accéder à son espace.



Installation : Etape 2/4

Étape 3 :

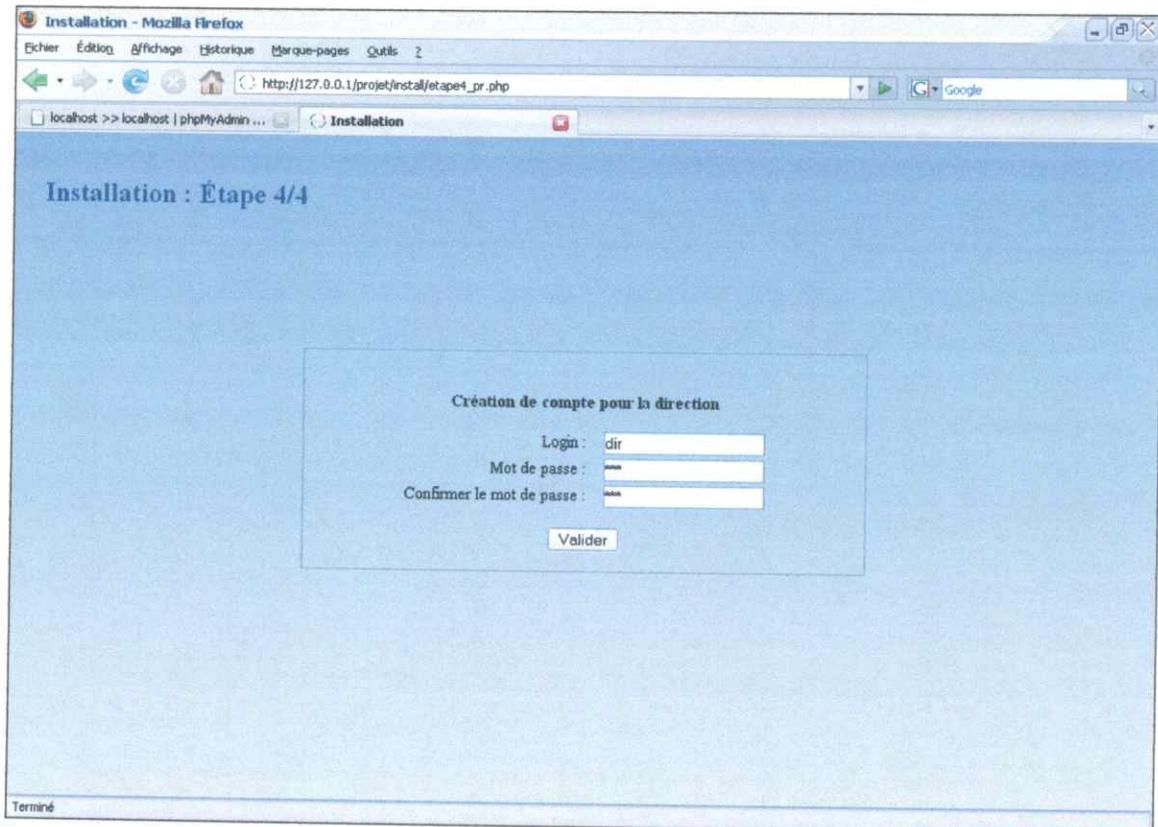
La définition des informations essentielles concernant l'établissement choisi. Dans ce cas une école primaire.



Installation : Etape 3/4

Étape 4 :

La création d'un compte pour la direction. Ce compte sera utilisé pour les principales fonctions de la scolarité. Il est possible de créer d'autres comptes pour la direction après l'installation.



Installation : Etape 4/4

5.6.2 Module « Authentification des utilisateurs »

Pour qu'un utilisateur puisse accéder à son espace, il faut qu'il fournisse son login et son mot de passe.

Le login est unique dans un même établissement. L'unicité du login est vérifiée avant la création d'un nouveau compte.

Code source : page de vérification avant la création de compte

```
<?php
@session_start();
$page1 = $_GET['page1'];
$page2 = $_GET['page2'];
include("ouvrir_bdd.php");
// Vérifier si tous les champs du formulaire sont remplis
if(empty($_POST['login'])or empty($_POST['mdp1'])or empty($_POST['mdp2']))
{
    $_SESSION['erreur']="Veuillez remplir tous les champs!";
    include($page1);
}
// Vérifier si les mots de passe sont identiques
elseif(strcmp($_POST['mdp1'],$_POST['mdp2']))
{
    $_SESSION['erreur']="Les mots de passe ne sont pas identiques!";
    include($page1);
}
else
{
    $login_entr = htmlentities($_POST['login']);
    $mdp_entr = htmlentities($_POST['mdp1']);
    $rep = mysql_query("SELECT mdp FROM utilisateur WHERE login='".$login_entr'");
    $ligne = mysql_num_rows($rep);
    // Vérifier si le login existe déjà dans la base de données
    if($ligne >0)
    {
        $_SESSION['erreur']="Ce login existe déjà. Choisissez en un autre!";
        include($page1);
    }
    else include($page2);
}
include("fermer_bdd.php");
?>
```

Le mot de passe est crypté dans un seul sens avec la fonction suivante :
string md5 (string str)

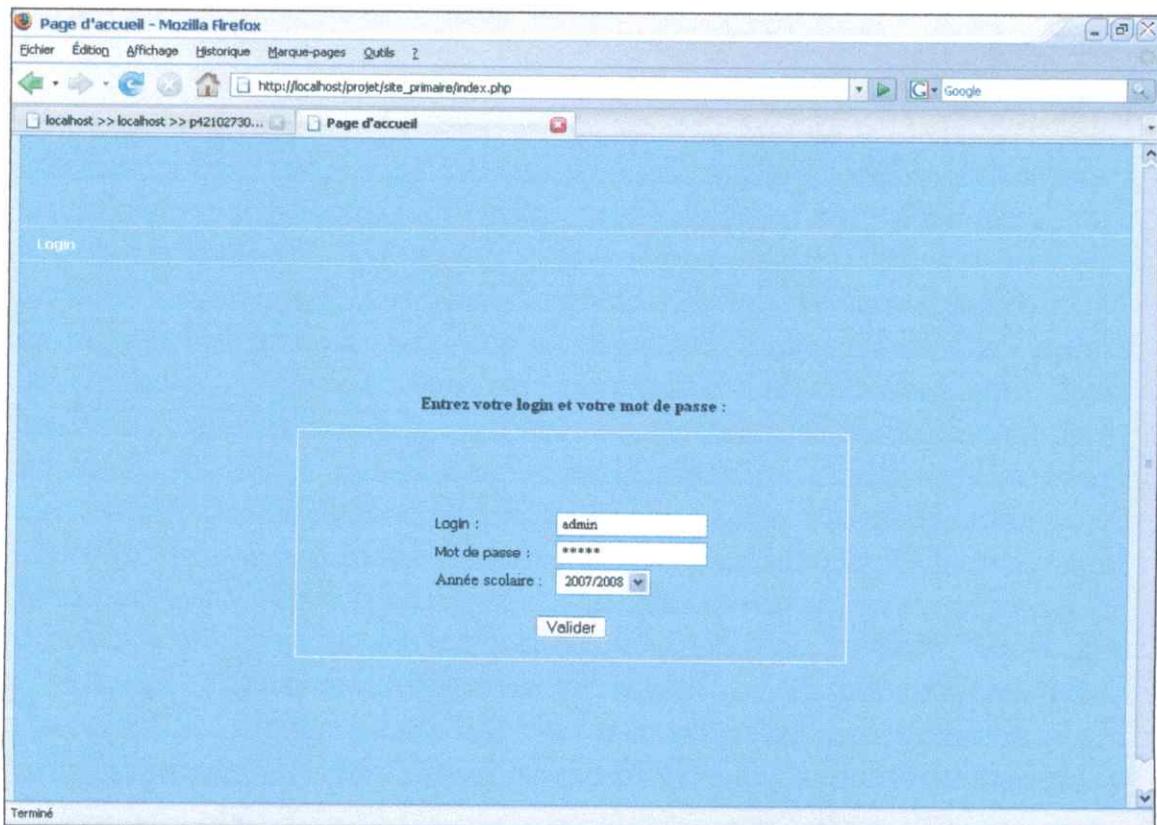
Cette fonction calcule le MD5 de la chaîne de caractères str en utilisant l'algorithme RSA Data Security, et retourne un nombre de 32 caractères hexadécimaux.

Code source : exemple de création d'un compte direction

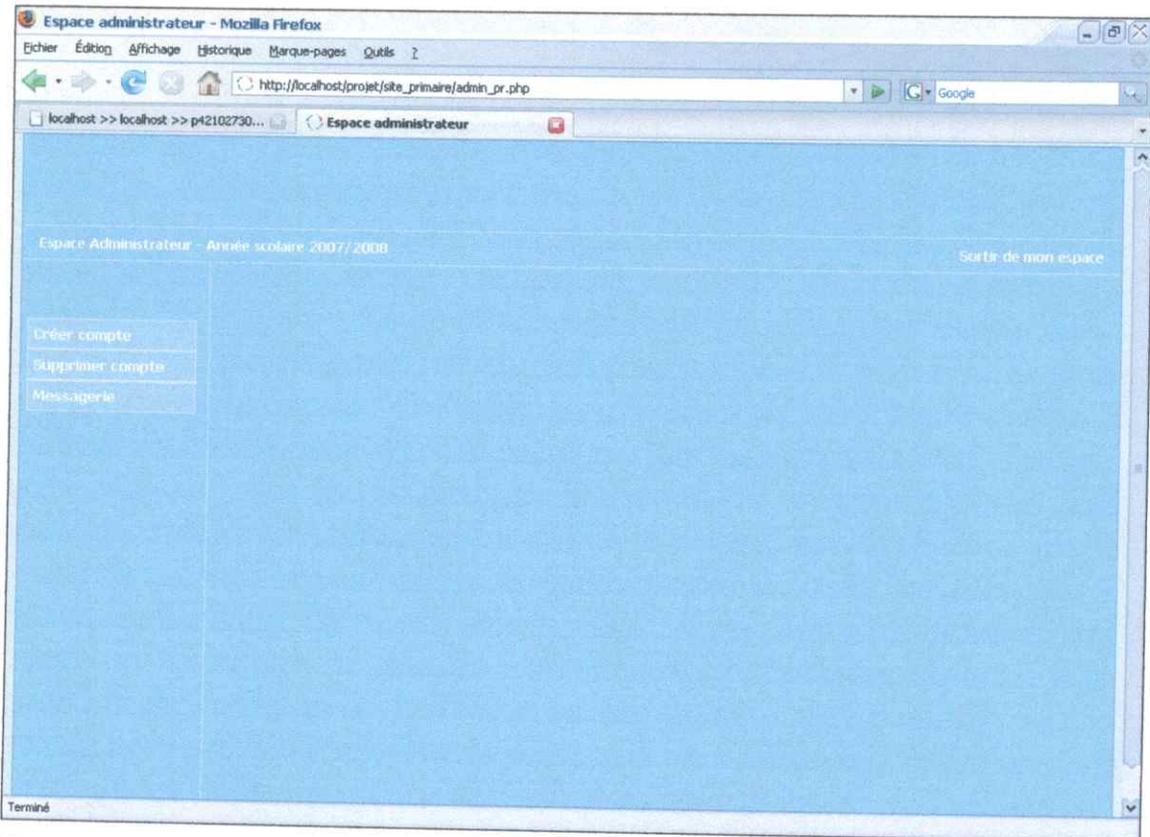
```
<?php
include("ouvrir_bdd.php");
@login = htmlentities($_POST['login']);
@mdp = htmlentities($_POST['mdp']);
@mdp = md5(@mdp);
mysql_query("INSERT INTO `utilisateur` (login,mdp,type_utili)
VALUES('".$_@login."', '".$_@mdp."', 'Direction')");
include("fermer_bdd.php");
?>
```

Exemple 1 :

1 – Authentification correcte de l'administrateur

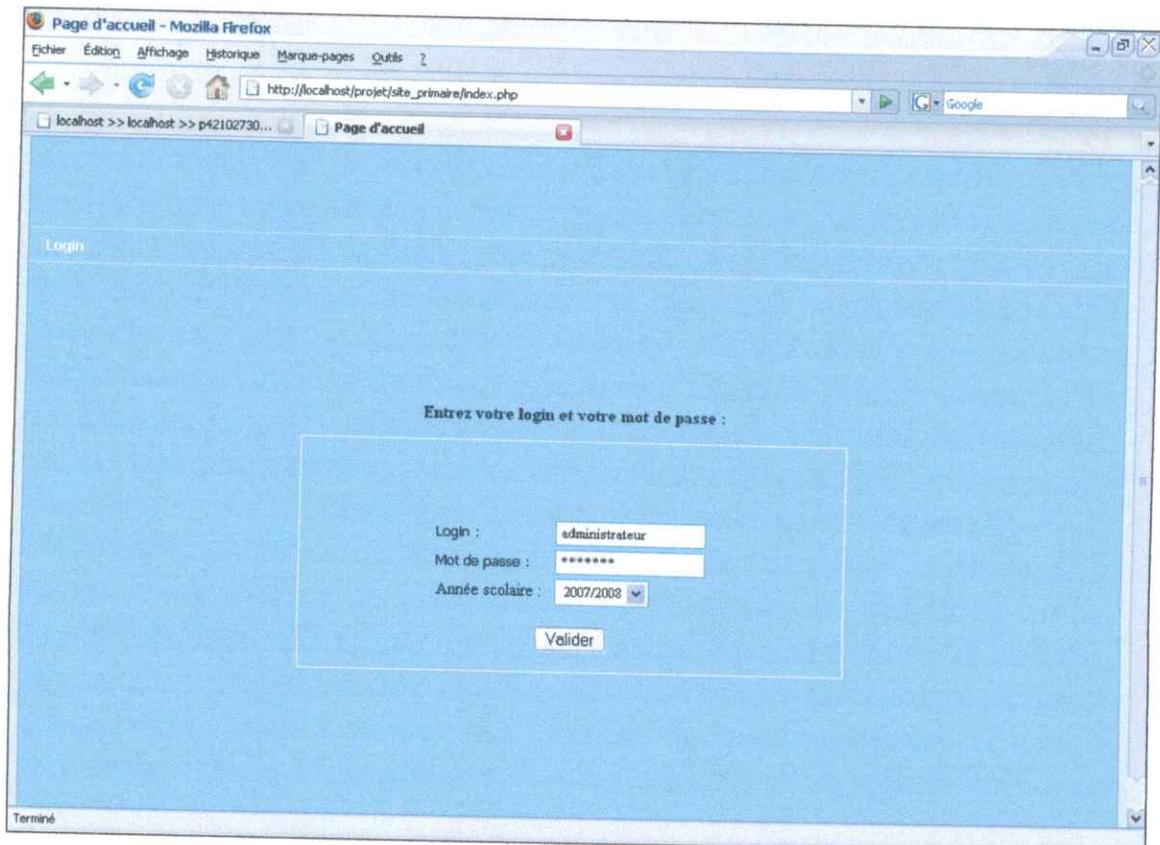


2 – Accès à l'espace administrateur

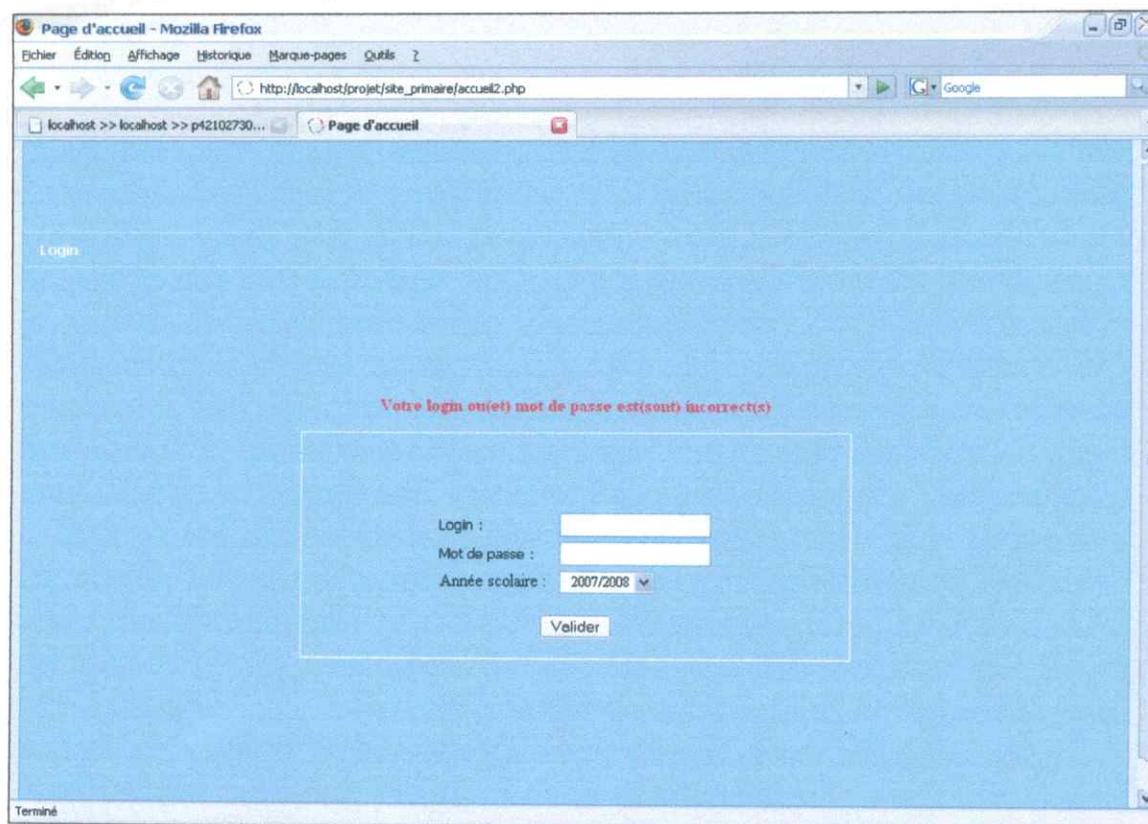


Exemple 2 :

1 – Authentification incorrecte



2 – Affichage du message d'erreur



Code source : Vérification du login et mot de passe

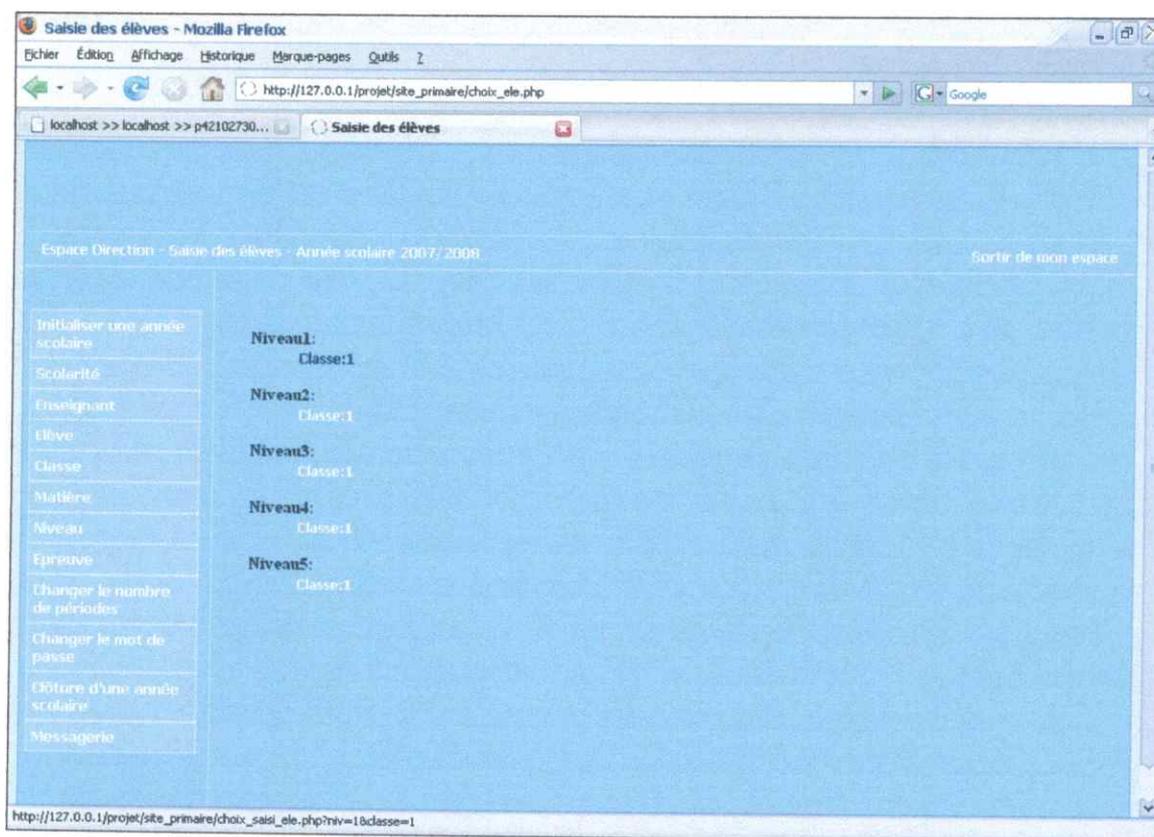
```
$login_entr = $_POST['login'];
$mdp_entr = $_POST['mdp'];
$mdp_entr = md5($mdp_entr);
$rep = mysql_query("SELECT login,mdp,type_utili,utili_id
FROM utilisateur WHERE login='$login_entr' AND mdp='$mdp_entr'");
$don = mysql_fetch_array($rep);
$_SESSION['login']=$don[0];
$_SESSION['mat_ens']=$don[3];
$_SESSION['type_utili'] = $don['type_utili'];
$ligne = mysql_num_rows($rep);
if($ligne == 1) //Authentification correcte
{
    if($don['type_utili']=="Administrateur")
        header("Location: admin_pr.php");
    elseif ($don['type_utili']=="Direction")
        header("Location: dir_pr_clo.php");
    elseif ($don['type_utili']=="Enseignant")
        header("Location: ens_pr_clo.php");
    elseif ($don['type_utili']=="Eleve")
        header("Location: elev_pr_clo.php");
    elseif ($don['type_utili']=="Parent")
        header("Location: parent_pr_clo.php");
}
else //Authentification incorrecte
{
    $_SESSION['erreur'] = 'Votre login ou(et) mot de passe est(sont) incorrect(s)';
    header("Location: accueil2.php");
}
```

5.6.3 Module « Gestion de scolarité »

La direction s'occupe de la saisie, des enseignants, des élèves, des matières, des épreuves, ...

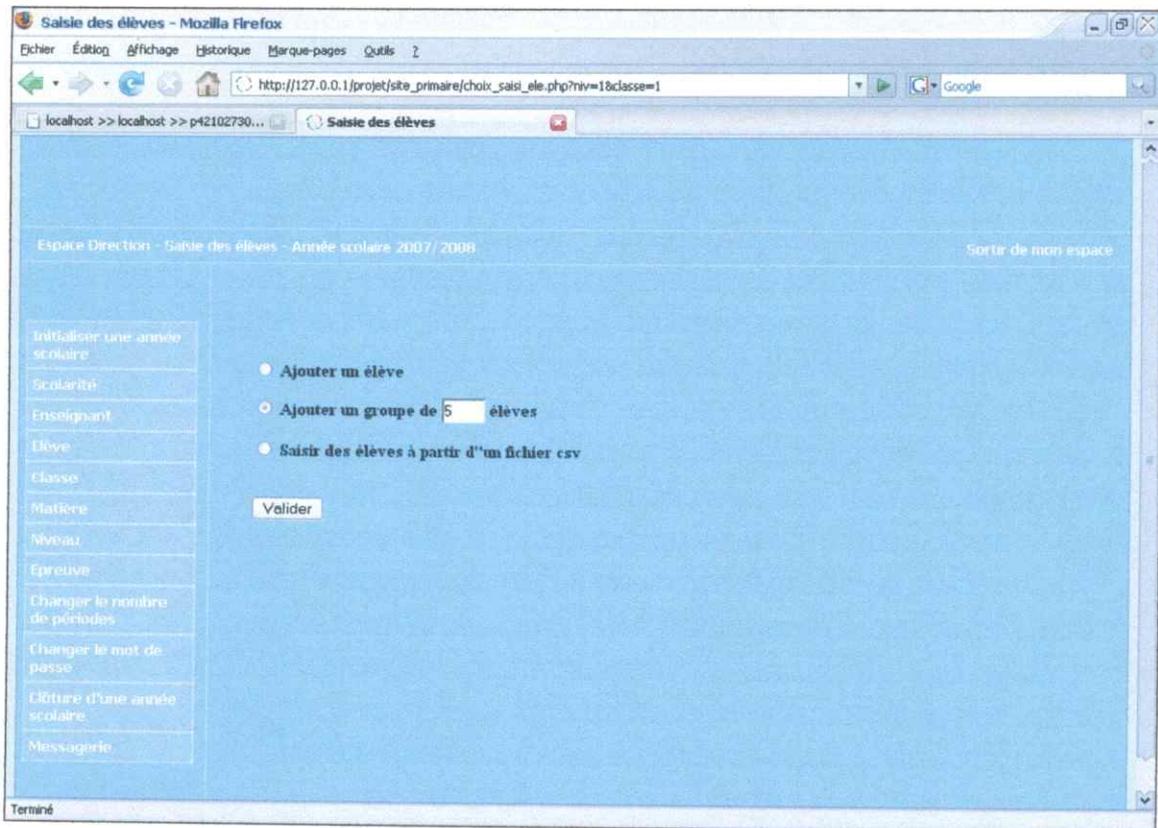
La saisie des élèves :

1 – Choisir le niveau et la classe



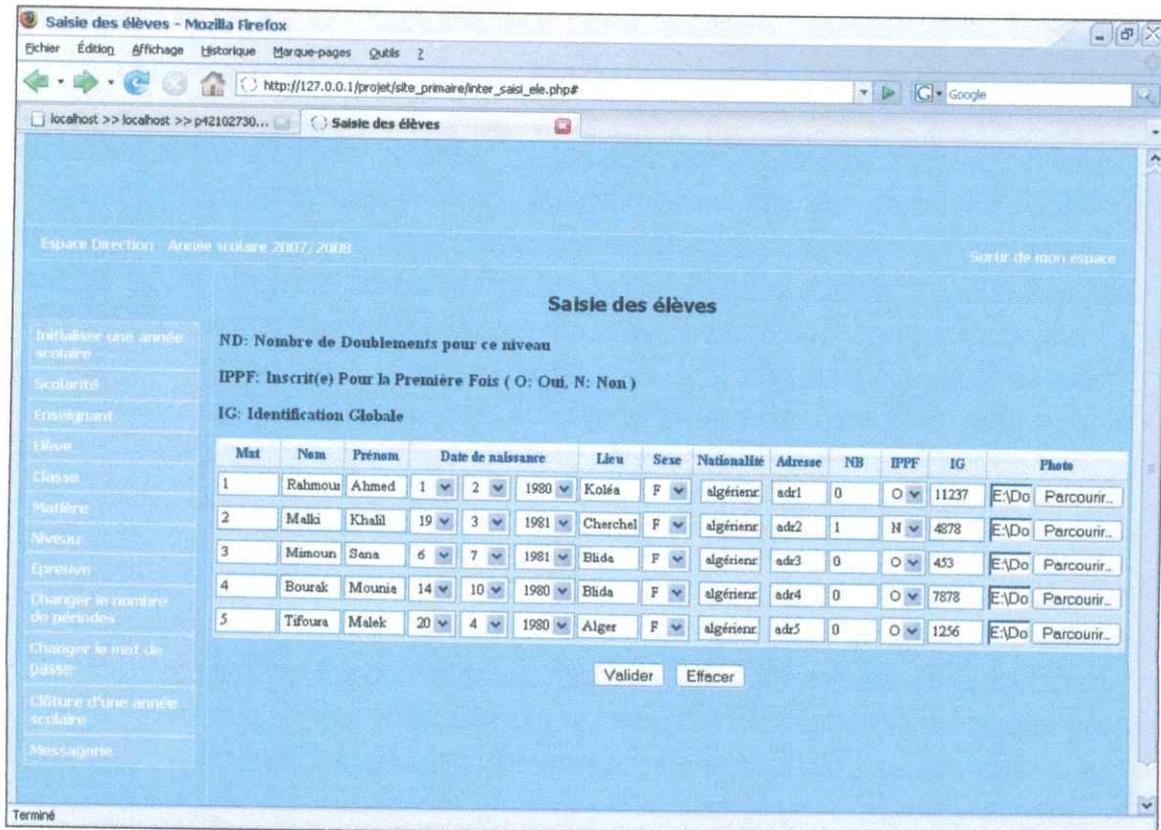
Choix du niveau et de la classe

2 – Choisir d'ajouter un élève, un groupe ou par un fichier csv



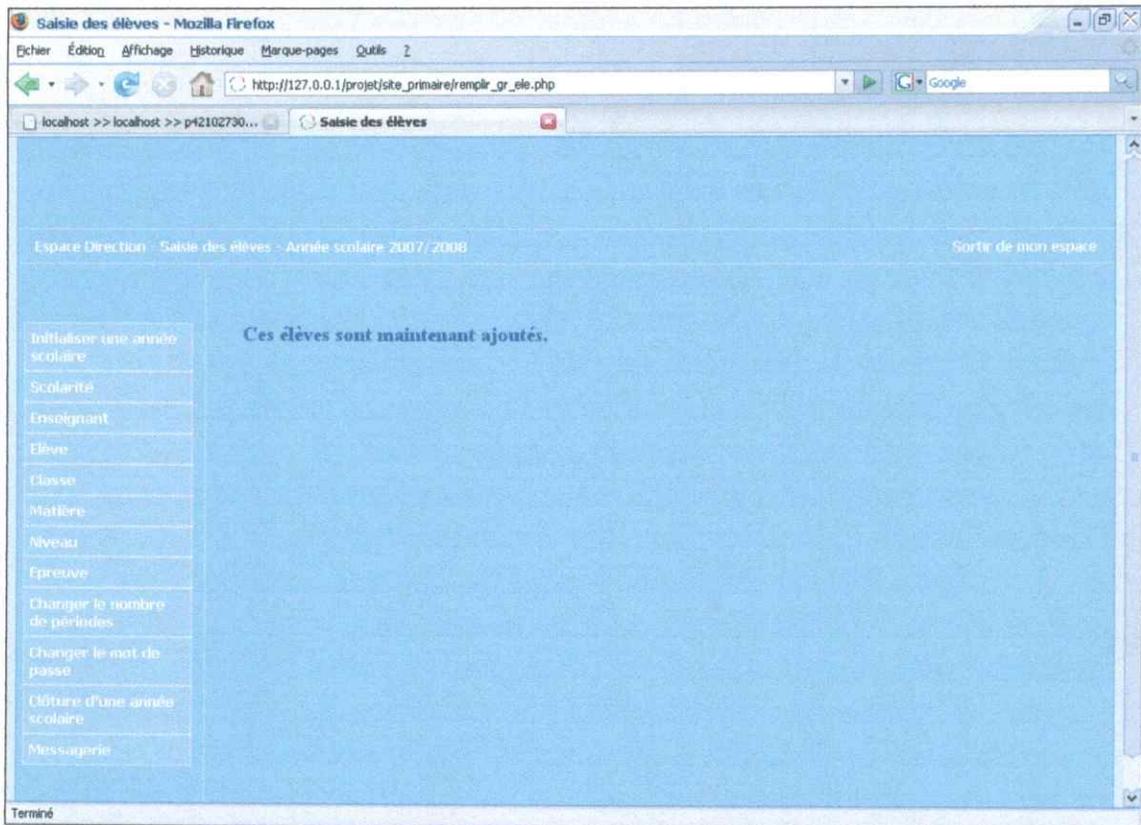
Choix de la manière d'ajouter les élèves

3 – Remplir le tableau avec les informations des élèves



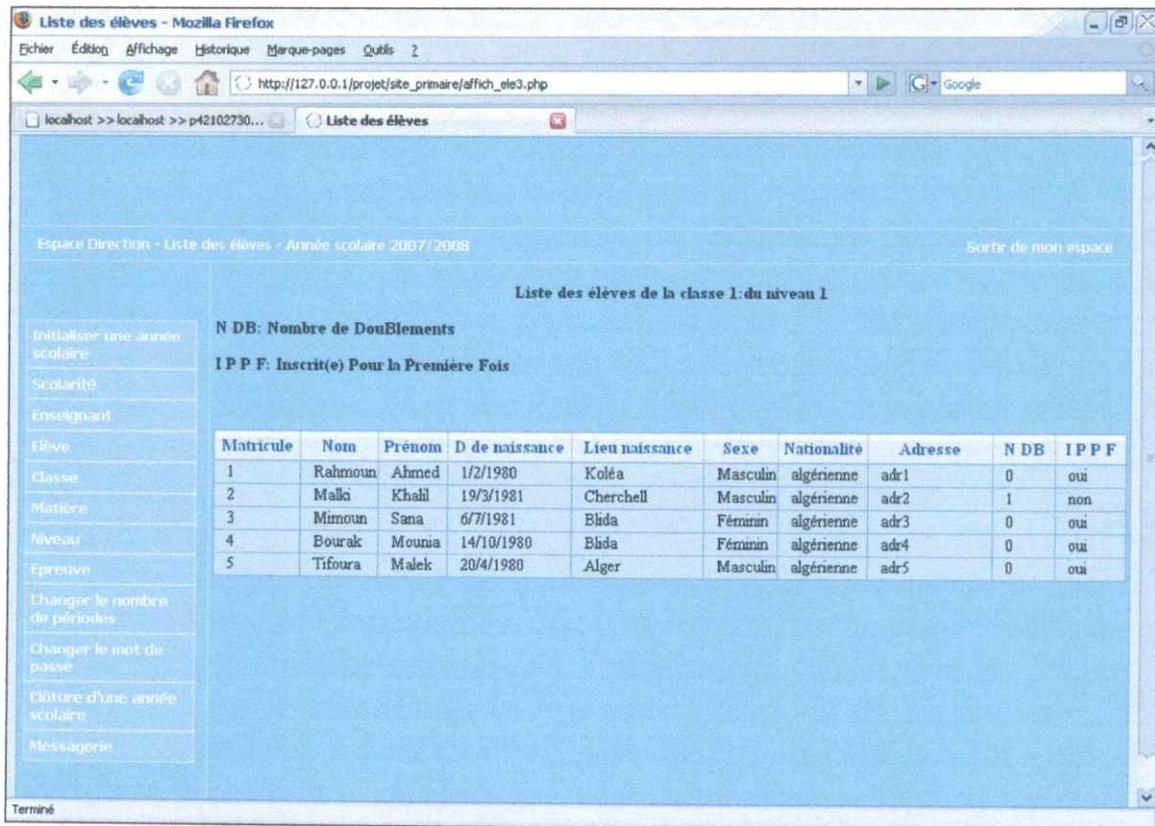
Saisie des élèves

4 – Confirmation de l'ajout des élèves

**Confirmation de l'ajout des élèves****Remarque :**

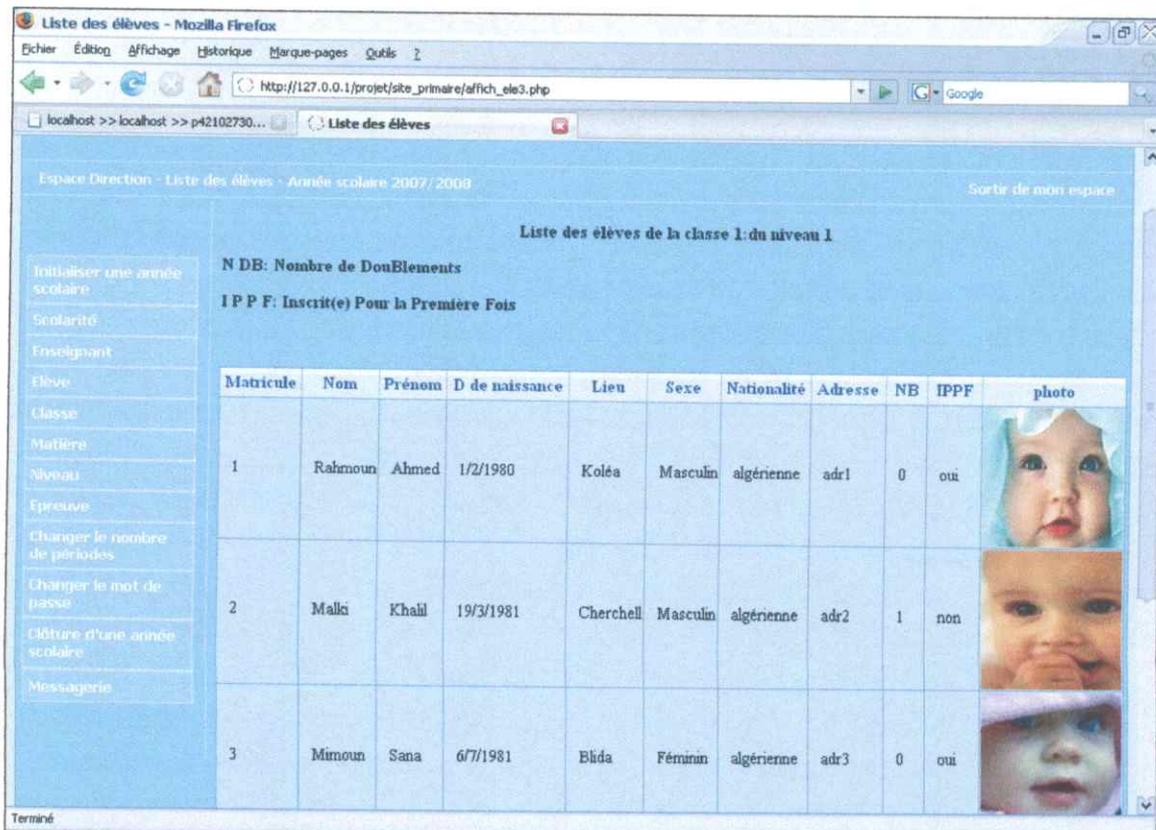
Avant d'ajouter les élèves, on teste si les matricules qui ont été saisis n'existent pas dans la base de données. Si c'est le cas un message de confirmation de l'ajout des élèves est affiché. Sinon, les matricules qui ont été trouvés dans la base de données seront affichés avec le message d'erreur indiquant que les matricules saisis existent déjà.

5 – Affichage des élèves sans photos



Affichage des élèves sans photos

6 – Affichage des élèves avec photos



Affichage des élèves avec photos

Les photos sont redimensionnées à 125x125, choisie comme dimension standard pour toutes les photos. Ainsi, si la photo fournie est plus grande, sa taille sera diminuée, et si elle est plus petite que 125x125, elle sera agrandie pour arriver à la dimension standard que nous avons choisie.

Un dossier sous le nom de « photos » est créé et les photos redimensionnées sont mises dedans. La photo aura pour nom :le matricule de l'élève concaténé au nom de la photo originale, pour éviter le cas de deux images de même nom.

Code source : redimensionnement des photos

```

$img_taille = $_FILES['photo']['size']; //récupérer la taille
$nom_photo = $_FILES['photo']['name']; //récupérer le nom
$ext = strrchr($nom_photo, '.'); //récupérer l'extension
$ext = substr($ext, 1);
//récupérer la photo
$photo = imagecreatefromjpeg($_FILES['photo']['tmp_name']);
$photo_width = imagesx($photo);
$photo_height = imagesy($photo);
$dst_width = 125;
$dst_height = 125;
//créer une image vide de dimension 125x125
$dst = imagecreatetruecolor($dst_width, $dst_height);
//redimensionner la photo
imagecopyresampled($dst, $photo, 0, 0, 0, 0, $dst_width,
    $dst_height, $photo_width, $photo_height);
imagedestroy($photo);
imagegammacorrect($dst, 1.1, 1);
//le chemin de la photo
$nom_img = 'photos/'. $mat. $nom_photo;
mkdir("photos", 777); //créer le dossier s'il n'existe pas
//enregistrer la photo redimensionnée dans le dossier
imagejpeg($dst, $nom_img);

```

Redimensionnement des images

Quand les élèves sont insérés, la photo est redimensionnée et le chemin de la photo est stocké dans la base de données. Nous avons choisi la méthode de stockage du chemin plutôt que de stocker la photo de type blob dans la base de données à cause de la lourdeur de cette dernière.

L'utilisation des fonctions précédentes requiert la bibliothèque GD.

Remarque :

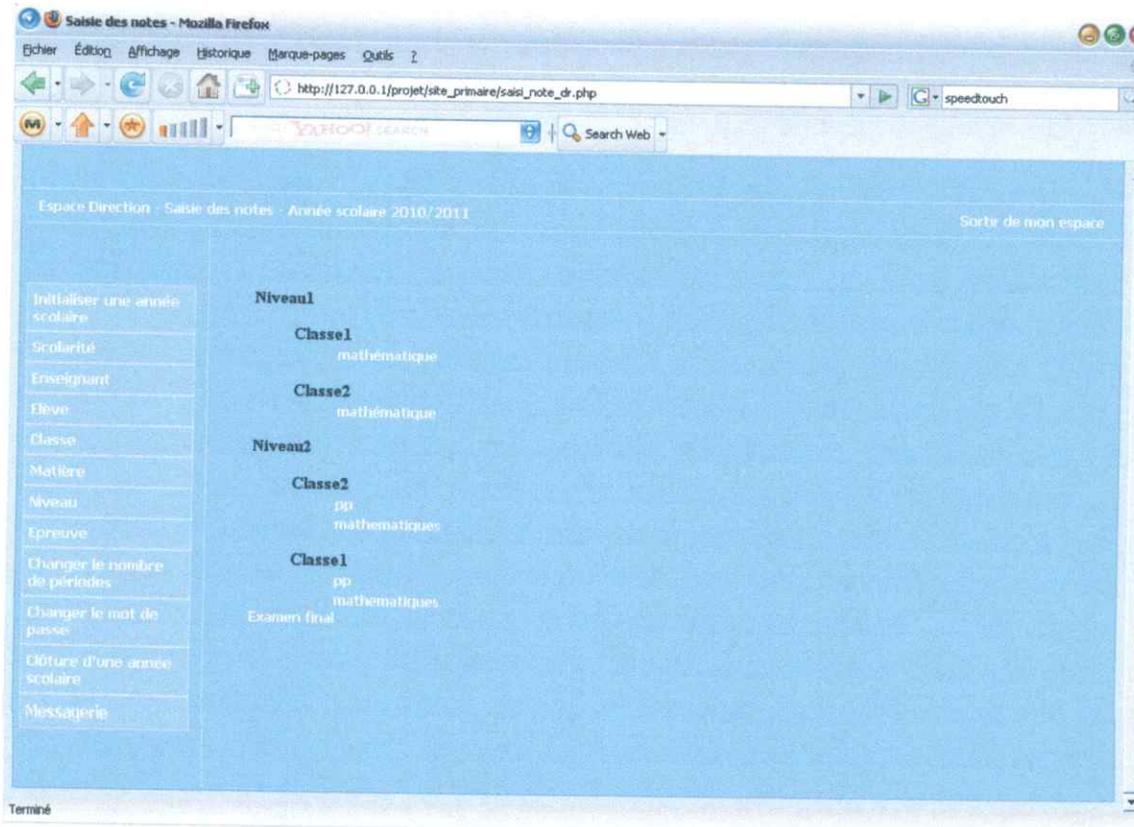
Les photos doivent être de type JPEG, puisque nous avons utilisé des fonctions spécifiques pour ce type.

La saisie des notes :

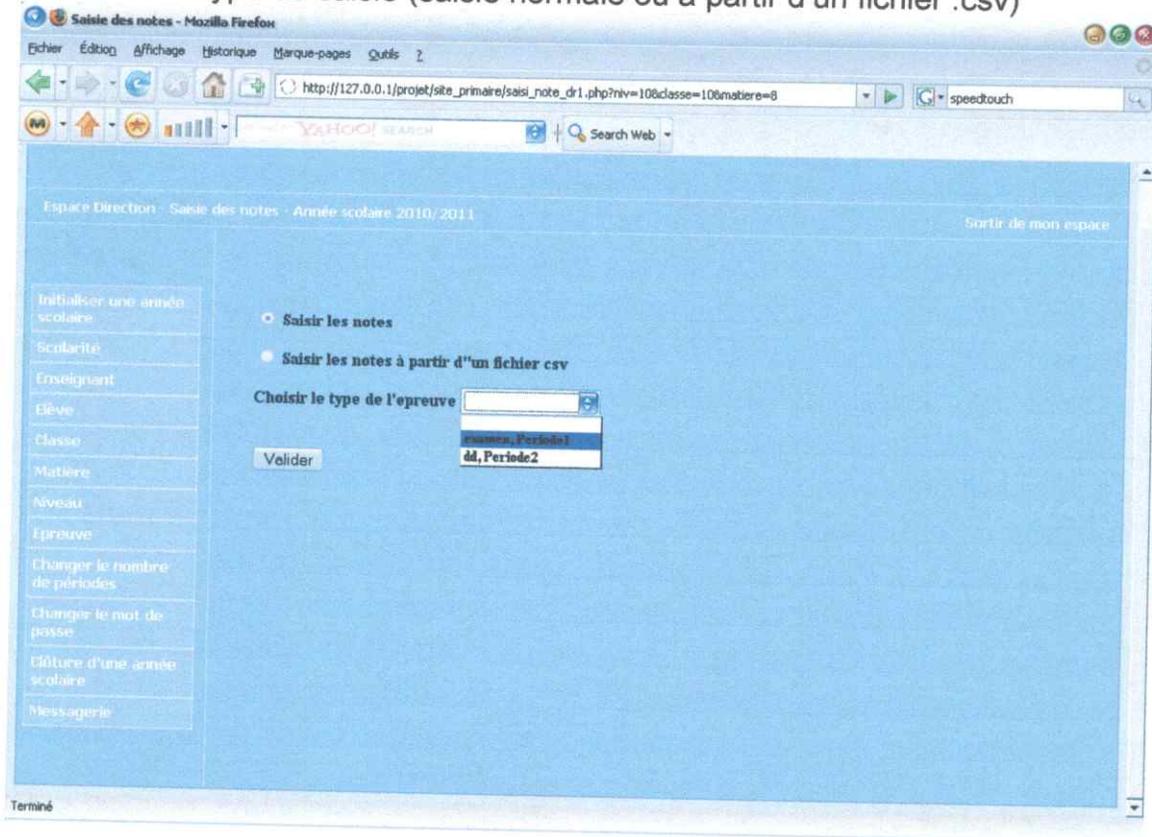
La saisie des notes est une des principales tâches de l'enseignant. La direction affecte à chaque enseignant qu'elle recrute un nombre de matières où il pourra par la suite saisir ses notes. Et puisque pas tous les enseignants saisissent leurs notes dans les délais, on a proposé à la direction de les saisir pour ne pas empêcher le fonctionnement du système comme le calcul des moyennes et l'établissement des relevés des notes.

Exemple de saisie des notes :

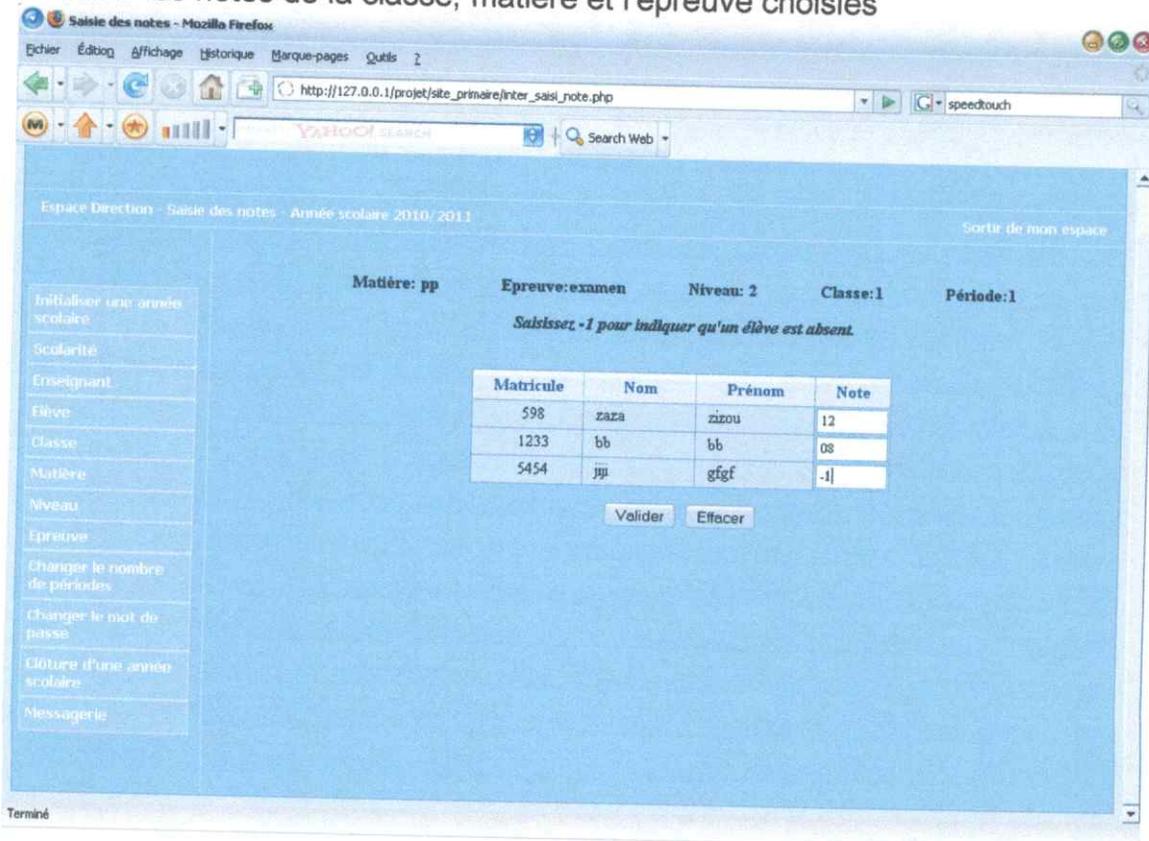
1- Choisir la classe et la matière



2- Choisir le type de saisie (saisie normale ou à partir d'un fichier .csv)



3- Saisir les notes de la classe, matière et l'épreuve choisies



5.6.5 Module « Impression des états »

Il y a 3 documents à imprimer :

- Le certificat de scolarité.
- Le relevé de notes.
- L'attestation de succès.

L'impression se fait à partir d'un fichier PDF généré à la demande d'impression. Pour la génération du fichier PDF, nous avons utilisé la classe FPDF. C'est une classe libre très aboutie qui permet de générer des fichiers PDF, même complexes, en PHP. La principale motivation d'utilisation de cette classe est probablement la liberté d'utilisation et de modification qui l'accompagne. FPDF est aussi très compétitive en termes de performances ainsi qu'en termes de fonctions de haut niveau.

Code source : exemple de génération d'un certificat de scolarité en PDF

```

$pdf=new FPDF('L');
$pdf->Open();
$pdf->SetLeftMargin(40);
$pdf->SetRightMargin(40);
$pdf->AddPage();
$pdf->SetFillColor(232,232,232);
$pdf->SetFont('Arial','B',16);
$pdf->SetY(20);
$pdf->SetX(20);
//L'entête
$pdf->MultiCell(0,20,"République Algérienne Démocratique et Populaire",0,'C');
$pdf->SetFont('Arial','',14);
$pdf->SetY(40);
$pdf->SetX(40);
$pdf->MultiCell(0,10,"Direction de l'éducation de la wilaya de "
.$res[1]."\nEcole Primaire : ".$res[0],0);
$pdf->SetFont('Arial','B',16);
$pdf->SetY(70);
$pdf->SetX(40);
$pdf->MultiCell(0,15,"CERTIFICAT DE SCOLARITE",0,'C');
$pdf->SetFont('Arial','',14);
$pdf->SetY(90);
$pdf->SetX(40);
$pdf->MultiCell(0,10,"Le Directeur de l'école ".$res[0].",
certifie que l'élève ".$resl[1]." ".$resl[2].",
né(e) le ".$resl[3]."/".$resl[4]."/".$resl[5]." à ".$resl[16].",
de nationalité ".noaccent($resl[17]).", est inscrit(e) en ".$niv.",
sous le matricule ".$resl[0].", pour l'année scolaire ".$res[2].",0);
//génération du pdf
$pdf->Output("Certif_scol.pdf","I");

```

Génération d'un certificat de scolarité en PDF

5.6.6 Module « Communication »

La messagerie

La messagerie permet à tous les utilisateurs de communiquer entre eux. Tout utilisateur possède une boîte d'envoi et une boîte de réception de messages. Pour cela, nous avons créé une table « messagerie » contenant les champs suivants :

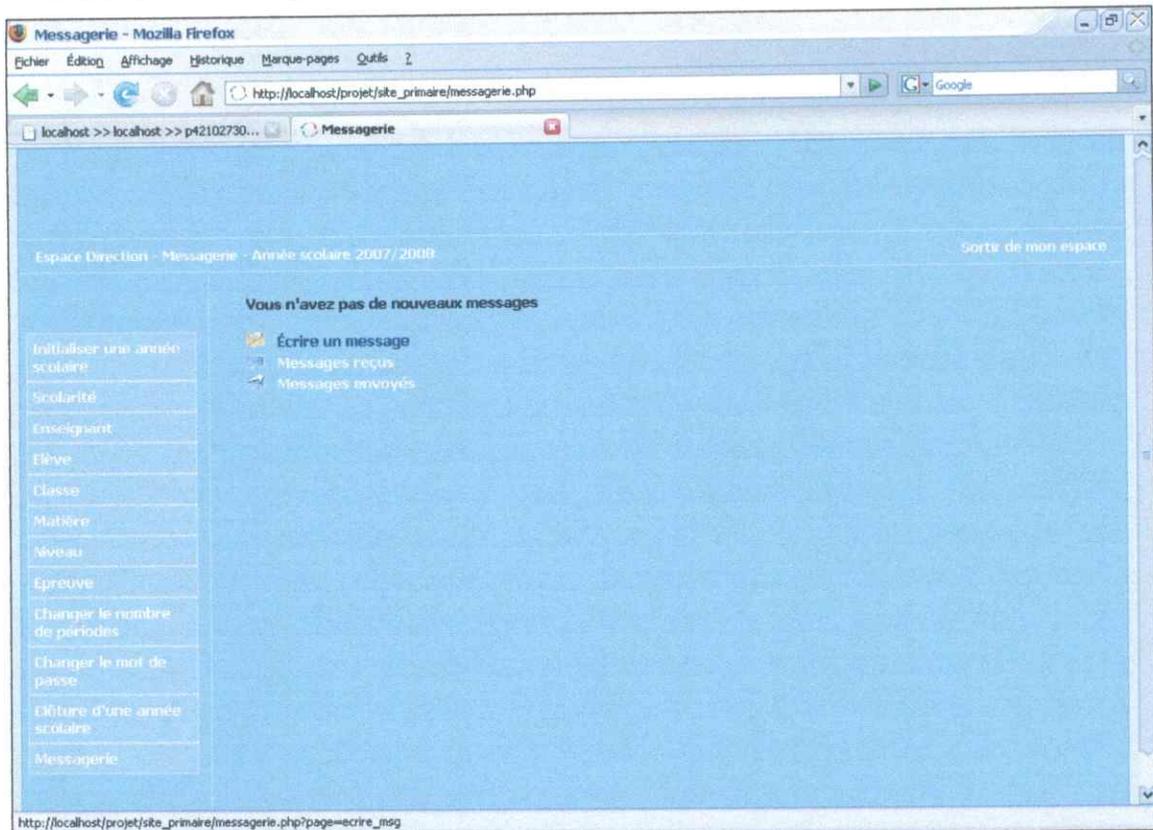
La table « messagerie »

Champ	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/> <u>id_mess</u>	bigint(20)			Non		auto_increment
<input type="checkbox"/> date	date			Non	0000-00-00	
<input type="checkbox"/> heure	time			Non	00:00:00	
<input type="checkbox"/> sujet	text	latin1_swedish_ci		Non		
<input type="checkbox"/> texte	text	latin1_swedish_ci		Non		
<input type="checkbox"/> id_exp	bigint(20)			Non	0	
<input type="checkbox"/> id_dest	bigint(20)			Non	0	
<input type="checkbox"/> lu	int(2)			Non	0	
<input type="checkbox"/> type	char(3)	latin1_swedish_ci		Oui	NULL	

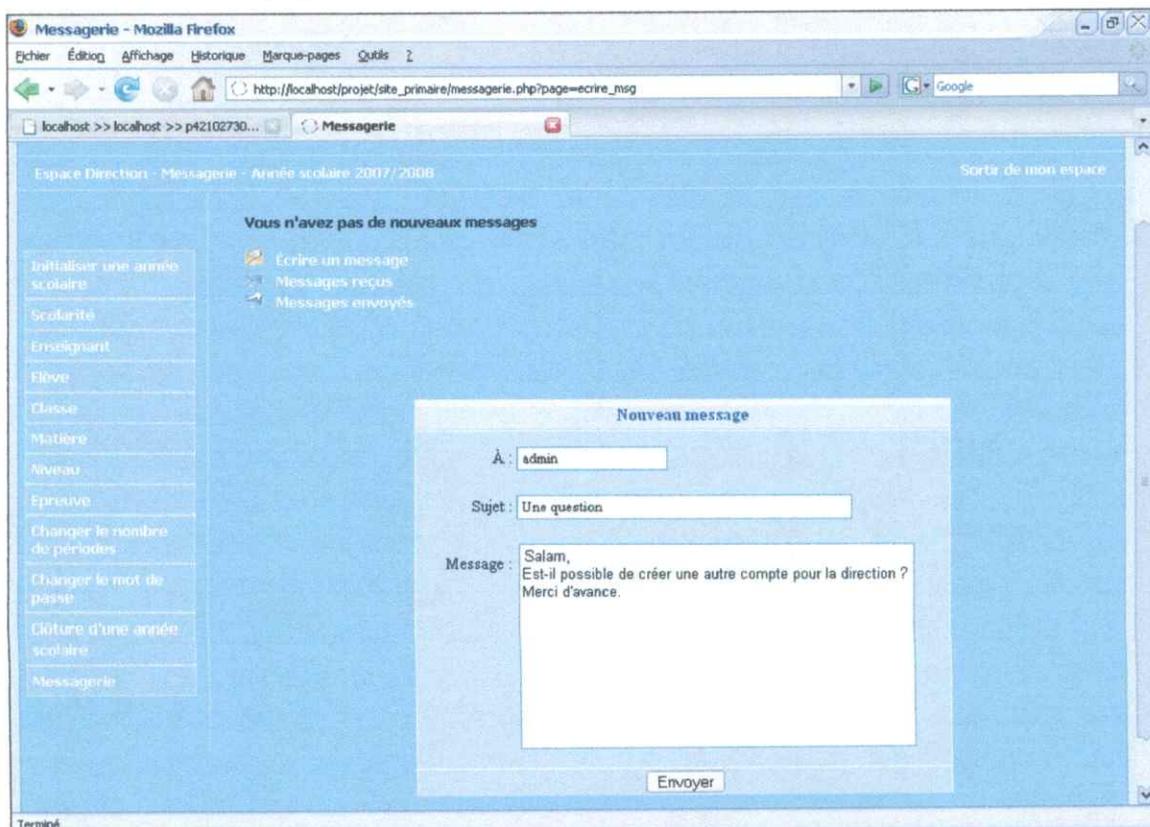
Table messagerie

Exemple : Envoi de message de la direction à l'administrateur

1 – Aller sur la messagerie dans l'espace direction et cliquer sur « Ecrire un message »

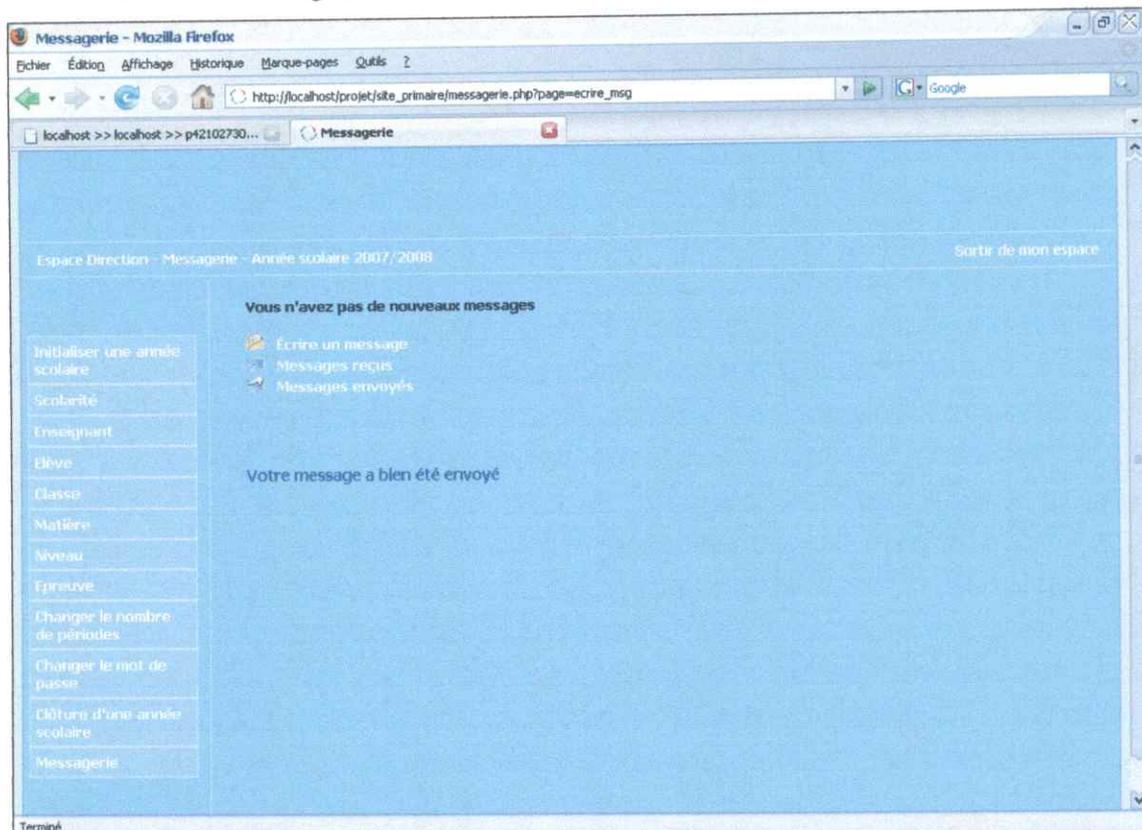


2 – Ecrire le message puis cliquer sur « envoyer ». Il faut donner un login valide.

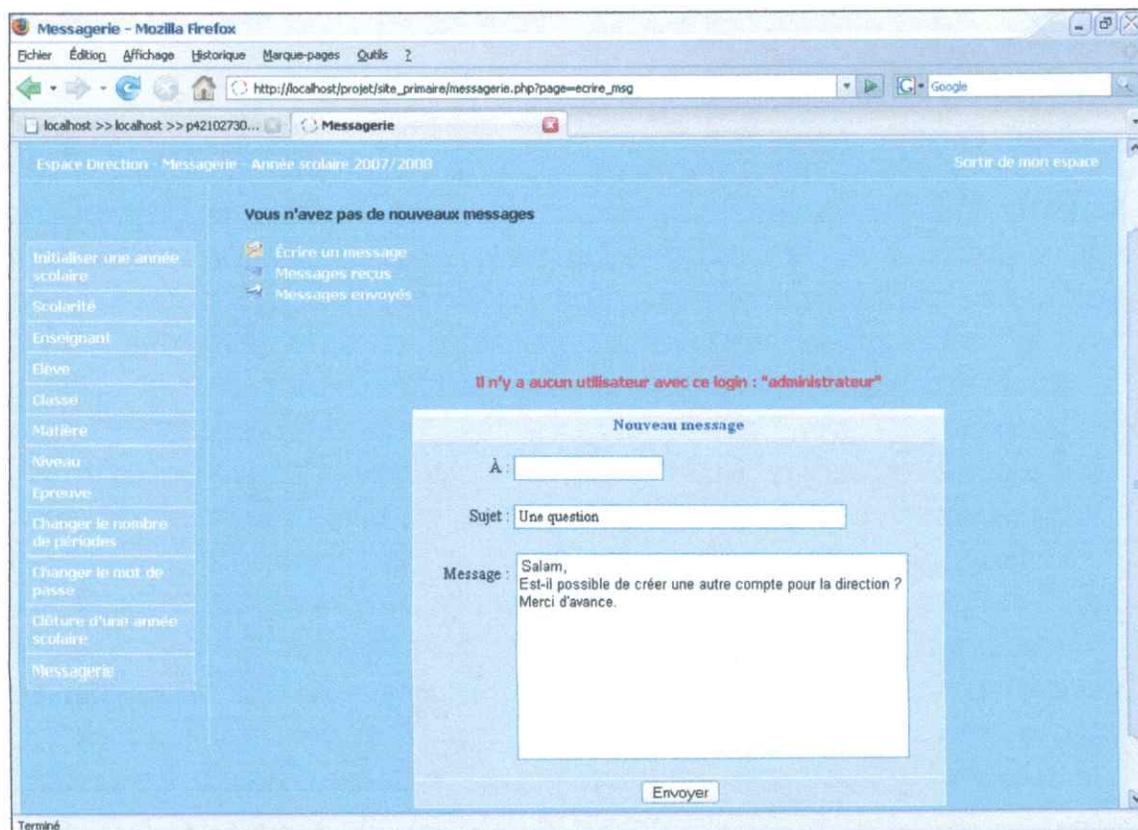


3 – La confirmation de l'envoi du message

- Le cas où le login existe :

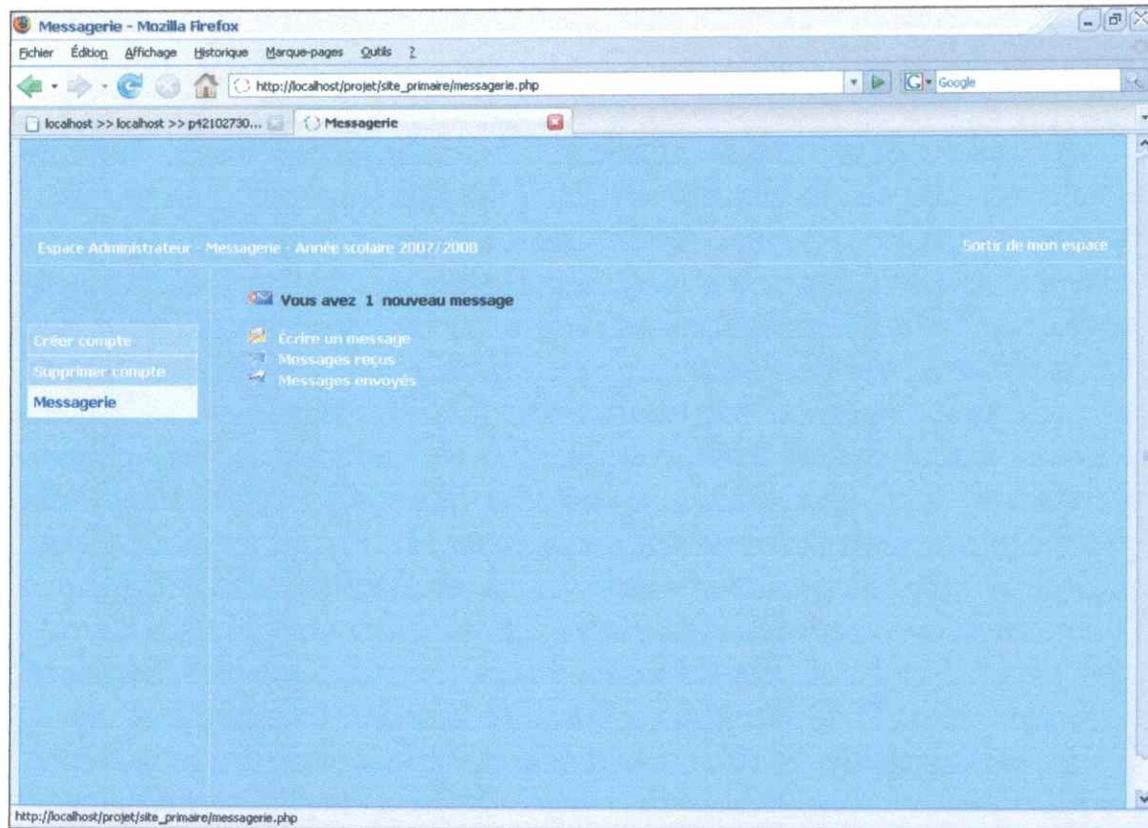


- Le cas où le login n'existe pas

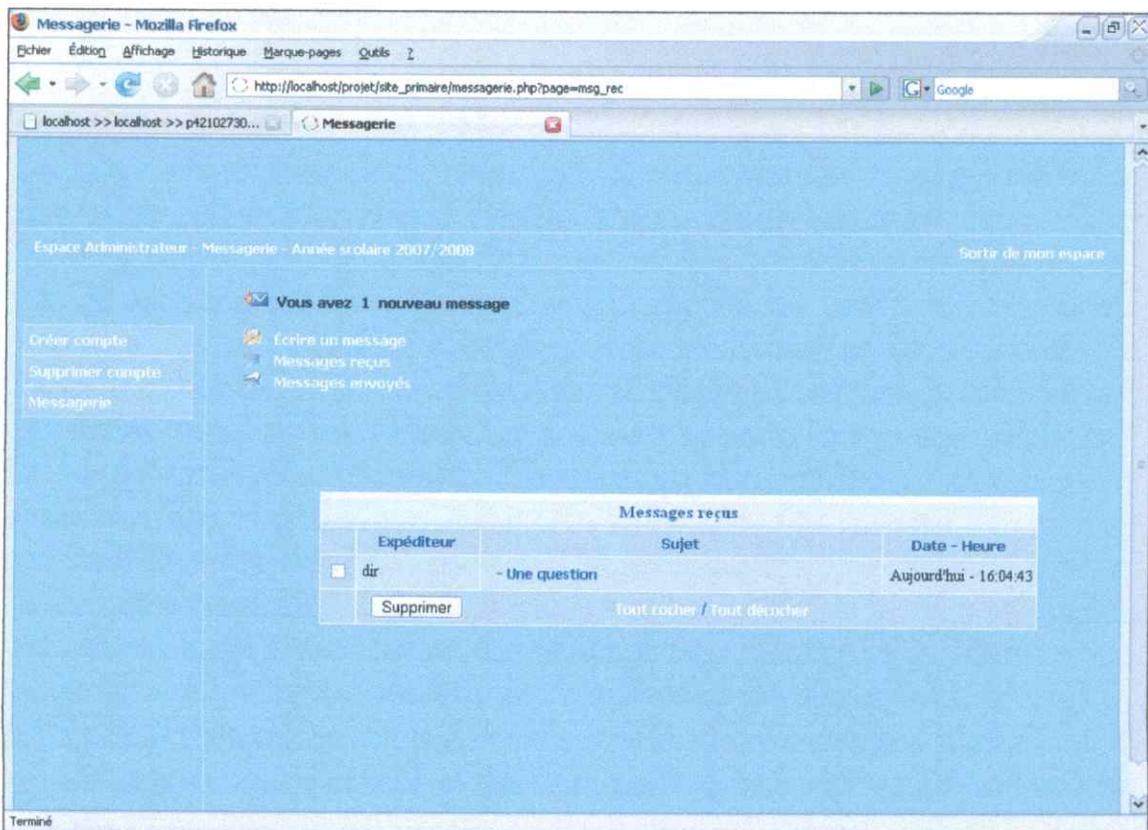


Remarque : Le login de l'administrateur est « admin ».

4 – La réception du message dans l'espace administrateur

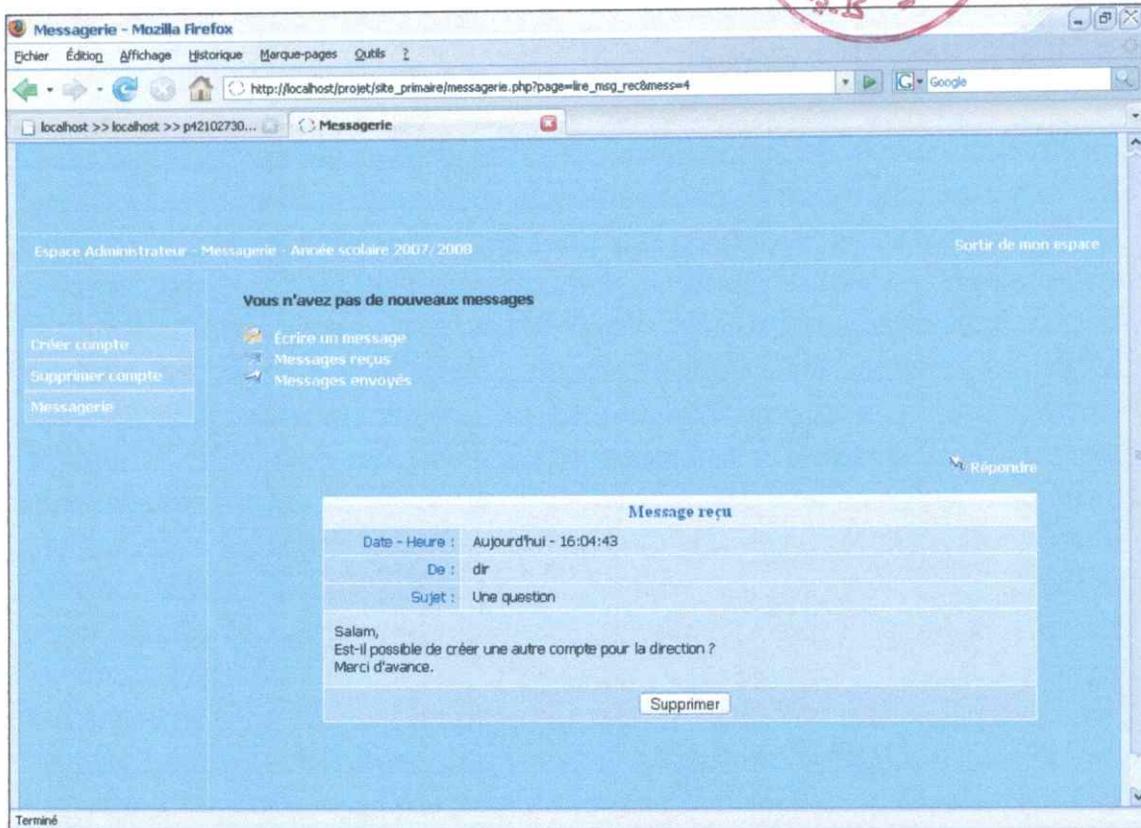


5 – Cliquer sur le message qui est en gras pour le lire

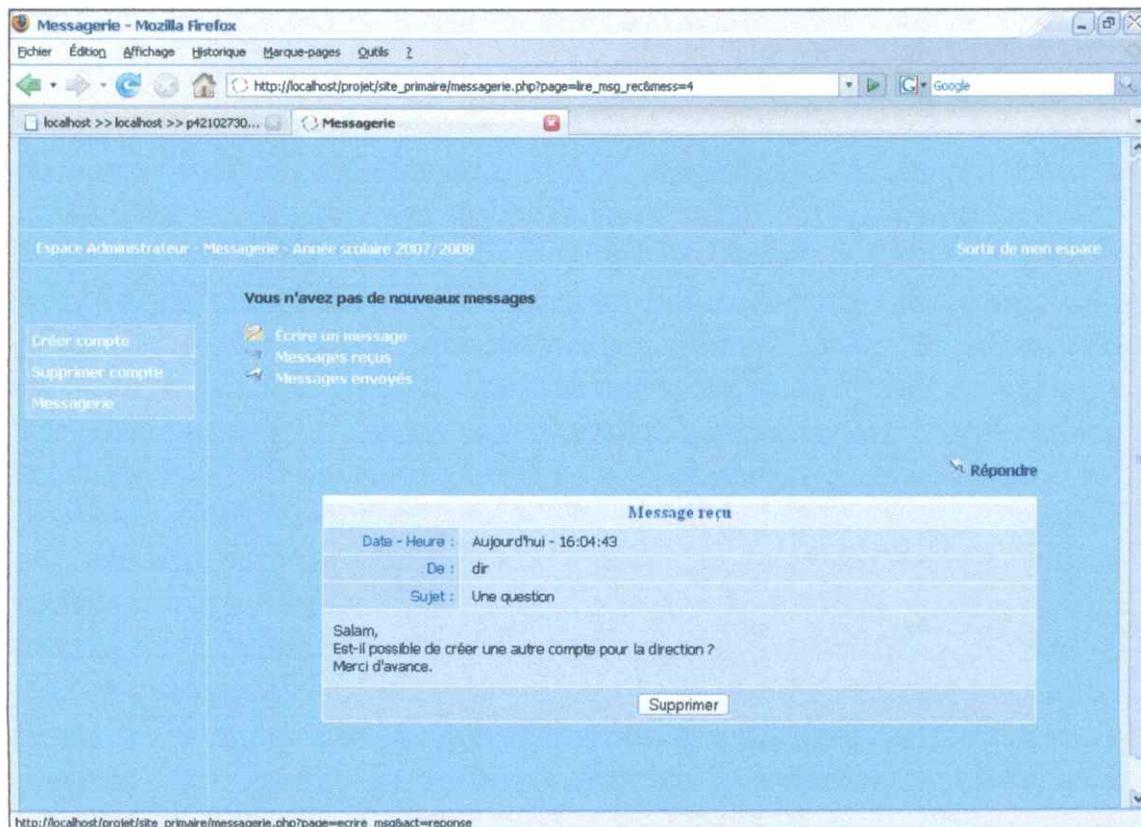




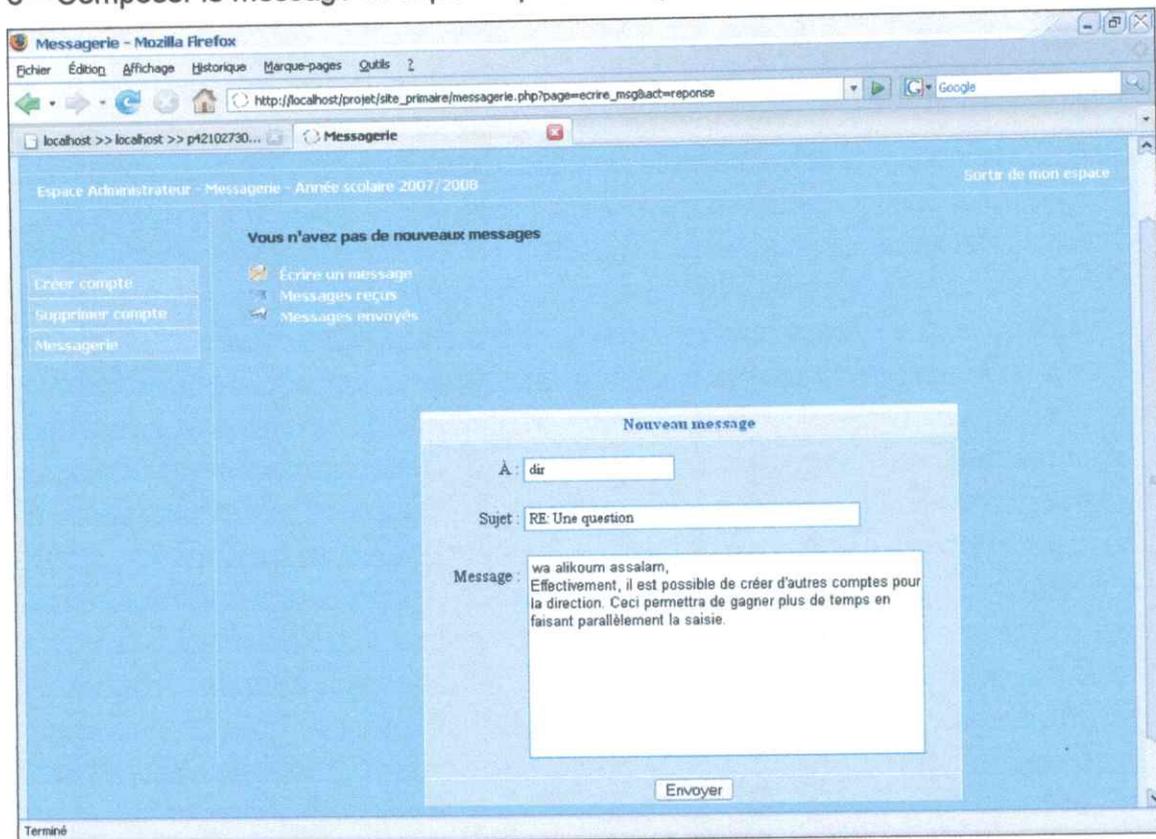
6 – Lecture du message



7 – Cliquer sur répondre pour écrire une réponse



8 – Composer le message de réponse puis l'envoyer



Conclusion et perspective

Une telle application pourra réellement améliorer, et d'une manière efficace, les établissements scolaires (primaire, moyen, lycée). Avec leurs composantes : direction, administration, enseignants, élèves et parents d'élèves, ils pourront profiter de certaines fonctionnalités, comme la saisie des notes, l'impression des états qui sont le certificat de scolarité et les relevés des notes, utiliser la messagerie ce qui permet à tous les utilisateurs inscrits dans le système de communiquer, ainsi que la consultation des notes par les élèves et par leurs parents.

L'objectif de notre projet était de commencer à partir d'une scolarité toute prête, puis l'étendre avec certaines caractéristiques comme le parcours pédagogique et la distribution, mais vue la grande difficulté pour comprendre un travail déjà fait, surtout avec un code source mal commenté, et vue la contrainte du temps, nous avons opté pour la réalisation d'une scolarité complète. Mais le sujet de la scolarité s'est avéré être un sujet assez vaste et compliqué et a pris plus de temps que prévu pour sa réalisation, ce qui nous a empêché d'atteindre tous nos objectifs.

Nous les citons à présent comme des perspectives :

- La détermination du parcours pédagogique complet d'un étudiant, du primaire à l'université.
- La distribution : celle-ci doit permettre de faire communiquer, via internet, un nombre important de systèmes installés.

Pour terminer, nous pourrions affirmer qu'un tel système de gestion de la scolarité, rendra meilleures, les pratiques administratives dans les établissements scolaires et ouvrira de nouveaux horizons et de nouvelles perspectives dont toute la communauté scolaire pourra bénéficier. Il est d'ailleurs de notre devoir et notre responsabilité de trouver des moyens afin d'améliorer le quotidien de nos concitoyens, spécialement dans le domaine scolaire qui constitue la pièce maîtresse sur laquelle repose l'avenir de notre pays.

Bibliographie

- [1] Pascal Roques, et Franck Vallée, *UML 2 en action – De l'analyse des besoins à la conception*, Eyrolles, 4^{ème} édition.
- [2] Pascal Roques, *UML2 – Modéliser une application web*, Eyrolles, 3^{ème} édition.
- [3] Paul-Hubert des Mesnards, *Réussir l'analyse des besoins*, Eyrolles.
- [4] Hugues MARCHAT, *La gestion de projet par étapes – L'analyse des besoins*, Eyrolles.

Webographie

- [1] <http://www.developpez.com/>
- [2] <http://www.eixa6.fr/technos/etudesUML2besoins.htm>
- [3] <http://fdigallo.online.fr/cours/uml.pdf>
- [4] http://www-adele.imag.fr/users/Jean-Marie.Favre/ENSEIGNEMENT/CCIGL-00-01/IL/AnalyseDesBesoinsEtIntroductionAUML-1_fichiers/v3_document.htm
- [5] http://www-adele.imag.fr/users/Jean-Marie.Favre/ENSEIGNEMENT/CCIGL-00-01/IL/AnalyseDesBesoinsEtIntroductionAUML-1_fichiers/v3_document.htm
- [6] <http://www.reseaucerta.org/didactique/uc-intro/>
- [7] http://www.univ-angers.fr/docs/etudquassi/Methodes_agiles.pdf
- [8] <http://medina.ftp-developpez.com/presentationxp.pdf>