



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



CENTRE UNIVERSITAIRE DE MILA
INSTITUT DES SCIENCES ET DE LA TECHNOLOGIE

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

Licence Académique

Domaine : **Mathématiques et Informatique**
Filière : **Informatique**
Spécialité : **Informatique**

Thème

*Conception et développement d'une
application Web pour l'inscription en ligne des
étudiants au niveau du Centre Universitaire de
Mila*

Présenté par :

- Bara Sihem
- Bensiammar karima

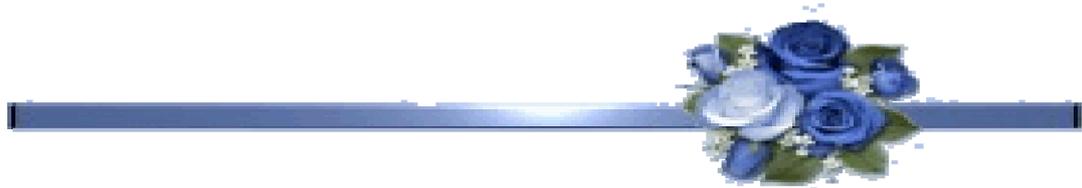
Dirigé par :

Mr. Dib Abderrahim

Année Universitaire 2010-2011



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



Remerciement

Nous tenons à remercier toujours et par cette occasion, en premier et avant tout, notre créateur <<ALLAH>> car sans sa majesté et son aide nous n'aurions pu arriver à se stade scientifique.

Nous présentons nos sincères gratitudes et remerciements à notre encadreur MR DIB ABDRAHIM pour le grande soutien morale qu'il nous a apportée au cours de notre projet et aussi pour son aide précieuse et ses conseils judicieux qu'il nous a fait bénéficier

Un grand remerciement se dirige également vers BARA MOHAMED qui nous a permit de donner des conditions nécessaire pour la réalisation de ce projet.

Nous tenons à exprimer nos sincères remerciements à tout le personnel de l'institut de l'informatique Ainsi que tous les enseignants qui nous ont enseigné durant toutes nos années d'étude.

Enfin, Nous exprimons nos plus vifs remerciements à toute personne qui nous a aidé à élaborer ce travail de proche ou de loin.

Sihem.B et Karima.B

إهداء إهداء

الحمد لله الذي جعلنا من خير أمة أخرجت للناس ، وألبسنا لباس التقوى خير لباس ، أحمده وأشكره ، وأتوب إليه وأستغفره ، رفع شأن العلم والعلماء ، وقرنهم به وبملائكته بالوحي من السماء ، وصلى الله على الرسول الكريم ، حث على العلم ورغب فيه ، وفرق بين سائر الخلق وبين سالكيه .
من أين أبدأ والمشاعر ليس تبلغ ما تريد في مثل هذه المواقف يعجز عن الكلام اللسان، ويضعف بين الضلوع الجنان، وربما وأنتم لا تشعرون تسيل دموعها العينان

أمي الحبيبة استميتك عذرا في أن اقبل يدك الطاهرتين قبل كل شيء وقبل ما اشرع في كتابة سطوري إلى ينبوع الحنان إلى من عندها أجد الراحة والأمان إلى شمسي وقمري، إلى أغلى كواكب الدنيا، أمي الغالية "دليلية".
إلى من تعذب من أجلي وشقي لمرضي إلى من عمل لراحتي وسهر علي ، يامن واجهت المتاعب من أجل إرضائي ، أنت زرعت الحب في وعلمتني حسن الخلق لك خالص حبي و واحترامي ودعواتي،أبي العزيز"مولود".

إلى إخوتي الأعمام : " سامي" وشكر خاص له ، " حمزة " ، " أمير " ، " هدى " و خطيبها " مراد " ،
والحبوبة " زينب"
كما لا أنسى جدتي و أخوالي و أعمامي وكل بناتهم و أولادهم خصوصا "صالح" ، "محمد" .

يحتار حرفي..ولا يدري كيف يسطر لك كلمات الشكر التي تقي بحقك..وتعبر عن مدى امتناني ووقوفك معي
وبجانبي..صدقيني لن أنسى وكيف أنسى وفاء قلب عرف معنى الوفاء فنسج حروفه في حياتي تحياتي لك ياأحلى صديقة
"ليندة"

إلى كل الصديقات العزيزات:كريمة،إلهام،نعيمة،أسمهان،عائدة،صفاء،مريم،ليلي،سمية،ياسمينية،حميدة،ونام،شافية، و داد، و
إلى كل من عرفتها في حياتي.

إهداء خاص إلى طلبة الإعلام آلي بدون استثناء و أقول لهم عند الرحيل تعزف الرياح لحنها الحزين وبشتد العناق وتذبل الجفون لكن الذي سيبتعد عنكم هو الجسد و ليس القلب.

سهام سهام



الإهداء الإهداء

بسم الله الرحمن الرحيم

(قل إعملوا فسيرى الله عملكم ورسوله والمؤمنون)

صدق الله العظيم.

إلهي لا يطيب الليل إلا بشكرك ولا يطيب النهار إلا بطاعتك .. ولا تطيب اللحظات إلا بذكرك .. ولا تطيب الآخرة إلا بعفوك .. ولا تطيب الجنة إلا بروحيتك ورحمتك .

إلى من بلغ الرسالة وأدى الأمانة .. ونصح الأمة .. إلى نبي الرحمة ونور العالمين سيدنا محمد عليه الصلاة وأزكى التسليم.

إلى من كلله الله بالصيبة والوقار .. إلى من علمني العطاء بدون انتظار .. إلى من أحمل اسمه بكل افتخار أرجو من الله أن يمد في عمرك لتري ثماراً قد حان قطافها بعد طول انتظار وستبقى كلماتك نجوم أهدى بها اليوم وفي الغد والبالأبد والدي العزيز "عبد المعبود".

إلى ملائكتي في الحياة .. إلى معنى الحب وإلى معنى الحنان والتفاني .. إلى بسمه الحياة وسر الوجود إلى من كان دماغها سر نجاحي إلى أمي الحبيبة "حنينة".

إلى روح سكنه روعي إلى ملائكتي في الحياة ولبس جروحي، زوجي وقرّة عيني "نافع".

إلى من أرى التفاؤل بعينه .. والسعادة في ضحكته أخي الحبيب "بوعلاء".

إلى من كن شمعاته ينرن دربي : أخواتي "سامية"، "سارة"، "دنيا"، "دلال"، "طليحة"

و المكتوتة "كوثر" وعمتي "لويذة" و "موريتة".

إلى أبي بعد أبي جدي "عبد الرحمن" و جدتي الحبيبة "عطري".

إلى الأخوات اللواتي لم تلدمن أمي .. إلى من تلو بالإخاء وتميزوا بالوفاء والعطاء إلى ينابيع الصدق الصافي إلى من معهم سعدت ، وبرفتهم في دروب الحياة الحلوة والحزينة سرى.

إلى من عرفته كجنت أجدهم وعلموني أن لا أضيعهم صدقاتي كل واحدة باسمها.



كريمة
كريمة

SOMMAIRE

Introduction générale	1
Chapitre 01 : Le Web	3
1. introduction	3
2. Principe et Architecture du modèle Client / Serveur	3
2.1 Principe du modèle Client /Serveur	4
2.2 Architecture des systèmes Client / Serveur	5
3. Caractéristiques client/serveur	5
4. Les avantages et les inconvénients	6
5. Le Web	6
5.2 Serveur Web	6
5.2 Client Web	6
5.3 site web	7
6. Programmation Web	7
6.1 Le Web statique	7
6.2 Web dynamique	7
6.3 Le Common Gateway Interface CGI	8
7. Web et bases de données	8
7.1 Accès aux bases de données	8
7.2 Avantages cumulés du Web et des bases de données	8
8. Sécurité sur le Web	9
8.1 Objectifs de la sécurité Web	9
9. conclusion	10
Chapitre 02 : Langage UML et la démarche de conception	
1. introduction	12
2. Définition et historique d'UML	12
3. Caractéristiques d'UML	12
4. Les Axes de la Modélisation	13
5. Les diagrammes UML	13
5.1) Le diagramme de cas d'utilisation	13
5.2) Le diagramme de classe	14
5.3) Le diagramme d'objet	14
5.4) Le diagramme état transition	14
5.5) Le diagramme d'activités	15
5.6) Le diagramme de la collaboration	15
5.7) Le diagramme de séquence	15

5.8) Le diagramme de composants	16
5.9) Le diagramme de déploiement	16
6. La démarche de conception de l'application	16
6.1 Introduction	16
6.2 La démarche de conception	16
a. Première étape : Identification des besoins	16
b. Deuxième étape : Phase d'analyse	17
c. Troisième étape : Phase de conception	17
7. Conclusion	18

Chapitre 03 : La conception

1. Cahier de charge	20
a) Présentation du projet	20
b) Les besoins fonctionnels	20
c) Les besoins techniques	20
d) Identification des acteurs	20
2. Diagramme de contexte	21
3. Le diagramme de cas d'utilisation	21
4. La description des cas d'utilisation	23
4.1 Administrateur	23
4.1.1 s'authentifier	23
4.1.2 Créer compte pédagogie	23
4.2 agent pédagogie	24
4.2.1 Rechercher	24
4.2.2 Supprimer compte étudiant	24
4.3 étudiant	25
4.3.1. Créer compte étudiant	25
4.3.2. S'inscrire	25
4.3.3. Remplir formulaire	26
4.3.4. Modifier	26
4.3.5. Enregistrer	26
4.3.6. Valider	27
5. Les diagrammes de séquence	28
5.1 Administrateur	28
5.1.1 s'authentifier	28
5.1.2 créer compte pédagogie	29
5.2 agent pédagogie	30
5.2.1 rechercher	30
5.2.2 supprimer compte étudiant	31

5.3 Etudiant	32
5.3.1 créer compte étudiant	32
5.3.2 s'inscrire	33
5.3.3 remplir formulaire	33
5.3.4 modifier	34
5.3.5 Enregistrer	34
5.3.6 Valider	35
6. Les Diagrammes d'activités	36
6.1 Administrateur	36
6.1.1 s'authentifier	36
6.1.2 créer compte pédagogie	36
6.2 agent pédagogie	37
6.2.1 rechercher	37
6.2.2 supprimer compte étudiant	38
6.3 étudiant	38
6.3.1 créer compte étudiant	38
6.3.2 s'inscrire	39
6.3.3 remplir formulaire	39
6.3.4 modifier	40
6.3.5 enregistrer	40
6.3.6 valider	41
7. Les diagrammes de classe	42
8. les diagrammes de la classe participants	43
8.1 administrateur	43
8.1.1 s'authentifier	43
8.1.2 créer compte pédagogie	43
8.2 agent pédagogie	43
8.2.1 rechercher	43
8.2.2 supprimer compte étudiant	44
8.3 étudiant	44
8.3.1 créer compte étudiant	44
8.3.2 s'inscrire	45
8.3.3 remplir formulaire	46
8.3.4 modifier	47
8.3.5 enregistrer	48
8.3.6 valider	49
9. les diagrammes de la classe de conception	50
9.1 administrateur	50
9.1.1 s'authentifier	50
9.1.2 créer compte pédagogie	51

9.2	agent pédagogie	51
9.2.1	rechercher	51
9.2.2	supprimer compte étudiant	52
9.3	étudiant	53
9.3.1	créer compte étudiant	53
9.3.2	s'inscrire	54
9.3.3	remplir formulaire	55
9.3.4	modifier	56
9.3.5	enregistrer	57
9.3.6	valider	58
10.	conclusion	58
<i>Chapitre 04 : l'implémentation</i>		
1.	Introduction	60
2.	Outils de travail	60
2.1-	L'environnement de développement	60
2.1.1	mozilla firefox	60
2.1.2	notepad++	60
2.1.3	serveur apache et easy php	61
2.1.4	pacestar UML diagrammer 6.02	61
2.2-	Les langages de programmation	61
2.2.1	coté client	61
a-	html	61
b-	java script	62
c-	css	62
2.2.2	coté serveur	63
a-	php	63
b-	mysql	63
3.	passage de modèle objet au modèle relationnel	64
4.	le plan de site.	66
5.	manuel d'utilisation de site.	67
❖	<i>Conclusion générale</i>	74

Liste des figures

Chapitre	Figures	Page
Chapitre 01	Figure 1.1 : Modèle Client /Serveur	4
	Figure 1.2 : Architecture à deux niveaux	4
	Figure 1.3 : Architecture Multi niveau	5
	Figure 1.4 : Architecture CGI	8
Chapitre 02	Figure 2.1 : les trois axes de modélisation	13
Chapitre 03	Figure 3.1 : Diagramme de contexte	20
	Figure 3.2 : Diagramme de cas d'utilisation	21
	Figure 3.3 : Diagramme de séquence de cas d'utilisation 'authentifier	27
	Figure 3.4 : Diagramme de séquence de cas d'utilisation créer compte pédagogie	28
	Figure 3.5 : Diagramme de séquence de cas d'utilisation recherché	29
	Figure 3.6 : Diagramme de séquence de cas d'utilisation supprimer compte étudiant	30
	Figure 3.7 : Diagramme de séquence de cas d'utilisation créer compte étudiant	31
	Figure 3.8 : diagramme de séquence de cas d'utilisation s'inscrire	32
	Figure 3.9 : diagramme de séquence de cas d'utilisation remplir formulaire.	32
	Figure 3.10 : diagramme de séquence de cas d'utilisation modifier	33
	Figure 3.11 : diagramme de séquence de cas d'utilisation enregistrer	33
	Figure 3.12 : diagramme de séquence de cas	34

	d'utilisation valider	
	Figure 3.13 : diagramme d'activité de cas d'utilisation authentifier	35
	Figure 3.14 : diagramme d'activité de cas d'utilisation créer compte pédagogie	35
	Figure 3.15 : diagramme d'activité de cas d'utilisation rechercher	36
	Figure 3.16 : diagramme d'activité de cas d'utilisation supprimer compte étudiant	37
	Figure 3.17 : diagramme d'activité de cas d'utilisation créer compte étudiant	37
	Figure 3.18 : diagramme d'activité de cas d'utilisation s'inscrire	38
	Figure 3.19 : diagramme d'activité de cas d'utilisation remplir les formulaires.	38
	Figure 3.20 : diagramme d'activité de cas d'utilisation modifier	39
	Figure 3.21 : diagramme d'activité de cas d'utilisation enregistrer	39
	Figure 3.22 : diagramme d'activité de cas d'utilisation valider	40
	Figure 3.23 : le diagramme de classe	41
	Figure 3.24 : diagramme de classe participant de cu authentifier.	42
	Figure 3.25 : diagramme de classe participant de cas d'utilisation créer compte pédagogie.	42
	Figure 3.26 : diagramme de classe participant de cas d'utilisation rechercher.	42
	Figure 3.27 : diagramme de classe participant de cas d'utilisation supprimer compte étudiant.	43
	Figure 3.28 : diagramme de classe participant de cas d'utilisation créer compte étudiant.	43
	Figure 3.29 : diagramme de classe participant	44

	de cas d'utilisation inscrire.	
	Figure 3.30 : diagramme de classe participant de cas d'utilisation remplir formulaire.	45
	Figure 3.31 : diagramme de classe participant de cas d'utilisation modifier.	46
	Figure 3.32 : diagramme de classe participant de cas d'utilisation enregistrer.	47
	Figure 3.33 : diagramme de classe participant de cas d'utilisation valider.	48
	Figure 3.34 : diagramme de classe de conception de cas d'utilisation authentifier.	49
	Figure 3.35 : diagramme de classe de conception de cas d'utilisation créer compte pédagogie.	50
	Figure 3.36 : diagramme de classe de conception de cu rechercher.	50
	Figure 3.37 : diagramme de classe de conception de cas d'utilisation supprimer compte étudiant	51
	Figure 3.38 : diagramme de classe de conception de cas d'utilisation créer compte étudiant.	52
	Figure 3.39 : diagramme de classe de conception de cas d'utilisation inscrire.	53
	Figure 3.40 : diagramme de classe de conception de cas d'utilisation remplir formulaire.	54
	Figure 3.41 : diagramme de classe de conception de cas d'utilisation modifier.	55
	Figure 3.42 : diagramme de classe de conception de cas d'utilisation enregistrer.	56
	Figure 3.43 : diagramme de classe de conception de cas d'utilisation valider	57

Chapitre 04	Figure 4.1 : page d'accueil	67
	Figure 4.2 : page créer compte.	68
	Figure 4.3 : page d'étudiant	69
	Figure 4 .4 : page remplir formulaires.	70
	Figure 4.5 : page agent pédagogie.	71
	Figure 4.6 : page administrateur.	72

Liste des tableaux

Chapitre	Tableau	Page
Chapitre 03	Tableau 3.1 : fiche d'inscription de cas d'utilisation authentifier.	22
	Tableau 3.2 : fiche d'inscription de cas d'utilisation créer compte pédagogie.	22
	Tableau 3.3 : fiche d'inscription de cas d'utilisation rechercher.	23
	Tableau 3.4 : fiche d'inscription de cas d'utilisation supprimer compte étudiant.	23
	Tableau 3.5 : fiche d'inscription de cas d'utilisation créer compte étudiant.	24
	Tableau 3.6 : fiche d'inscription de cas d'utilisation inscrire.	24
	Tableau 3.7 : fiche d'inscription de cas d'utilisation remplir formulaire.	25
	Tableau 3.8 : fiche d'inscription de cas d'utilisation modifier.	25
	Tableau 3.9 : fiche d'inscription de cas d'utilisation enregistrer.	26
	Tableau 3.10 : fiche	26

	d'inscription de cas d'utilisation valider	
Chapitre 04	Tableaux 4.1: table du passage du modèle objet au modèle relationnel	64

Introduction Générale

Le Web est le service qui a placé Internet sous les feux de projecteur, intéressant ainsi les medias et le grand public. Il était un support bien adapté pour la coopération sur un réseau à grande distance, pour permettre par exemple à plusieurs personnes de publier et maintenir un ensemble de documents consultable sur un serveur.

L'Internet offre des possibilités considérables et accroît les chances de communication à travers un lieu virtuel qui regroupe des pages Internet présentées par des personnes ou des organisations, c'est le site web.

Les anciens sites Web permettaient des applications simples supportant la navigation et la recherche d'information dans un cadre restreint pour une simple consultation (lecture seul). Aujourd'hui leur évolution a permis de supporter des applications interactives avec des opérations affectant le contenu (information) et l'état de la navigation ainsi que des opérations intégrées dotant d'une capacité de recherche prédéfinie des hypermédia, d'où la convergence entre le domaine de l'hypermédia et les systèmes d'information qui adoptent de plus en plus la navigation comme paradigme d'interaction fondamental.

L'internet va prendre en charge les besoins de l'utilisateur, et met à ça disposition le développement des applications dynamiques qui est le sujet de notre projet.

Ce projet est divisé en quatre chapitres en plus de l'introduction générale qui présente le plan de se travaille et la conclusion générale.

Dans le premier chapitre, nous définissons quelques concepts jugés nécessaires sur le principe, architecture, caractéristique, les avantages et les inconvénients de modèle client / serveur, ainsi que le Web, programmation Web, Web et base de données et la sécurité Web.

Le deuxième chapitre explique le langage UML avec ses caractéristiques et diagrammes puis la démarche de développement que nous avons adopté pour aboutir à notre système.

Le troisième chapitre constitue l'essentiel de notre travail ; en effet il décrit de façon détaillée toutes les phases que nous avons suivies pour parvenir au logiciel, démarrant de

l'étape identification des besoins, phase d'analyse et la phase conception détaillée de notre application.

Chaque phase contient des diagrammes spéciaux comme le diagramme de contexte, diagramme de cas d'utilisation et le diagramme de séquence dans la phase d'identification de besoins, en passant à la phase d'analyse qui contient le diagramme d'activité, le diagramme de classe, et à la fin en parle de la phase de conception qui contient le diagramme de classe participant et le diagramme de classe de conception.

Le quatrième chapitre présente les outils de développement que nous avons utilisé tels que : PHP, HTML,et expose également quelques interfaces de l'application.

Nous terminons ce mémoire par une conclusion générale, qualificatif des objectifs déjà tracé au début. Nous aborderons aussi les perspectives de notre travail ainsi que les nouvelles améliorations qu'on peut apporter pour compléter la réalisation.

1. Introduction

Le modèle client/serveur est une architecture client/serveur permettant le traitement coopératif d'applications, c'est-à-dire la communication directe des deux applications via un réseau ou encore la communication directe entre deux processus d'une même application. La partie client d'une application est souvent optimisée pour l'interaction avec l'utilisateur, alors que la partie serveur fournit les fonctions multi utilisateurs

2. Principe et Architecture du modèle Client / Serveur

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles des fichiers, des statistiques, des données multimédias, etc. Les services sont exploités par des programmes, appelés clients, s'exécutant sur les machines clientes. On parle ainsi de client (client FTP, client de messagerie, etc.) lorsque l'on désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client de messagerie il s'agit du courrier électronique). [1]

2.1 Principe du modèle Client /Serveur

Client/serveur désigne un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs clients du serveur : Le client émet une requête vers le serveur grâce à son adresse IP et son port, qui désigne un service particulier du serveur. Le serveur reçoit la demande, la traite et renvoie la réponse à l'aide de l'adresse IP de la machine client et son port. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique.

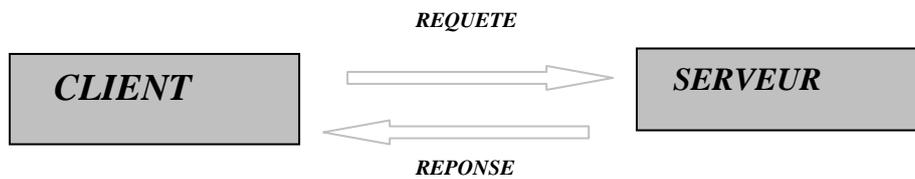


Figure 1.1 : *Modèle Client /Serveur*

2.2 Architecture des systèmes Client / Serveur

L'architecture client/serveur possède deux types d'ordinateurs sur un réseau : les clients et les serveurs, cette architecture possède donc deux niveaux et s'appelle two-tier en anglais. Les architectures multi-tier (ou distribuées) divisent le serveur en plusieurs entités (par exemple, un serveur d'application qui lui-même est le client d'un serveur de base de données).

- *Architecture à deux niveaux :*

Aussi appelée architecture 2-tier (étage en anglais), caractérise les systèmes client / serveur dans lesquelles le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service.

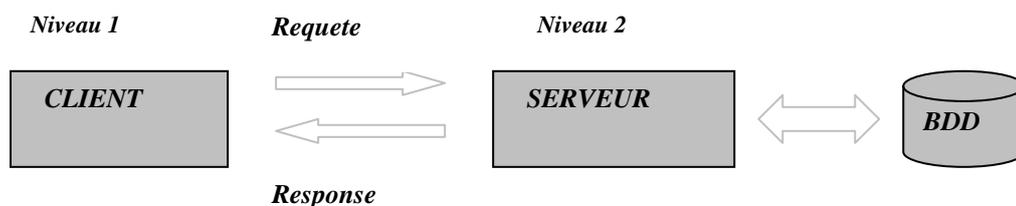


Figure 1.2 : *Architecture à deux niveaux*

- *Architecture Multi niveaux :*

Dans l'architecture à N niveaux chaque serveur effectue une tâche (un service) spécialisée. Ainsi un serveur peut utiliser les services d'un ou plusieurs autres serveurs afin de fournir ses propres services. [3]

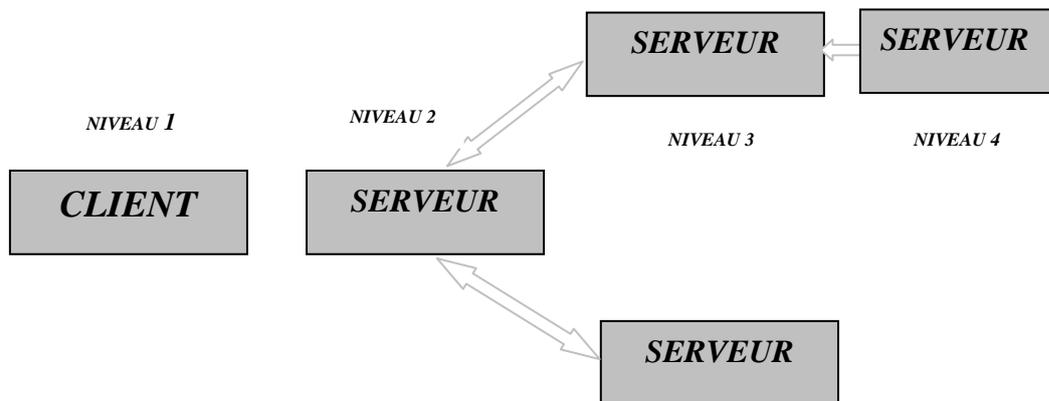


Figure 1.3 : Architecture Multi niveau

3. Caractéristiques du modèle Client/Serveur :

- **Le partage de ressources**

Un serveur peut traiter plusieurs clients en même temps et contrôler leurs accès aux ressources.

- **Echange de messages**

Clients et serveurs sont des systèmes à couplage faible qui interagissent au moyen de messages. Le message est le mécanisme d'émission de demandes des services et des réponses à celles-ci.

- **Intégrité**

Le code et les données du serveur sont gérés de façon centralisée, ce qui garantit un moindre coût de maintenance et une meilleure intégrité de données partagées, et les clients restent indépendants.

4. Les avantages et les inconvénients :

- **Avantages:**

- une meilleure sécurité
- un réseau évolutif
- une administration au niveau serveur

- **Inconvénients:**

- un coût élevé
- Complexité de la programmation

5. Le Web

Le World Wide Web, littéralement la « toile d'araignée mondiale », communément appelé le Web, parfois la Toile ou le WWW, est un système hypertexte public fonctionnant sur Internet qui permet de consulter, avec un navigateur, des pages accessibles sur des sites. L'image de la toile d'araignée vient des hyperliens qui lient les pages web entre elles.

Le concept du Web a été mis au point au CERN (Centre Européen de Recherche Nucléaire) en 1991 par une équipe de chercheurs à laquelle appartenaient Tim-Berners LEE, le créateur du concept d'hyperlien, considéré aujourd'hui comme le père fondateur du Web.

Le principe de Web repose sur l'utilisation d'hyperliens pour naviguer entre des documents (appelés «pages web») grâce à un logiciel appelé navigateur (en anglais browser). Une page web est ainsi un simple fichier texte écrit dans un langage de description appelé HTML , permettant de décrire la mise en page du document et d'inclure des éléments graphiques ou bien des liens vers d'autres documents à l'aide de balises. Au-delà des liens reliant des documents formatés, le web prend tout son sens avec le protocole HTTP permettant de lier des documents hébergés par des ordinateurs distants (appelés serveurs web, par opposition au client que représente le navigateur). [6]

5.1 Serveur Web

Est un ordinateur qui répond aux requêtes émises par le client web. Un serveur Web peut être :

- Un ordinateur tenant le rôle de serveur informatique sur lequel fonctionne un logiciel serveur HTTP.
- Le serveur HTTP lui-même
- Un ensemble de serveurs garantissant le fonctionnement d'applications Web. Le plus souvent, un serveur Web fait fonctionner plusieurs logiciels qui s'exécutent en parallèle. On retrouve la combinaison Apache (serveur HTTP), MySQL (serveur de base de données) et PHP.

5.2 Client Web

Est un logiciel utilisé pour demander les pages web du serveur web mentionné dans le champ adresse et puis les afficher pour l'utilisateur. Il est appelé aussi navigateur web pour communiquer avec le serveur web doit utiliser un protocole.

Les deux navigateurs les plus utilisés sont Internet Explorer de Microsoft, et Mozilla Firefox.

5.3 Site Web

Un site Web est un ensemble de fichiers HTML stockés sur un ordinateur connecté en permanence à Internet et hébergeant les pages Web. Un site web est habituellement architecturé autour d'une page centrale, appelée «page d'accueil» et proposant des liens vers un ensemble d'autres pages ou ressources hébergées sur le même serveur, et parfois des liens dits «externes», c'est-à-dire de pages hébergées par un autre serveur.

6. Programmation Web

6.1 Le Web statique

- **Le Langage HTML** : (Hypertexte markup langage) c'est un langage qui permet de construire les pages web. Ce n'est pas un langage de programmation, c'est un langage de description qui permet de décrire l'aspect d'un document, d'y inclure des informations variées (texte, image, sons, etc.) et établir les liens entre les pages grâce aux liens hypertexte.
- **Pages statiques** : Les pages Web définies ainsi sont dites statiques, leur contenu est fixé à l'avance, et l'utilisateur ne dispose d'aucun moyen d'action pour sélectionner les informations qui l'intéressent.

Du côté serveur, la maintenance du site implique la manipulation d'un ensemble de pages qui peut rapidement devenir volumineux. [3]

6.2 Web dynamique

- **Pages dynamiques** : Une page Web dynamique est une page Web générée à la demande, par opposition à une page Web statique. Le contenu d'une page Web dynamique peut donc varier en fonction d'informations (heure, nom de l'utilisateur, formulaire rempli par l'utilisateur, etc.) qui ne sont connues qu'au moment de sa consultation. A l'inverse, le contenu d'une page Web statique est a priori identique à chaque consultation. [2]
- **Formulaire** : Les formulaires constituent un moyen privilégié d'interaction permettant à l'utilisateur de formuler ses données par l'intermédiaire des champs de saisie. Les formulaires permettent de transmettre les paramètres saisis par le client au serveur pour être traités et produire les réponses correspondantes par la suite.

Tous les paramètres saisis dans un formulaire sont transférés au serveur et plus précisément au script (programme) associé à ce formulaire, qui doit produire un document (généralement un document HTML) et le transmettre au navigateur du client.

Les techniques de production de réponses sont variées, chacune d'elles présente des avantages et des inconvénients et dont le choix dépend de plusieurs paramètres. Comme exemple illustratif de ces techniques nous présentons le CGI. [9]

6.3 Le Common Gateway Interface CGI

Constitue la technique traditionnelle, son principe consiste à produire des documents HTML par des programmes résidants sur le serveur qui reçoivent en entrée les paramètres prévenants d'un client par l'intermédiaire des formulaires.

Les programmes CGI peuvent être écrits en n'importe quel langage de programmation (C, C++, Perl,...) et le programmeur est libre de faire toutes les opérations nécessaires pour satisfaire les demandes des clients (dans la limite des droits d'accès).

La figure suivante présente l'architecture CGI. [3]

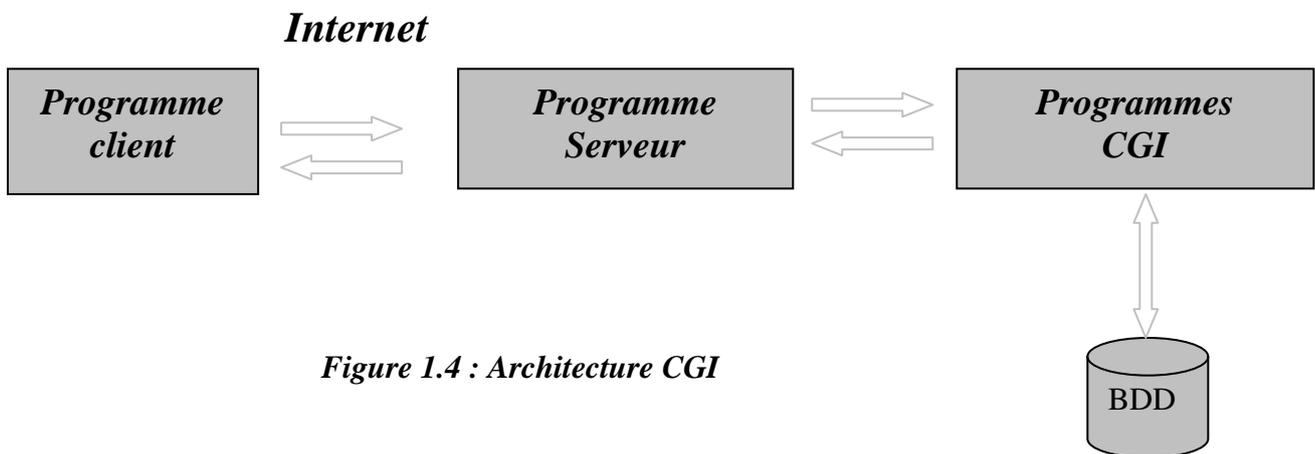


Figure 1.4 : Architecture CGI

7. Web et bases de données

7.1 Accès aux bases de données

Une base de données est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. L'accès aux bases de données via le Web permet de mettre des données à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour

7.2 Avantages cumulés du Web et des bases de données

On utilise conjointement ces deux types de technologies : bases de données et Web pour construire une application Web (inscription en ligne spécialement) qui bénéficie des avantages de ces deux technologies.

- **Données hypermédiées et données sécurisées :** Le Web apporte la puissance la convivialité de la navigation hypertexte pour accéder à des données hétérogènes

riches (textuelles, multimédias, binaires ...).

L'interface avec un SGBD permet d'accéder à des données plus structurées et Plus homogènes et extraire les informations voulues.

- **Portabilité et ouverture** : En ajoutant les standards du Web aux standards des bases de données on obtient une architecture ouverte, qu'on construit au dessus des applications portables grâce à HTML, http, SQLetc.
- **Homogénéité** : On utilise une interface homme/machine unique (navigateur Web) pour accéder à des applications situées sur un serveur, ce qui simplifie leur utilisation.
- **Sécurité** : On pourra utiliser conjointement les techniques de sécurité du Web et des SGBD pour avoir un niveau de sécurité plus élevé.

8. Sécurité sur le Web

Vol d'identité, des informations très personnelles, de coordonnées bancaires, sont autant de problème de sécurité qui émerge de nos jours. Nous utilisons tous Internet tous les jours, et celui-ci est devenu vecteur d'attaques très modernes, visant le plus souvent à soutirer de l'argent, ou se faire passer pour quelqu'un d'autre.

Avec le développement de l'utilisation du Web, de plus en plus d'entreprises ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs sur Internet, il est donc essentiel de protéger les ressources de l'entreprise à et de maîtriser le contrôle d'accès et les droits des utilisateurs du système d'information. [3]

8.1 Objectifs de la sécurité Web:

Le système d'information est généralement défini par l'ensemble des données et des ressources matérielles et logicielles de l'entreprise permettant de les stocker ou de les faire circuler. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger. La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu. La sécurité informatique vise généralement cinq principaux objectifs :

- **La confidentialité** : La confidentialité consiste à rendre l'information inintelligible à d'autres personnes que les seuls acteurs de la transaction.
- **L'intégrité** : Vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle).

- **La disponibilité** : L'objectif de la disponibilité est de garantir l'accès à un service ou à des ressources.
- **La non répudiation** : La non répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.
- **L'authentification** : L'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées. [1]
- **La cryptographie** : Le chiffrement des données fut inventé pour assurer la confidentialité des données. Il est assuré par un système de clé (algorithme) appliqué sur le message. Ce dernier est décryptable par une clé unique correspondant au cryptage.

9. conclusion

Tout au début de ce chapitre, on a discuté le principe du modèle client/serveur et ces différentes architectures, ainsi qu'on a parlé sur l'histoire du Web en passant par la programmation Web, les bases de données et en fin la sécurité sur le web .

1. Introduction

Ce chapitre présente les concepts de base du langage de modélisation UML, pour permettre une bonne lecture de ce qui va suivre et exprimer de manière uniforme l'analyse, la conception et la réalisation d'une application informatique. Par la suite nous abordons les notions d'UML (Unified Modeling Languages) avec ses neuf diagrammes.

2. Définition et historique d'UML

UML (Unified Modeling Language) est le langage de modalisation unifié qui est le résultat d'une opération d'unification d'un ensemble de concepts pris des méthodes orientées objet dans le but de modéliser d'une façon claire et précise la structure et le comportement d'un système indépendamment de toute méthode et tout langage de programmation.

UML a été accepté comme standard industriel en novembre 1997 par le groupe OMG sous le patronat de trois méthodologues orientées objet expérimentes : J.Rumbaugh, I.Jacobson et G.Booch. La modélisation orientées objet dans tous les domaines est aujourd'hui dominée par UML.

3. Caractéristiques d'UML

- **UML n'est pas une méthode ou un processus :** UML est un langage qui modélise les activités de l'entreprise et pas une méthode ou processus qui régit l'enchaînement de ces activités.
- **UML est un langage semi formel :** UML est fondé sur un méta modèle, qui définit les éléments de modélisation (les concepts manipulés par le langage) et leur sémantique, et permet de classer ces éléments selon leur niveau d'abstraction ou domaine d'application, ce méta modèle limite les ambiguïtés et encourage la construction d'outils. [5]
- **UML a une représentation graphique standardisée :** qui permet à tout les développeurs de l'aborder, l'appréhendant et la comprennent d'une seule et même manière.
- **UML est un cadre d'analyse d'objet, on offrant :** Différentes vues (perspectives) complémentaires d'un système, qui guide l'utilisation des concepts objets et plusieurs niveaux d'abstraction, qui permettent de mieux contrôler la complexité dans l'expression des solutions objets. [5]
- **UML est visuel** (un ensemble de diagramme) un tel langage est peu compréhensible qu'un langage textuel.

- **UML est complet**: tous les aspects d'un système peuvent être décrits. [5]

4. Les Axes de la Modélisation

Les concepteurs orientent leurs modélisations selon trois axes sur lesquels ils répartissent les diagrammes :

- L'axe fonctionnel qui est utilisé pour décrire ce que fait le système à réaliser.
- L'axe structurel et statique qui est relatif à sa structure.
- L'axe dynamique qui est relatif à la construction de ses fonctionnalités.

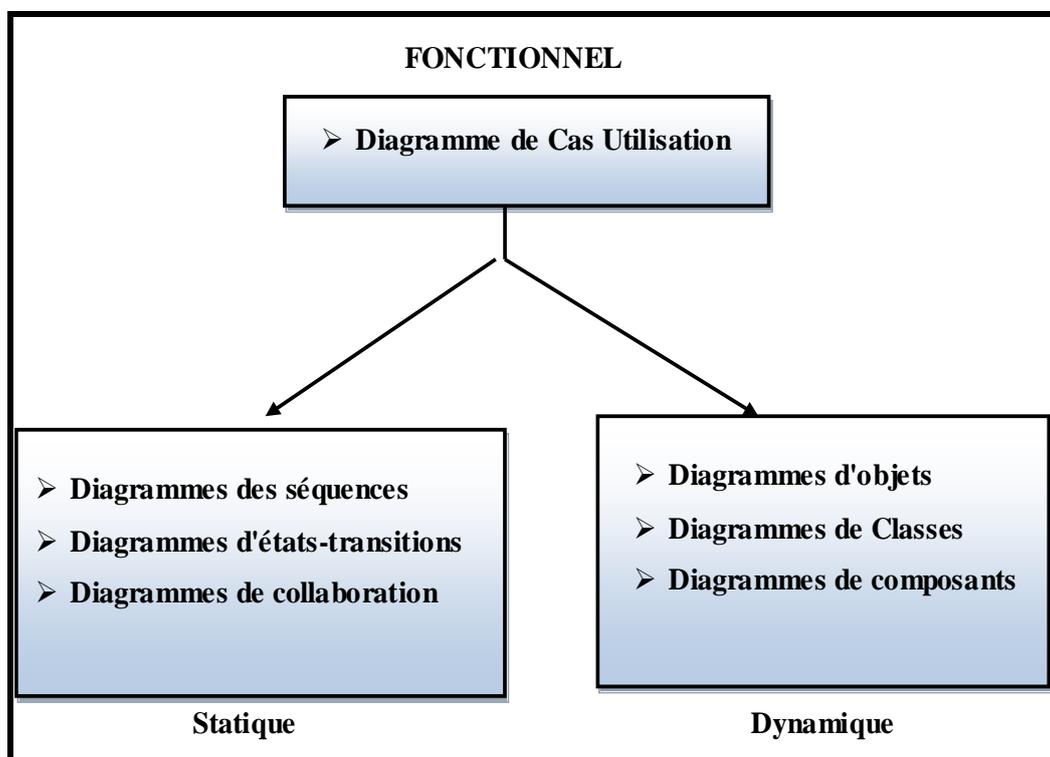


Figure 2.1 : Les trois axes de modélisation

5. Les diagrammes UML

UML propose de décrire un système à l'aide essentiellement de 9 diagrammes. Chacun de ces diagrammes correspond à la définition d'une partie du système selon un point de vue particulier.

5.1 Le diagramme de cas d'utilisation : sert à représenter les besoins d'utilisateurs par rapport au système et comprend les acteurs, le système et les cas d'utilisation eux-mêmes.

Les cas d'utilisation : décrivent sous la forme d'actions et de réactions le comportement d'un système du point de vue d'un utilisateur. Ils permettent de définir les limites du système et les relations entre le système et l'environnement.

Acteur: représente le rôle joué par une personne ou une chose interagissant avec un système. Il existe quatre catégories d'acteurs: principaux, secondaires, le matériel externe, et les autres systèmes.

5.2 Le diagramme de classe : représente la description statique de système en intégrant dans chaque classe, la description des données et des traitements. C'est le diagramme pivot de la modélisation objet d'un système et comprend les classes et les associations entre eux.

Les classes : Une classe est la description d'un ensemble d'objets ayant une sémantique, des attributs, des méthodes et des relations en commun. Elle spécifie l'ensemble des caractéristiques qui composent des objets de même type.

Association : Une association est une relation structurelle entre objets. Elle est modélisée par un trait entre classes. Cette trait peut être qualifiée avec le type de relation, et peut également comporter des règles de multiplicité par exemple un à un, un à plusieurs, plusieurs à plusieurs pour la relation.

5.3 Le diagramme d'objet : montrent des objets et des liens entre eux, comme les diagrammes de classes, les diagrammes d'objets représentent la structure classique.

Objet : Des objets sont identifiés en plaçant le nom d'instance suivi des deux points (:) devant le nom de la classe. Les valeurs de propriété sont écrites comme des paires "nom=valeur". L'icône représentant un objet est un rectangle divisé en sections.

Association : Le diagramme des objets peut contenir également des associations. Souvent, les contraintes, le détail des relations et les règles de multiplicité trouvées dans le diagramme de classe ne sont pas représentées pour ne se concentrer que sur les objets et leurs propriétés. Les associations entre les objets sont représentées simplement en utilisant une ligne les joignant. [7]

5.4 Le diagramme état transition : le diagramme état transition fait partie des modèles dynamiques qui décrit l'enchaînement de tous les états d'un objet.

Etat : La notation de l'état décrit le mode de l'entité. Elle est représentée par rectangle avec les coins arrondis, contenant state le nom de l'état.

Transition : Une transition décrit le changement de l'état d'un objet, provoqué par un événement. La notation utilisée pour représenter une transition est une flèche, avec le nom d'événement écrit au-dessus, au-dessous, ou à côté. [8]

5.5 Le diagramme d'activités : représente le comportement d'une opération ou d'un cas d'utilisation, en terme d'action le diagramme d'activité peut être un classe (de type processus, donc à un cas d'utilisation). En option donc une description du déroulement des activités des processus. le DAC sert à consolider la séquence d'actions décrites dans la fiche descriptive d'un CU, il permet de dessiner dans un même DAC les défèrent scénarios d'un même CU, et dans ce cas plusieurs classes peuvent intervenir dans le DAC.

5.6 Le diagramme de la collaboration : décrit les scénarios de chaque cas d'utilisation en mettant l'accent sur l'interaction des objets et les messages échangés.

5.7 Le diagramme de séquence : est une autre forme de représentation des scénarios qui met l'accent sur la chronologie des opérations d'interaction entre les objets

Objet : Les objets sont des instances des classes, et sont rangés horizontalement. La représentation graphique pour un objet est similaire à une classe (un rectangle) précédée du nom d'objet (facultatif) et d'un point-virgule (;) .

Acteur : Les acteurs peuvent également communiquer avec des objets, ainsi ils peuvent eux aussi être énumérés en colonne. Un acteur est modélisé en utilisant le symbole habituel: Stick Man.

Ligne de vie : Les lignes de vie, Life Line, identifient l'existence de l'objet par rapport au temps. La notation utilisée pour une ligne de vie est une ligne pointillée verticale partant de l'objet.

Activation : Les activations, sont modélisées par des boîtes rectangulaires sur la ligne de vie. Elles indiquent quand l'objet effectue une action.

Message : Les messages, modélisés par des flèches horizontales entre les activations, indiquent les communications entre les objets. [7]

5.8 Le diagramme de composants : représente les différents constituants logiciels d'un Système (architecture logicielle d'un système).

5.9 Le diagramme de déploiement : décrit l'architecture technique d'un système (Composants matériels du système informatique et leurs connexions).

6. La démarche de conception de l'application :

6.1 Introduction

Nous allons donc, proposer une démarche de modélisation nécessaire et suffisante pour construire efficacement un site Web. Pour cela nous utilisons un sous-ensemble du langage de modélisation UML. Qui sera de même nécessaire pour la plupart des projets. De même nature cette approche est le résultat de plusieurs années, elle a donc montré son efficacité dans la pratique.

6.2 La démarche de conception

La démarche est structurée en 03 étapes :

- Phase d'identification des besoins.
- phase d'analyse.
- phase de conception.

a. Première étape : Identification des besoins

Cette étape permet de faire un premier repérage des besoins fonctionnels et techniques en utilisant les textes et préparant la description des cas d'utilisation. Identification des besoins se décompose en plusieurs phases :

1. Présentation du projet : Consiste à faire une description générale des activités principales du site.
2. Les besoins fonctionnels : Permet de faire une description des grandes fonctionnalités attendues du site.
3. Les besoins techniques : Permet d'expliquer les besoins liés à l'exploitation du site
4. L'identification des acteurs.
5. Le diagramme de contexte : Permet de :
 - a. Identification des messages
 - b. Réalisation de diagramme de contexte [5]

6. diagramme de cas d'utilisation : Permet de recenser les grandes fonctionnalités d'un système, alors on peut élaborer les cas d'utilisation déclenchés par les acteurs du système. Pour chaque cas d'utilisation, on fait une fiche descriptive. Chaque cas se traduit à un nombre de scénarios et pour chaque scénario on peut spécifier un diagramme de séquence.

b. Deuxième étape : Phase d'analyse

A partir des diagrammes de séquence on peut extraire les classes existantes et les diagrammes d'activités :

- ***Le diagramme de classe*** (modèle de domaine) doit définir les classes qui modélisent les entités ou concepts présents dans le domaine (on utilise aussi le terme métier) de l'application. Il s'agit donc de produire un modèle des objets du monde réel dans un domaine donné. Il faut absolument utiliser le vocabulaire du métier pour nommer les classes et leurs attributs. Les classes du modèle du domaine ne doivent pas contenir d'opération, mais seulement des attributs.
- ***Diagramme de classes participantes*** : Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre les classes d'un cas d'utilisation, et le modèle du domaine, et d'autre part, les diagrammes de conception logicielle n'apparaissent pas encore à ce niveau. Il n'est pas souhaitable que les utilisateurs interagissent directement avec les instances des classes des domaines par le biais de l'interface graphique. En effet, le modèle du domaine doit être indépendant des utilisateurs et de l'interface graphique. De même, l'interface graphique du logiciel doit pouvoir évaluer sans répercussion sur le cœur de l'application.
- ***Diagramme d'activité*** : le diagramme d'activité représente l'enchaînement des différents scénarios des cas d'utilisations.

c. Troisième étape : Phase de conception

- Les diagrammes de classes de conception représentent bien la structure statique du code lié par le biais des attributs et des relations entre classes ; mais ils contiennent également les opérations qui décrivent les responsabilités (propriétés) dynamique des

classes logicielles. Pour chaque services ou fonction il faut décider quelle est la classe qui va le / la contenir. Nous devons aussi répartir tout le comportement du système entre les classes de conception et décrire les collaborations induites. [4]

7. Conclusion

Dans ce chapitre, on a présenté les notions essentielles d'UML et ses diagrammes qu'on va utiliser pour le développement de notre projet, puis on a vu les différentes étapes de notre démarche conception qu'on va suivre dans notre projet.

1. Cahier de charge :

a. Présentation du projet : L'activité principale du site est de permettre aux étudiants de s'inscrire automatiquement à une filière au Centre Universitaire de Mila.

b. Les besoins fonctionnels :

- Administrateur :

- ❖ crée des comptes pour les agents de la pédagogie.
- ❖ Peut faire toutes les opérations que les autres acteurs peuvent faire.

- Agent de la pédagogie :

- ❖ Rechercher les étudiants : l'agent peut rechercher les étudiants par nom ou par un numéro provisoire.
- ❖ l'agent peut faire toutes les opérations que l'étudiant peut par exemple : supprimer compte d'étudiant, imprimer une carte d'étudiant..Etc.

- Etudiant :

- ❖ Créer un compte étudiant : l'étudiant peut créer un compte pour son inscription sur le site.
- ❖ Modifier ses informations plusieurs fois (avant la validation finale)
- ❖ Valider son inscription après saisie de toutes les informations.
- ❖ Imprimer une Fiche d'inscription provisoire après la validation des informations.
- ❖ Pouvoir annuler et supprimer son compte avant la validation.

c. Les besoins techniques :

- Interface simple et conviviale : pour simplicité l'utilisation du site
- La sécurité : le système doit permettre de :
 - a. créer du compte Agent pédagogie : un administrateur système est chargé de créer les comptes et définir les profils des agents de la pédagogie.
 - b. s'authentifier : vérifier l'autorisation d'accéder au système par l'étudiant et par l'agent.

d. Identification des acteurs :

- a. *Administrateur* : c'est la personne qui gère le système.
- b. *Agent pédagogie* : c'est la personne qui suit l'opération d'inscription des étudiants.
- c. *Etudiant* : c'est la personne qui accède au site pour s'inscrire.

2. Diagramme de contexte :

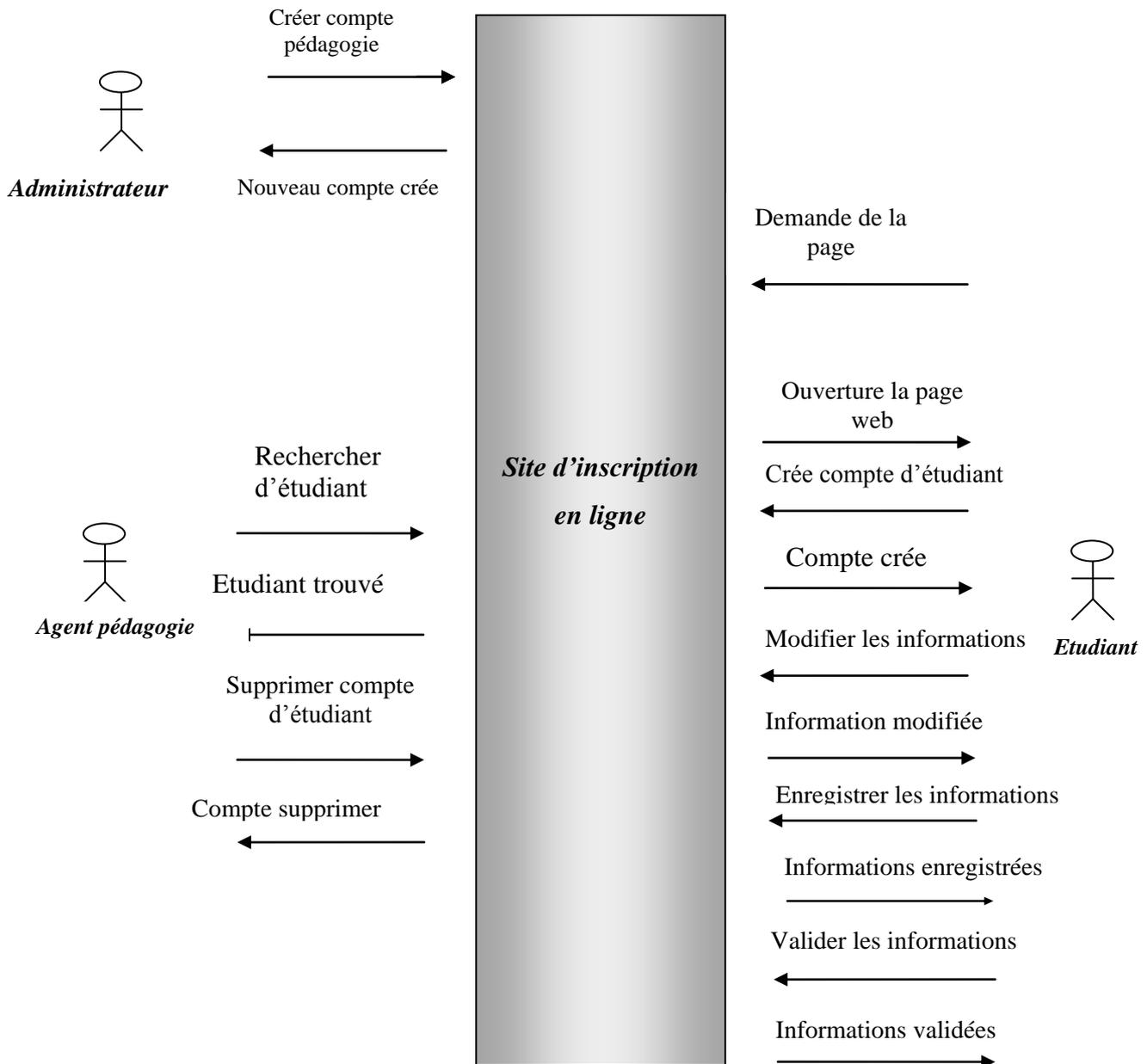


Figure 3.1 : Diagramme de contexte

3. Le diagramme de cas d'utilisation :

Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs / système et apporte une valeur ajoutée notable à l'acteur concerné. Il permet de

décrire ce que le futur système devra faire, sans spécifier comment il le fera. On exprimera donc des actions effectuées dans le cadre du métier de l'utilisateur, par opposition à des manipulations de l'application ou à des comportements technique. (4)

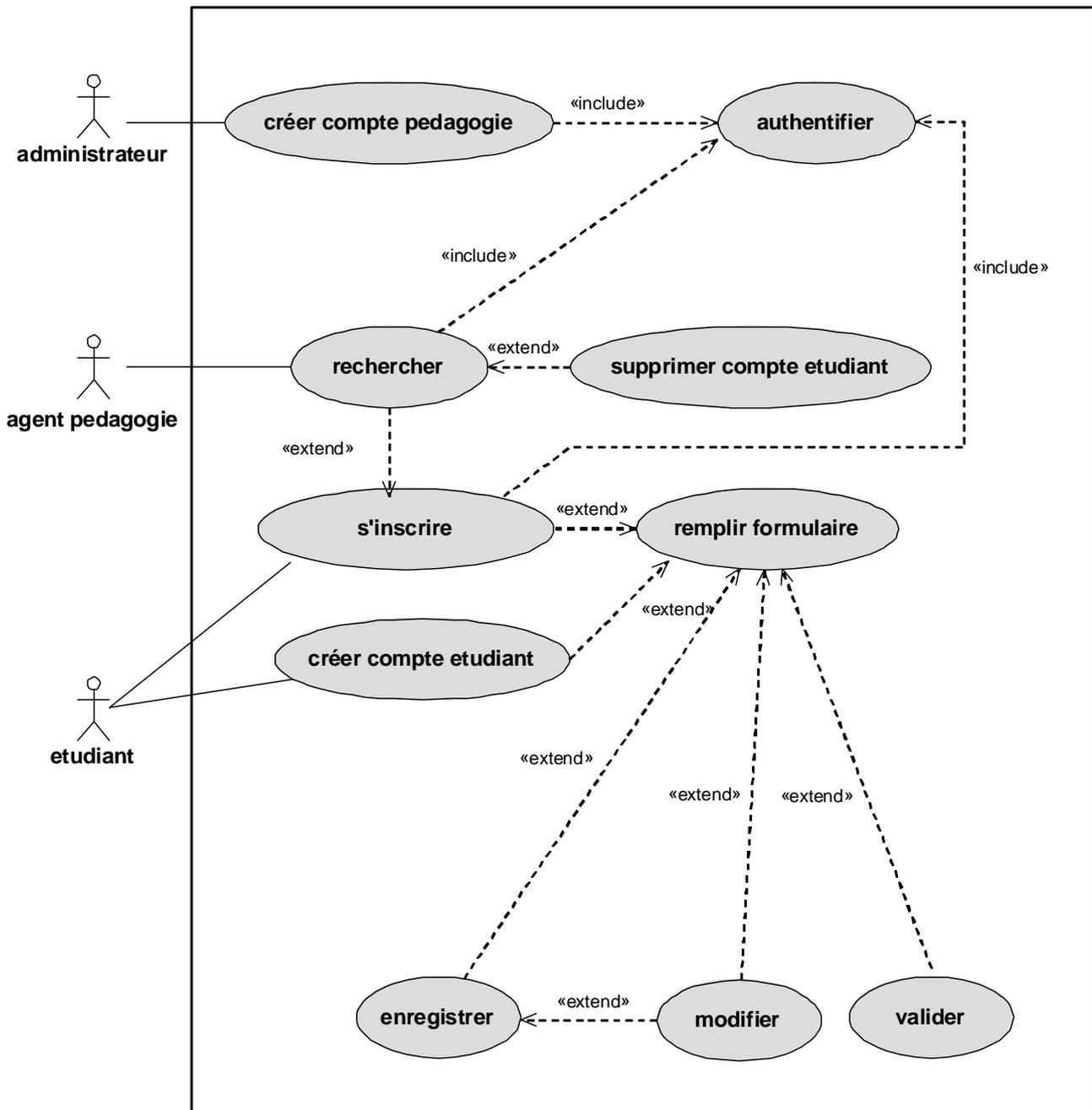


Figure 3.2 : Diagramme de cas d'utilisation

4. La description des cas d'utilisation :

4.1 Administrateur

4.1.1 S'authentifier

Cas d'utilisation	S'authentifier.
acteur	administrateur, agent pédagogie, étudiant.
But	Vérifier l'autorisation d'accès au système
Pré conditions	L'utilisateur possède un compte
Post conditions	L'utilisateur est authentifié par le système
Scénario nominal	<ul style="list-style-type: none"> ➤ L'utilisateur choisie d'ouvrir la page d'inscription. ➤ Le système demande à l'utilisateur d'entrer le nom et le mot de passe. ➤ L'utilisateur saisie le nom et le mot de passe et valide. ➤ Le système vérifie la validité du nom et mot de passe et ouvre la page d'accueil.
Scénario alternatif	<ul style="list-style-type: none"> ➤ Les données d'authentification sont erronées : <ul style="list-style-type: none"> ▪ Le système affiche un message d'erreur. ▪ Le système propose à l'administrateur ou à l'agent une nouvelle fois de saisir le nom et le mot de passe. ▪ Le système propose à l'étudiant de s'inscrire

Tableau 3.1 : Fiche de description du cas d'utilisation : authentifier

4.1.2 Créer compte pédagogie

Cas d'utilisation	Créer compte pédagogie.
acteur	Administrateur.
But	Permettre à des agents de suivre et traiter les inscriptions.
Pré conditions	L'administrateur doit être authentifié.
Post conditions	Un nouveau compte est créé.
Scénario nominal	<ul style="list-style-type: none"> ➤ L'administrateur demande la création d'un nouveau compte. ➤ Le système affiche le formulaire de création du compte. ➤ L'administrateur remplit le formulaire et valide. ➤ Le système vérifie et confirme la création du compte.
Scénario alternatif	<ul style="list-style-type: none"> ➤ Le compte à créer existe déjà : <ul style="list-style-type: none"> ▪ Le système affiche un message d'erreur. ➤ Les formulaires contiennent des champs vides ou incorrects : <ul style="list-style-type: none"> ▪ Le système indique à l'administrateur que le formulaire comporte des champs vides ou incorrects. ▪ Le système propose à l'administrateur de remplir les champs vides.

Tableau 3.2 : Fiche de description du cas d'utilisation : Créer compte pédagogie

4.2 Agent pédagogie

4.2.1 Rechercher

Cas d'utilisation	Rechercher
Acteur	Agent pédagogie.
But	<ul style="list-style-type: none"> ➤ Retrouver l'étudiant dans la liste des étudiants inscrits. ➤ Imprimer les cartes des étudiants.
Pré conditions	L'agent doit être authentifié.
Post conditions	L'étudiant recherché est trouvé ou non.
Scénario nominal	<ul style="list-style-type: none"> ➤ L'agent demande l'ouverture de la page de recherche. ➤ Le système ouvre la page demandée. ➤ Le système demande de saisir les critères de recherche. ➤ L'agent lance la recherche à par nom ou N⁰ d'inscription. ➤ Le système vérifie l'existence de l'étudiant. ➤ Le système affiche les informations concernées.
Scénario alternatif	<ul style="list-style-type: none"> ➤ L'étudiant n'est pas inscrit : <ul style="list-style-type: none"> ▪ Le système affiche un message d'absence.

Tableau 3.3 : Fiche de description du cas d'utilisation : rechercher

4.2.2 Supprimer compte d'étudiant

Cas d'utilisation	Supprimer compte d'étudiant.
Acteur	Agent pédagogie.
But	Supprimer totalement l'inscription de l'étudiant.
Pré conditions	<ul style="list-style-type: none"> • L'agent doit être authentifié. • L'étudiant existe dans la base de données.
Post conditions	Le compte d'étudiant est supprimé.
Scénario nominal	<ul style="list-style-type: none"> ➤ L'agent demander la page de suppression. ➤ Le système afficher la page demandé. ➤ Le système demande de saisir les critères de recherche. ➤ L'agent saisit les critères de recherche. ➤ Système effectue la recherche et affiche le résultat. ➤ L'agent sélectionne les comptes à supprimer. ➤ Système demande la confirmation. ➤ Système confirme la suppression.
Scénario alternatif	aucun

Tableau 3.4 : Fiche de description du cas d'utilisation : supprimer compte étudiant

4.3 Etudiant

4.3.1 Créer compte d'étudiant

Cas d'utilisation	Créer compte d'étudiant.
acteur	Etudiant, Agent pédagogie.
But	Permettre à l'étudiant de créer son propre compte pour s'inscrire.
Pré conditions	Ouverture de la page web de l'inscription.
Post conditions	Un nouveau compte est créé.
Scénario nominal	<ul style="list-style-type: none"> ➤ L'étudiant demande de créer un nouveau compte par l'ouverture de la page d'inscription. ➤ Le système affiche les formulaires de création de comptes. ➤ L'étudiant remplit les formulaires et valide. ➤ Le système vérifie le formulaire d'information et confirme la création du compte.
Scénario alternatif	<ul style="list-style-type: none"> ➤ Le compte à créer existe déjà : <ul style="list-style-type: none"> ▪ Le système affiche un message d'erreur. ➤ Les formulaires contiennent des champs vides ou incorrects : <ul style="list-style-type: none"> ▪ Le système propose à l'agent de remplir les champs vides.

Tableau 3.5 : Fiche de description du cas d'utilisation : Créer compte étudiant

4.3.2 Inscrire

Cas d'utilisation	Inscrire
acteur	Etudiant, Agent pédagogie
But	Permettre à l'étudiant de se s'inscrire.
Pré conditions	L'agent et l'étudiant doivent être authentifiés.
Post conditions	L'étudiant est inscrit.
Scénario nominal	<ul style="list-style-type: none"> ➤ L'étudiant demande les formulaires d'inscription. ➤ Le système affiche les formulaires. ➤ L'étudiant remplit les formulaires. ➤ L'étudiant enregistre et valide ses informations. ➤ Le système affiche les informations saisies.
Scénario alternatif	<ul style="list-style-type: none"> ➤ L'étudiant n'a pas complété son inscription. ➤ Le système affiche une page qui permet à l'étudiant de modifier ses informations.

Tableau 3.6 : Fiche de description du cas d'utilisation : inscrire

4.3.3 Remplir formulaire :

Cas d'utilisation	Remplir formulaire
acteur	Etudiant, Agent pédagogie
But	Remplir les formulaires d'inscription.
Pré conditions	<ul style="list-style-type: none"> ➤ L'étudiant doit être authentifié. ➤ Ouverture de la page.
Post conditions	Formulaires remplis.
Scénario nominal	<ul style="list-style-type: none"> ➤ L'étudiant demande les formulaires d'inscription. ➤ Le système affiche les formulaires. ➤ L'étudiant remplit les formulaires.
Scénario alternatif	Aucun.

Tableau 3.7 : Fiche de description du cas d'utilisation : remplir formulaire

4.3.4 Modifier

Cas d'utilisation	Modifier.
acteur	Etudiant, Agent pédagogie.
But	Modifier les informations incorrectes ou manquantes.
Pré conditions	<ul style="list-style-type: none"> ➤ Ouverture de la page web de l'inscription. ➤ L'étudiant doit être authentifié
Post conditions	Les informations sont modifiées.
Scénario nominal	<ul style="list-style-type: none"> ➤ L'étudiant demande la page. ➤ Le système affiche la page demandée. ➤ L'étudiant modifie les informations incorrectes. ➤ L'étudiant confirme ses modifications.
Scénario alternatif	Aucun

Tableau 3.8 : Fiche de description du cas d'utilisation : modifie

4.3.5 Enregistrer

Cas d'utilisation	Enregistrer.
Acteur	Etudiant, Agent pédagogie.
But	Enregistrer les informations d'inscriptions.
Pré conditions	<ul style="list-style-type: none"> ➤ Ouverture de la page web de l'inscription. ➤ L'étudiant doit être authentifié
Post conditions	Les informations sont enregistrées.
Scénario nominal	<ul style="list-style-type: none"> ➤ L'étudiant envoie les informations saisies. ➤ Le système confirme l'inscription.
Scénario alternatif	<ul style="list-style-type: none"> ➤ Les informations sont incomplètes. ➤ Le système propose à l'étudiant la correction des erreurs. ➤ L'étudiant corrige les erreurs. ➤ L'étudiant enregistre ses modifications.

Tableau 3.9 : Fiche de description du cas d'utilisation : Enregistrer

4.3.6 Valider

Cas d'utilisation	Valider.
Acteur	Etudiant, Agent pédagogie.
But	Valider les informations d'inscriptions.
Pré conditions	<ul style="list-style-type: none"> ➤ Ouverture de la page Web de l'inscription. ➤ L'étudiant doit être authentifié
Post conditions	Les informations sont validées.
Scénario nominal	<ul style="list-style-type: none"> ➤ L'étudiant demande les informations saisis. ➤ Le système affiche les informations demandées. ➤ L'étudiant enregistre et valide les informations.
Scénario alternatif	<ul style="list-style-type: none"> ➤ Les informations sont incomplètes. ➤ Le système propose à l'étudiant la correction des erreurs. ➤ L'étudiant corrige les erreurs. ➤ L'étudiant valide les modifications.

Tableau 3.10 : Fiche de description du cas d'utilisation : valider

5. Les diagrammes de séquence

5.1 Administrateur

5.1.1 S'authentifier

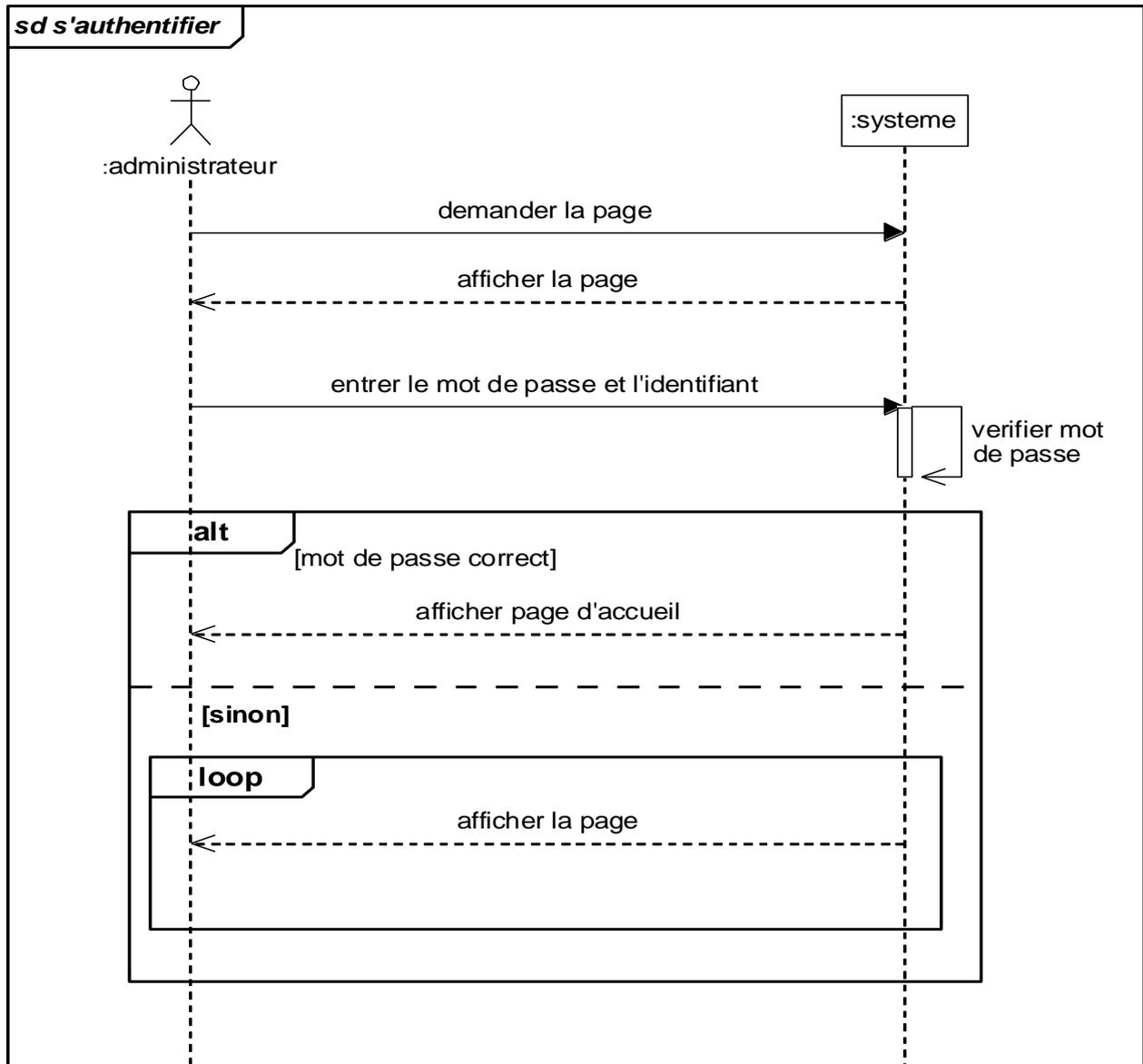


Figure 3.3 : Diagramme de séquence du cas d'utilisation: s'authentifier

5.1.2 Créer compte pédagogie

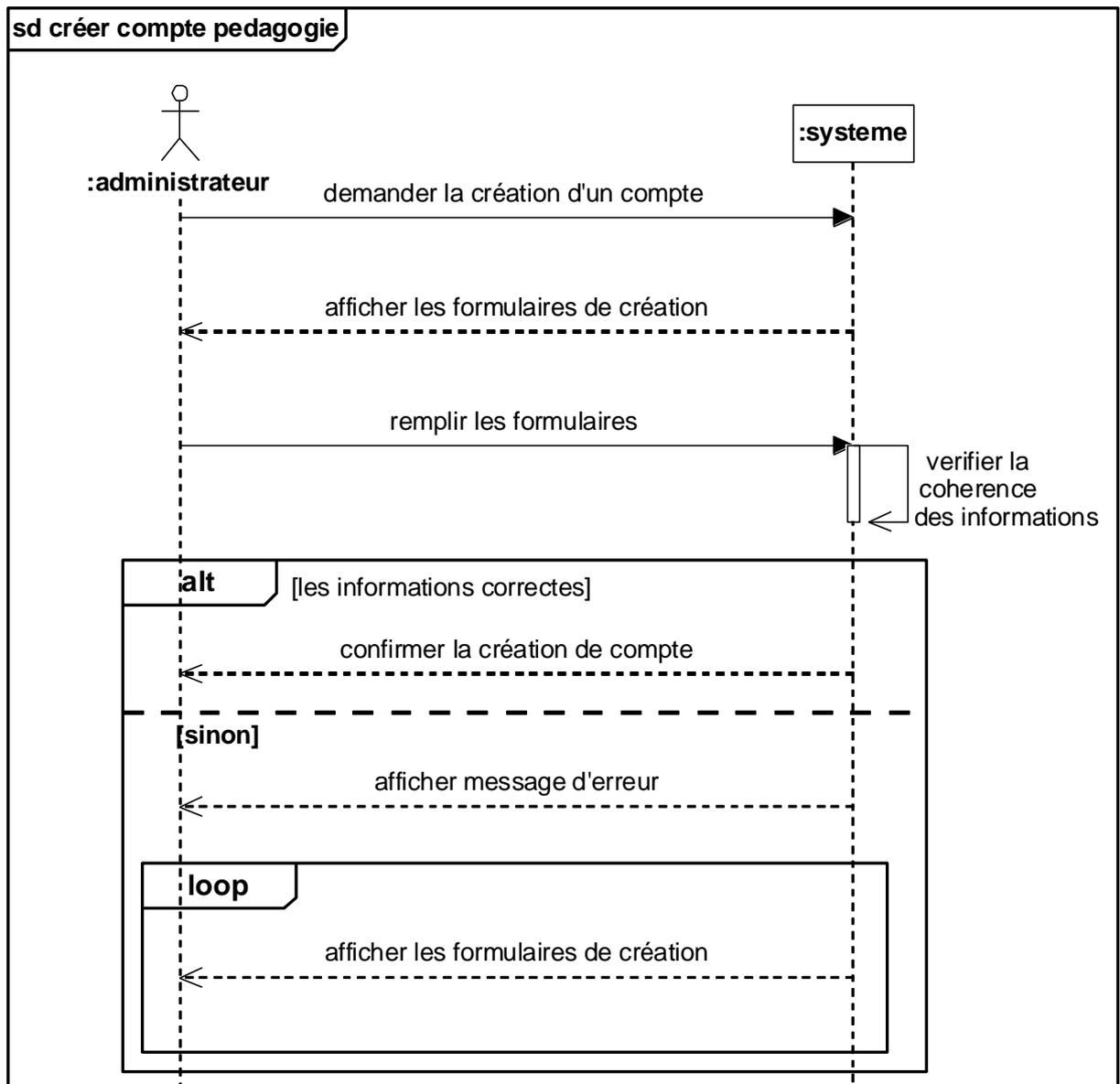


Figure 3.4: Diagramme de séquence du cas d'utilisation: créer compte pédagogie

5.2 Agent pédagogie
5.2.1 Rechercher

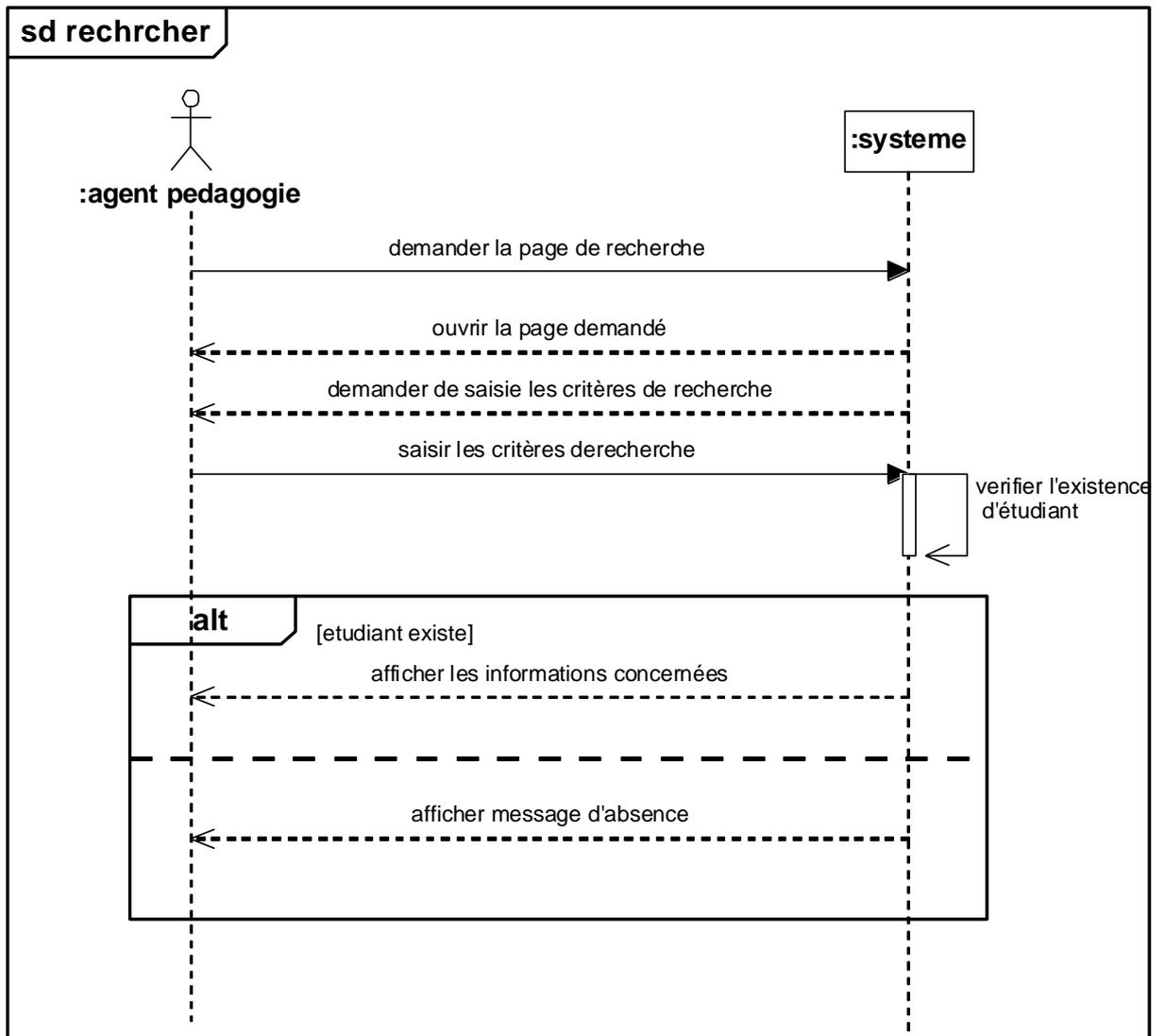


Figure 3.5 : Diagramme de séquence du cas d'utilisation: Rechercher

5.2.2 Supprimer compte étudiant :

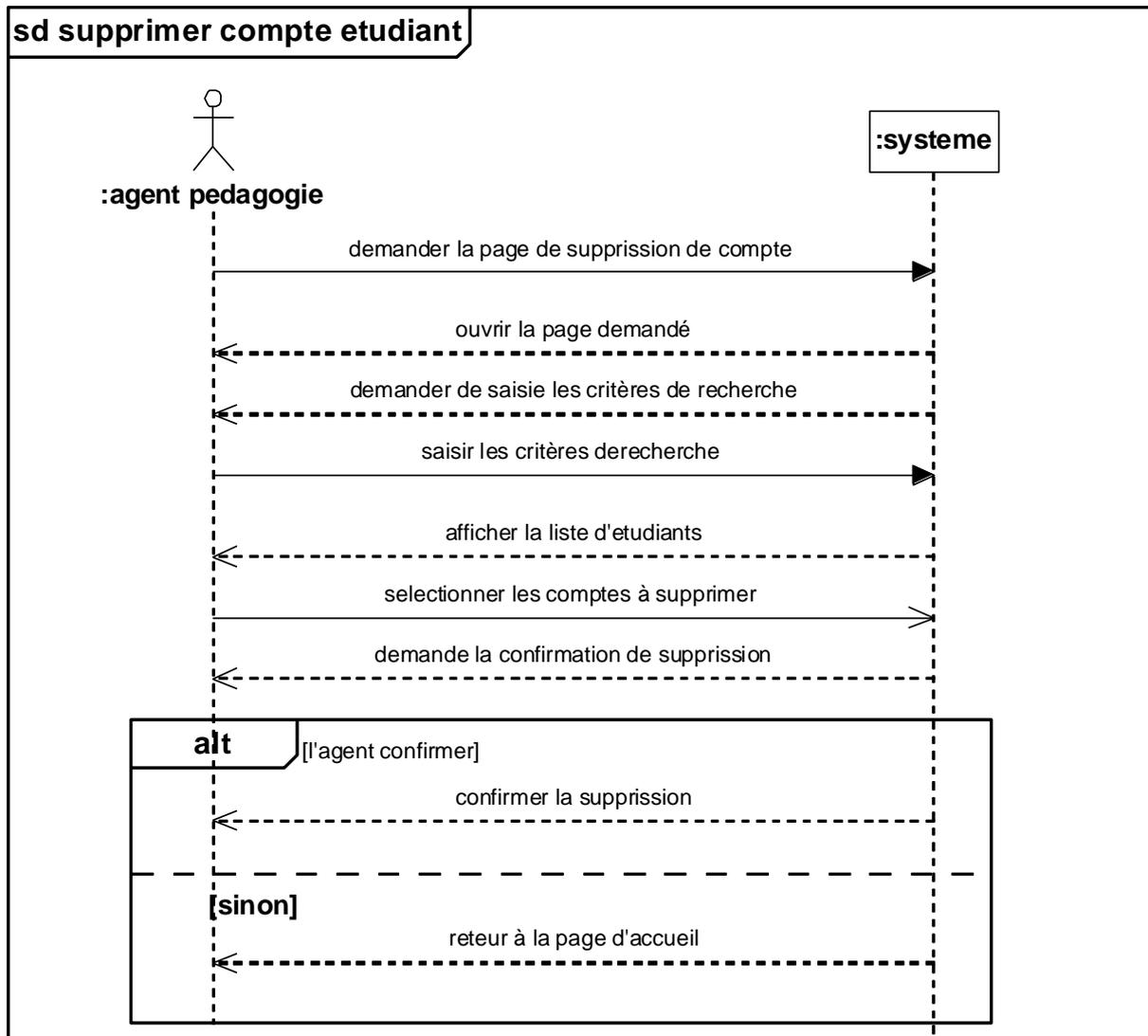


Figure 3.6 : Diagramme de séquence du cas d'utilisation: Supprimer compte étudiant

5.3 Etudiant :

5.3.1 Créer compte étudiant

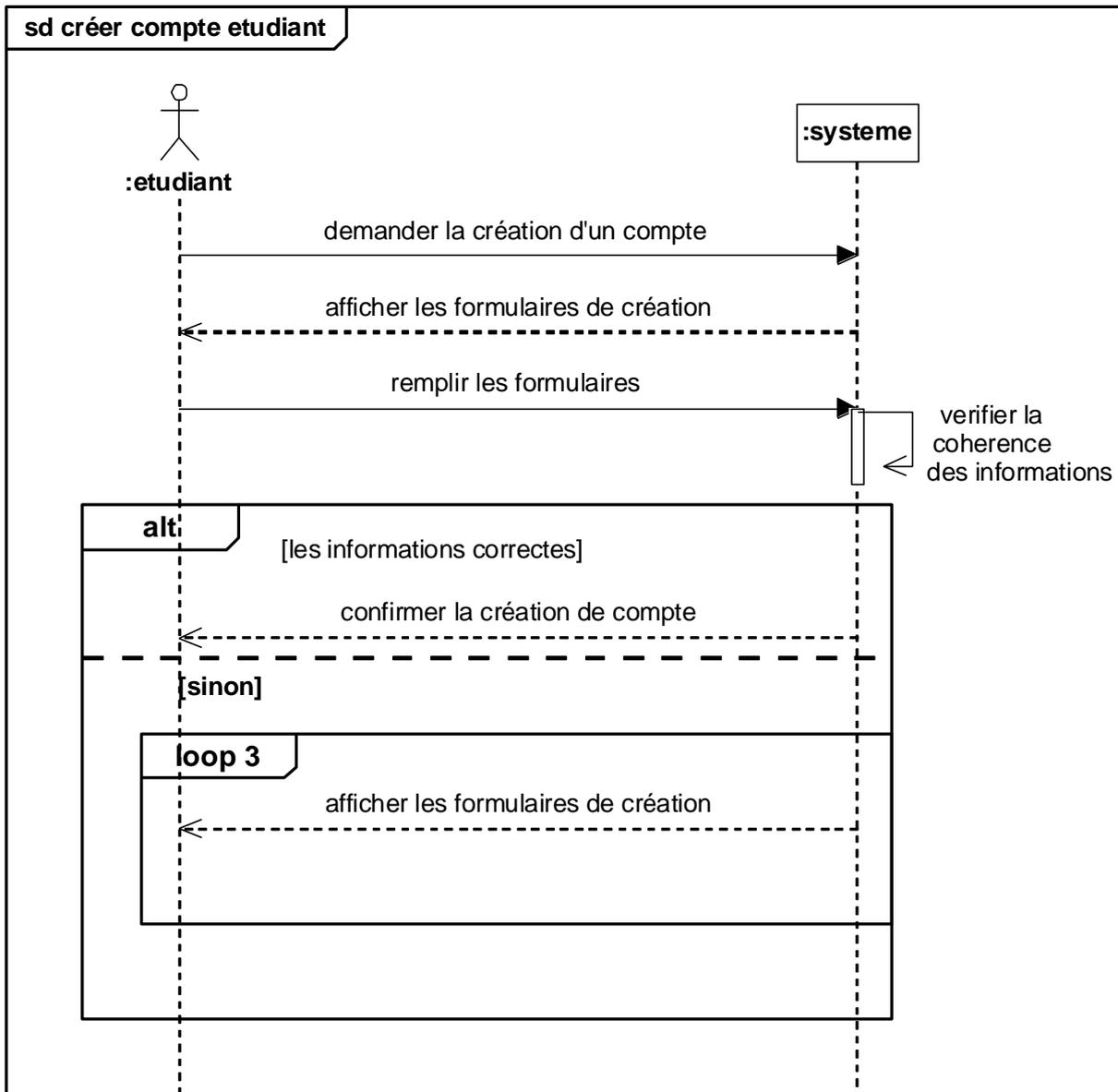


Figure 3.7 : Diagramme de séquence du cas d'utilisation: Créer compte étudiant

5.3.2 S'inscrire

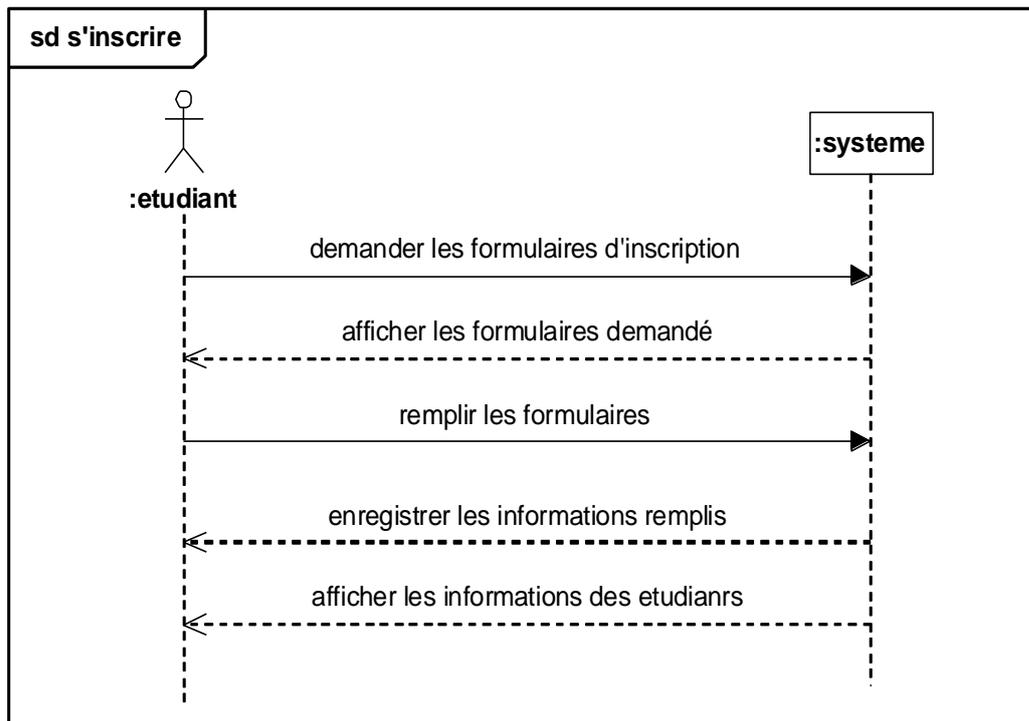


Figure 3.8: Diagramme de séquence du cas d'utilisation: S'inscrire

5.3.3 Remplir les formulaires :

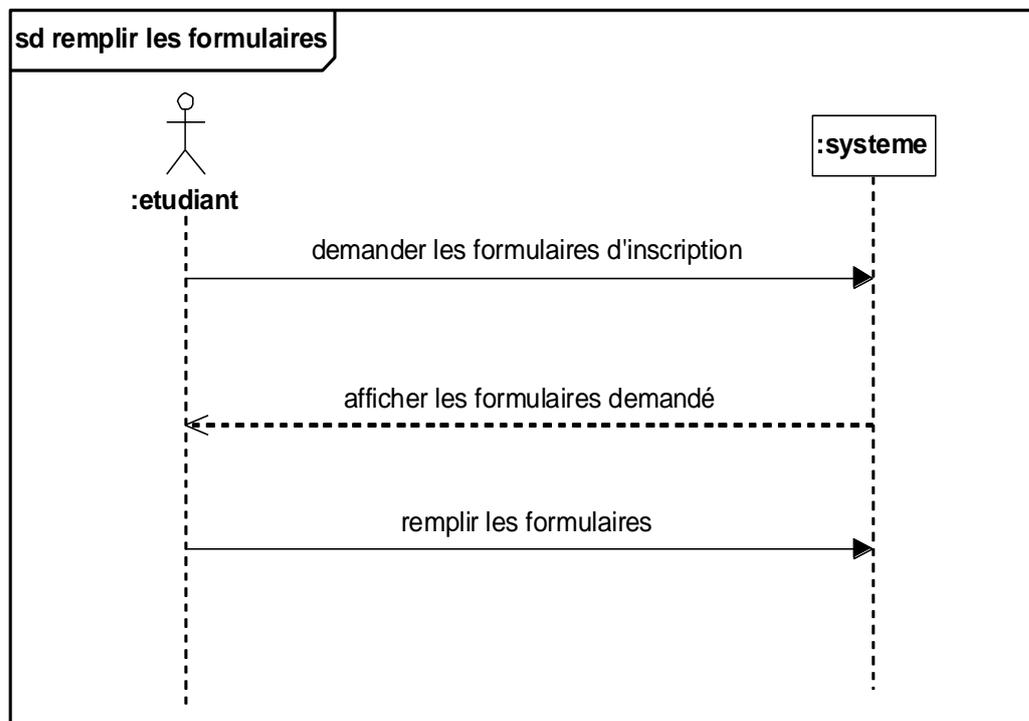


Figure 3.9: Diagramme de séquence du cas d'utilisation: Remplir formulaire

5.3.4 Modifier

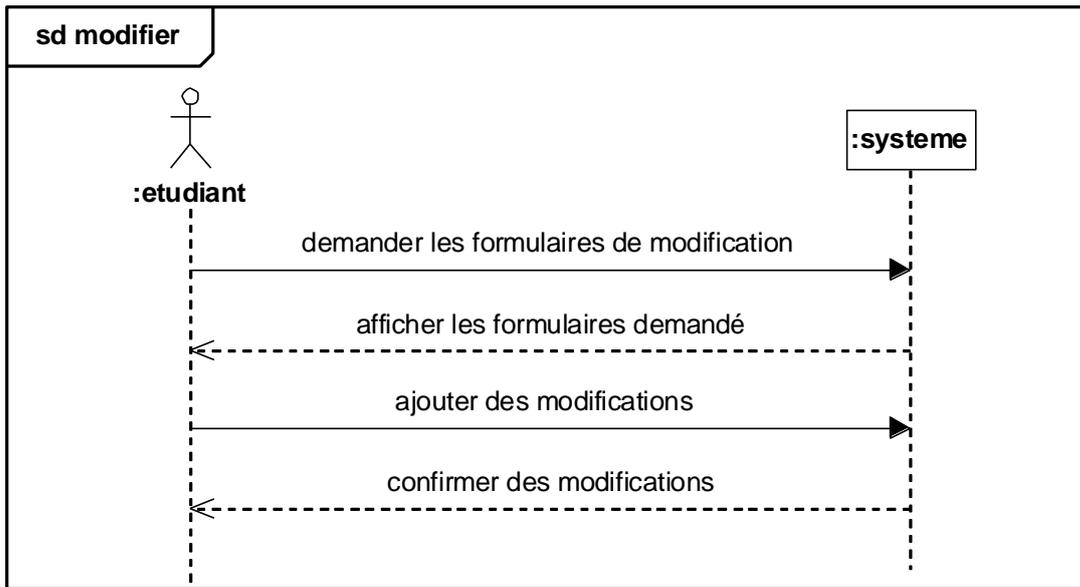


Figure 3.10 : Diagramme de séquence du cas d'utilisation: Modifier

5.3.5 Enregistrer

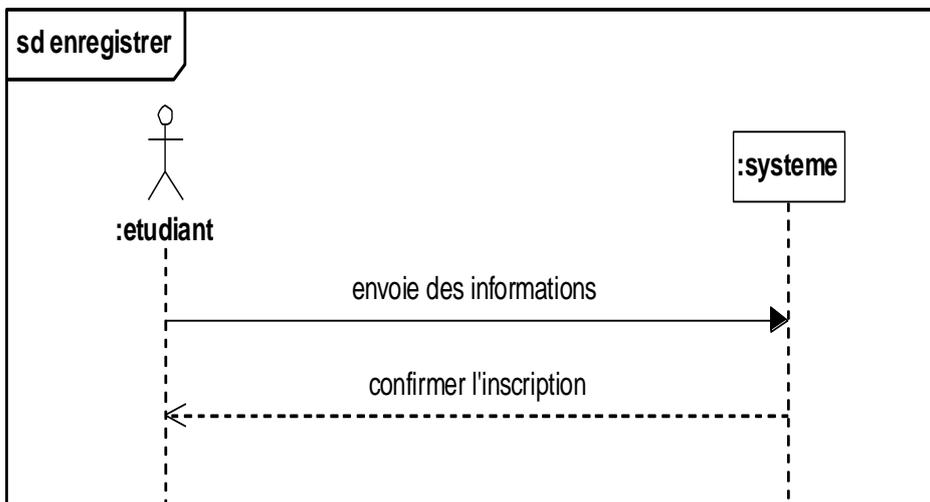


Figure 3.11: Diagramme de séquence du cas d'utilisation: Enregistrer

5.3.6 Valider

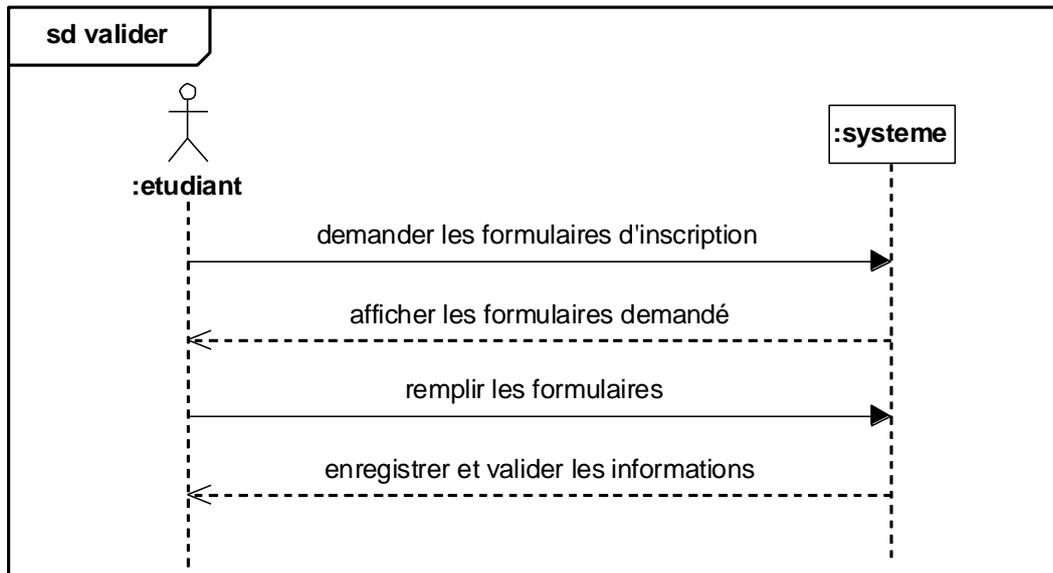


Figure 3.12 : Diagramme de séquence du cas d'utilisation: Valider

6. Les Diagrammes d'activités

6.1 Administrateur :

6.1.1 Authentifier

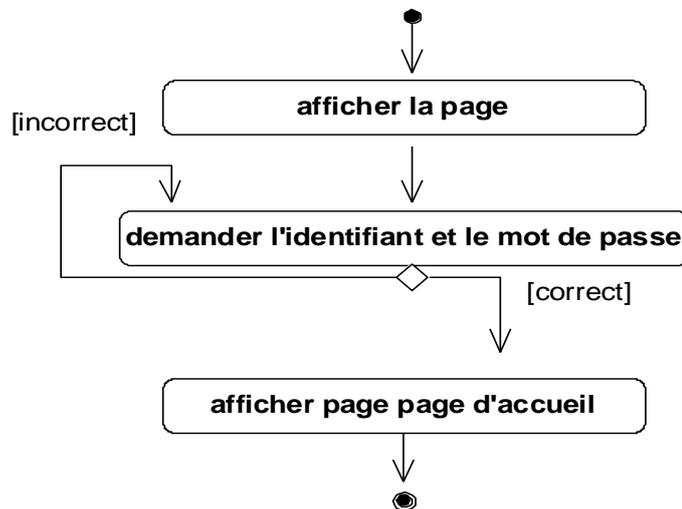


Figure 3.13: Diagramme d'activité du cas d'utilisation: Authentifier

6.1.2 créer compte pédagogie

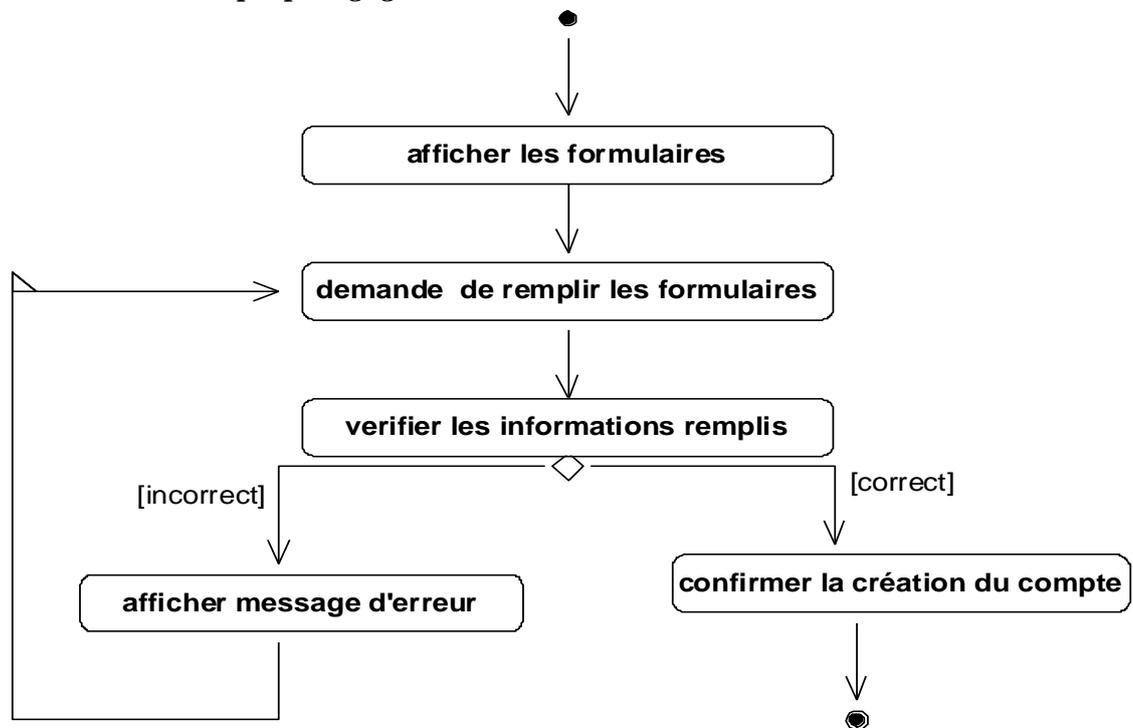


Figure 3.14 : Diagramme d'activité du cas d'utilisation: Créer compte pédagogie

6.2 Agent pédagogie

6.2.1 rechercher

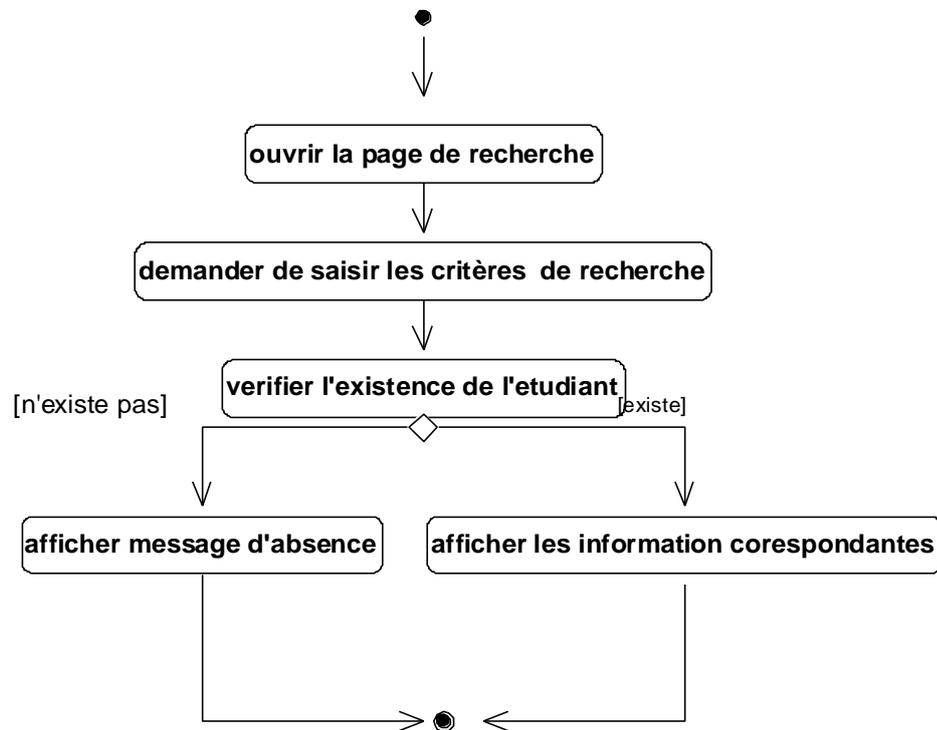


Figure 3.15 : diagramme d'activité du cas d'utilisation: rechercher

6.2.2 supprimer compte étudiant

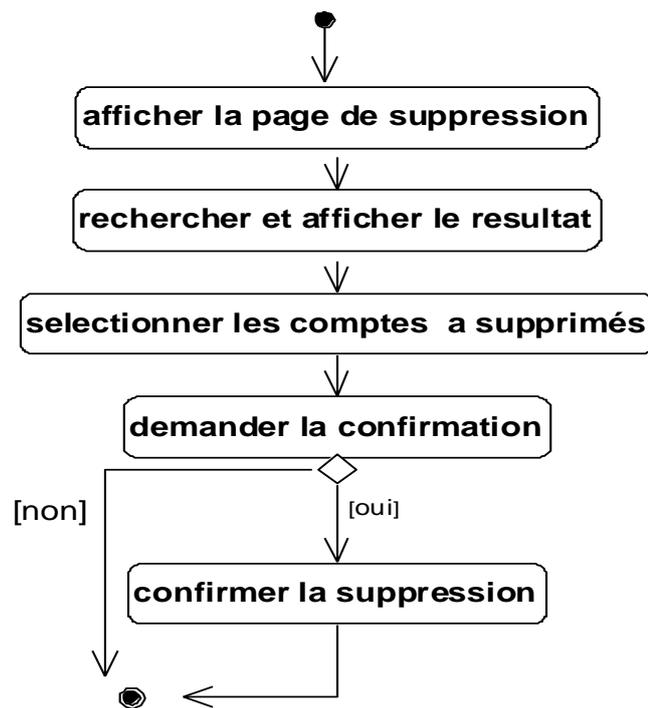


Figure 3.16 : diagramme d'activité du cas d'utilisation: supprimer compte étudiant

6.3 Etudiant

6.3.1 créer compte étudiant

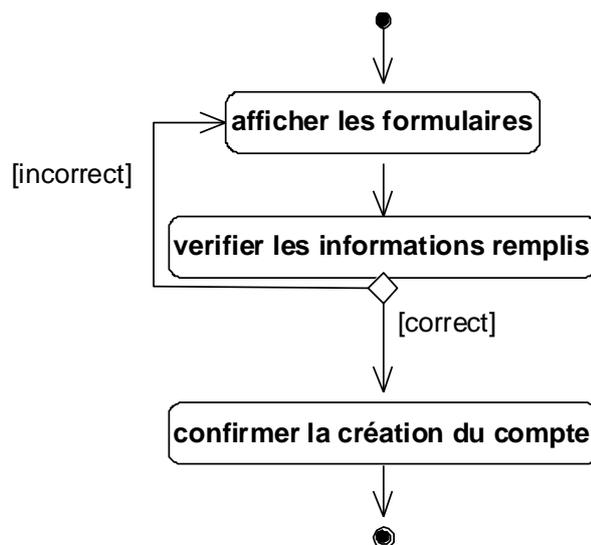


Figure 3.17: diagramme d'activité du cas d'utilisation: créer compte étudiant

6.3.2 s'inscrire

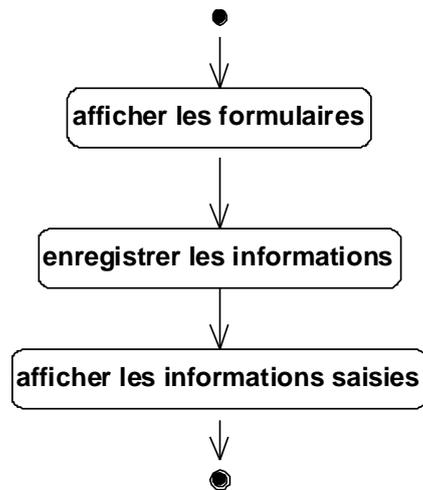


Figure 3.18 : diagramme d'activité du cas d'utilisation: s'inscrire

6.3.3 remplir les formulaires :

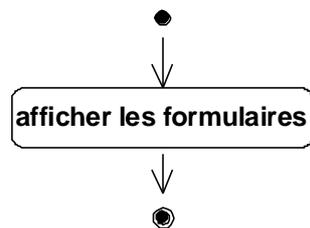


Figure 3.19: diagramme d'activité du cas d'utilisation: remplir les formulaires.

6.3.4 modifier

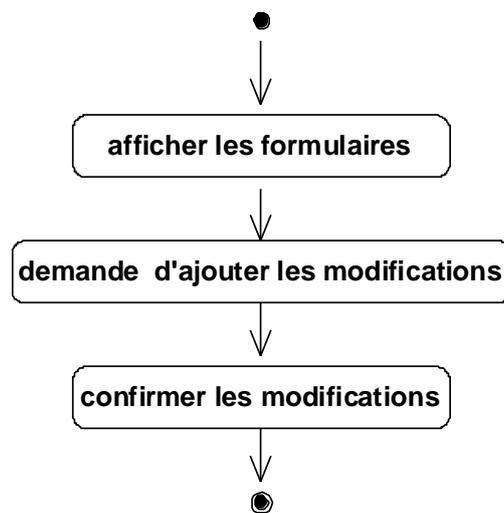


Figure 3.20: diagramme d'activité du cas d'utilisation:modifier

6.3.5 enregistrer

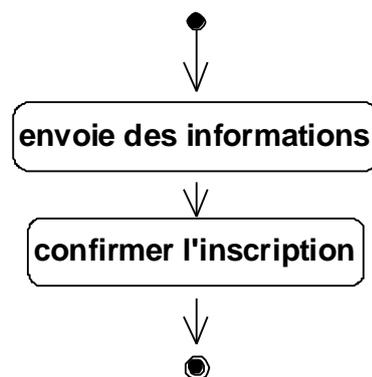


Figure 3.21 : diagramme d'activité du cas d'utilisation:enregistrer

6.3.6 valider

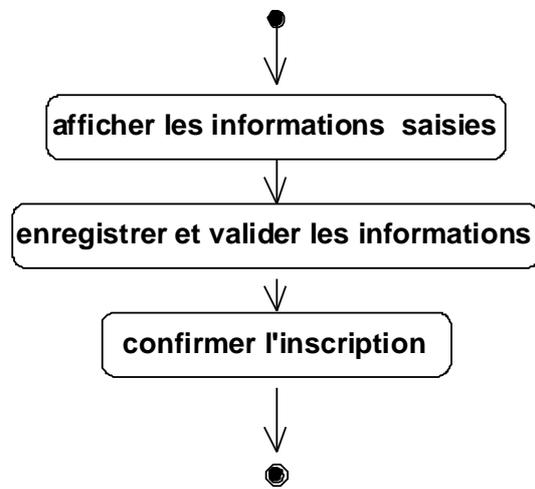


Figure 3.22 : diagramme d'activité du cas d'utilisation: valider

7. Le diagramme de classe

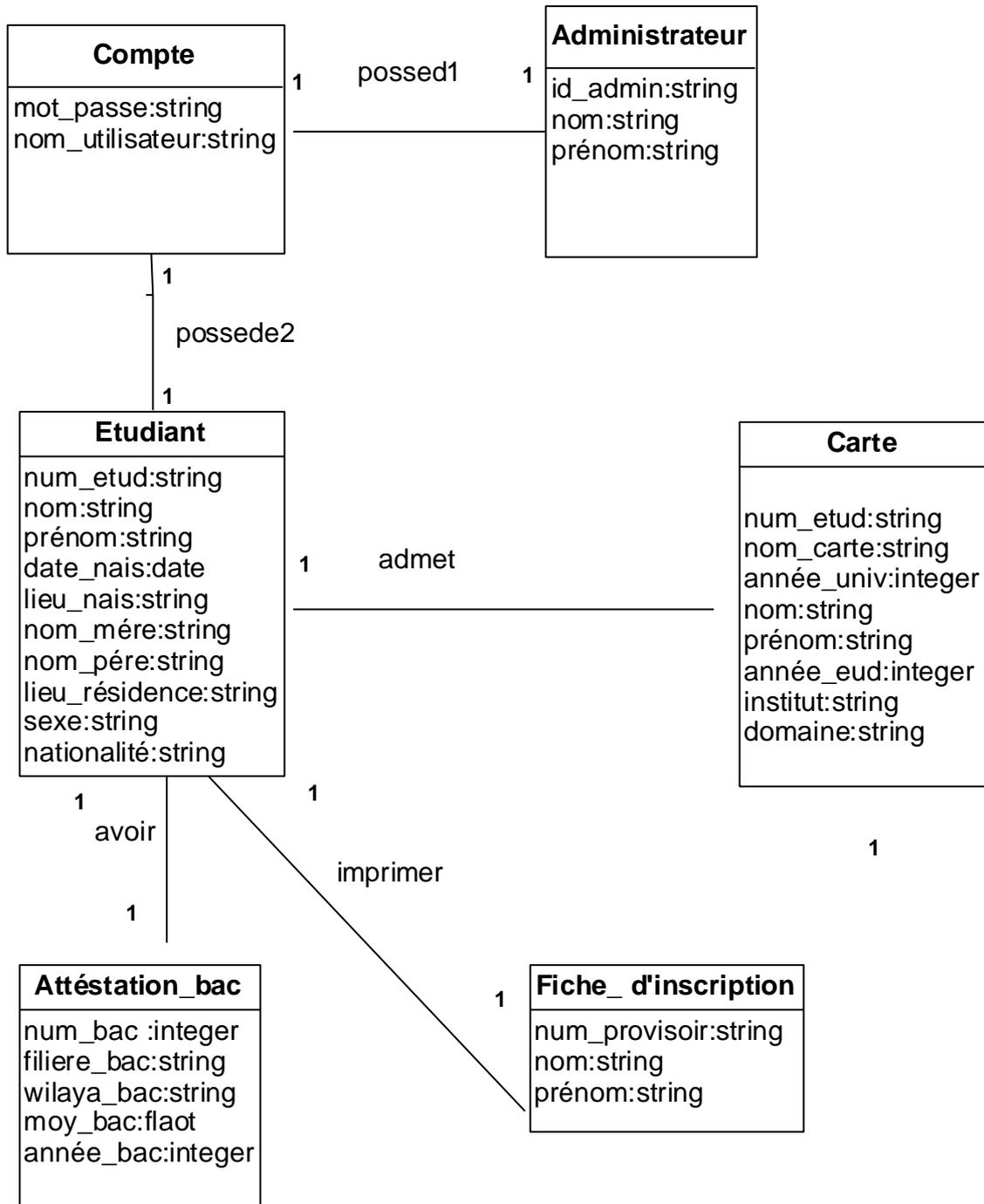


Figure 3.23 : le diagramme de classe

8. Les diagrammes de classe participants

8.1 Administrateur

8.1.1 s'authentifier

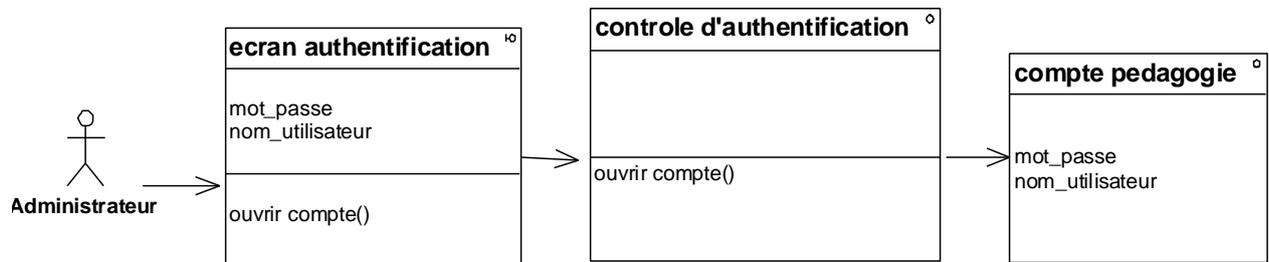


Figure 3.24 : diagramme de classe participant du cas d'utilisation: authentifier.

8.1.2 créer compte pédagogie

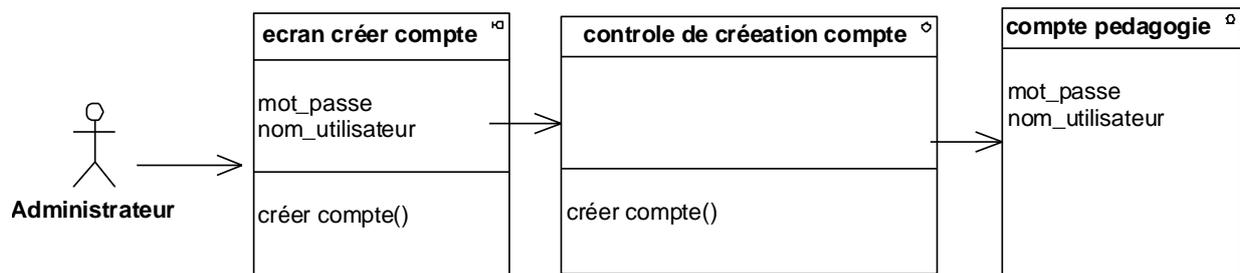


Figure 3.25: diagramme de classe participant du cas d'utilisation: créer compte pédagogique.

8.2 Agent pédagogie

8.2.1 Rechercher

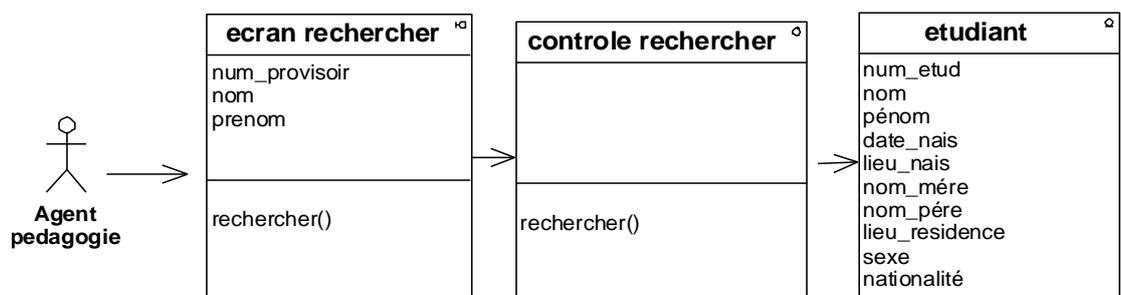


Figure 3.26 : diagramme de classe participant du cas d'utilisation: rechercher.

8.2.2 supprimer compte

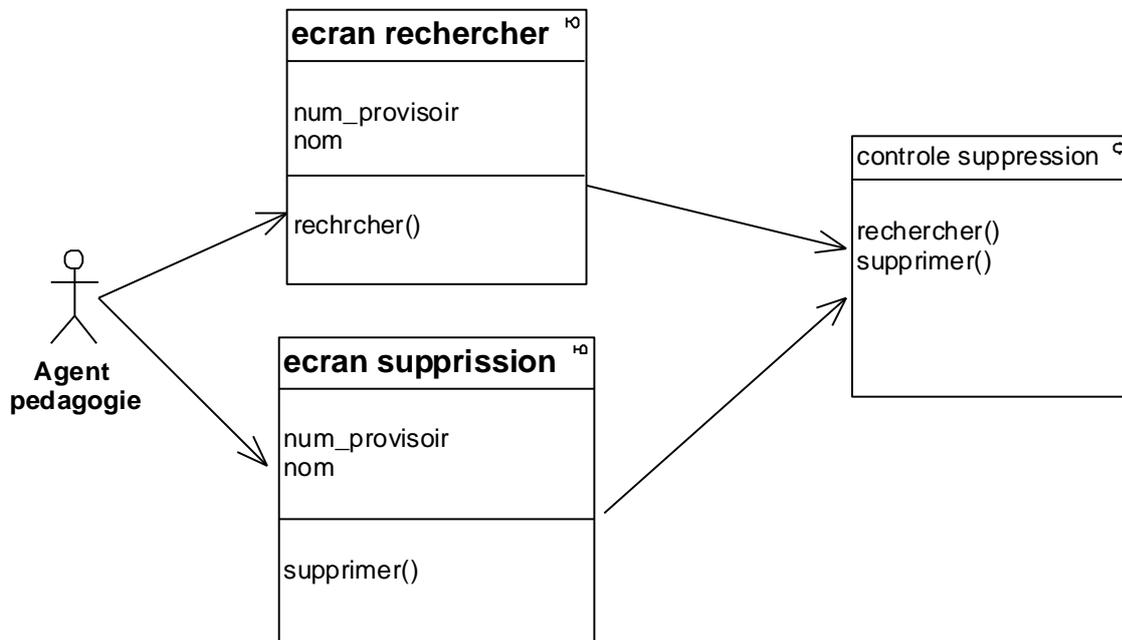


Figure 3.27 : diagramme de classe participant du cas d'utilisation: supprimer compte étudiant.

8.3 Etudiant

8.3.1 créer compte étudiant

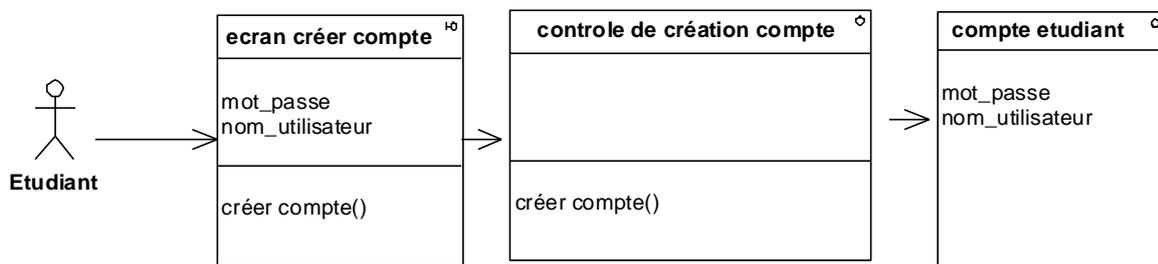


Figure 3.28 : diagramme de classe participant du cas d'utilisation: créer compte étudiant.

8.3.2 s'inscrire

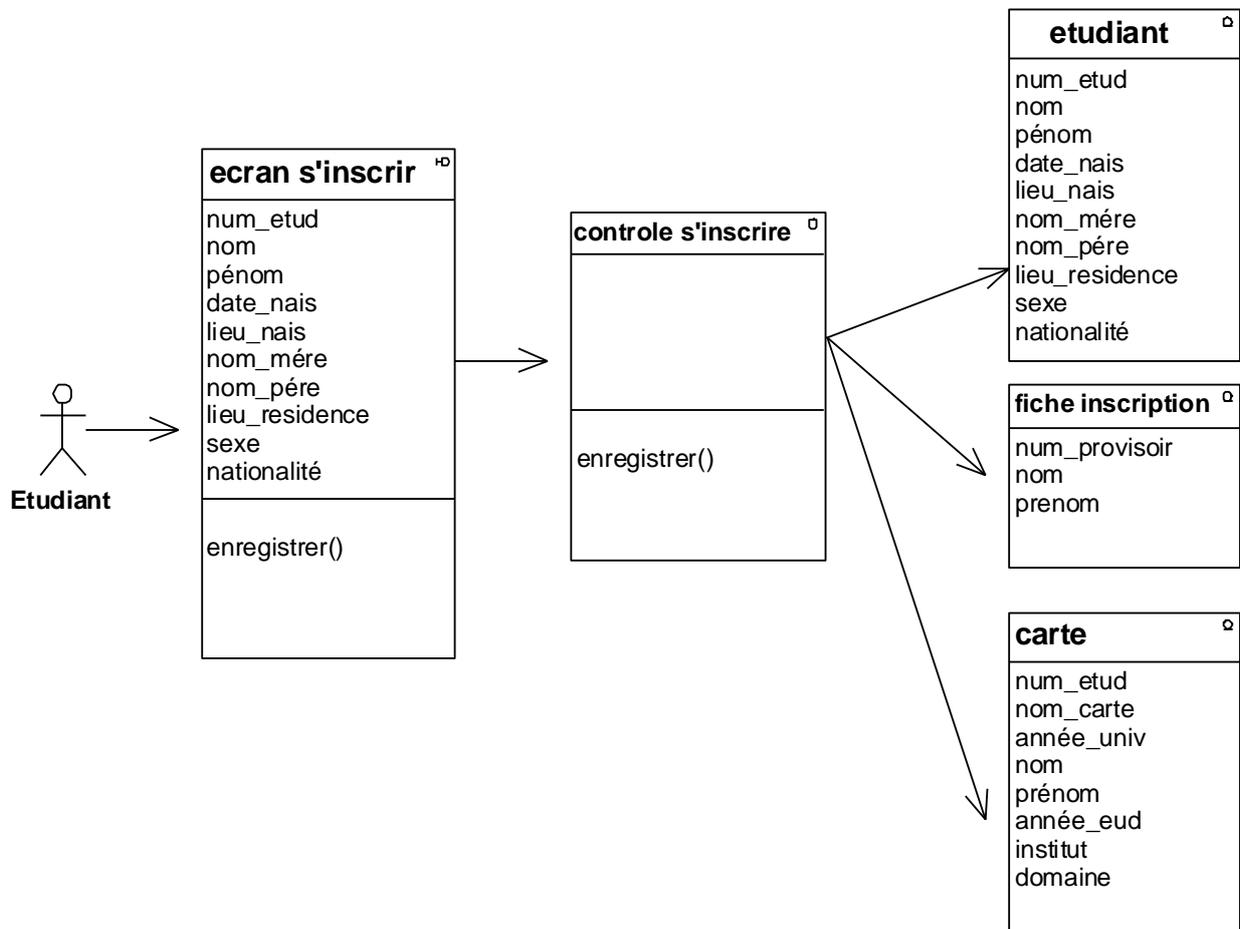


Figure 3.29 : diagramme de classe participant du cas d'utilisation: inscrire.

8.3.3 remplir les formulaires :

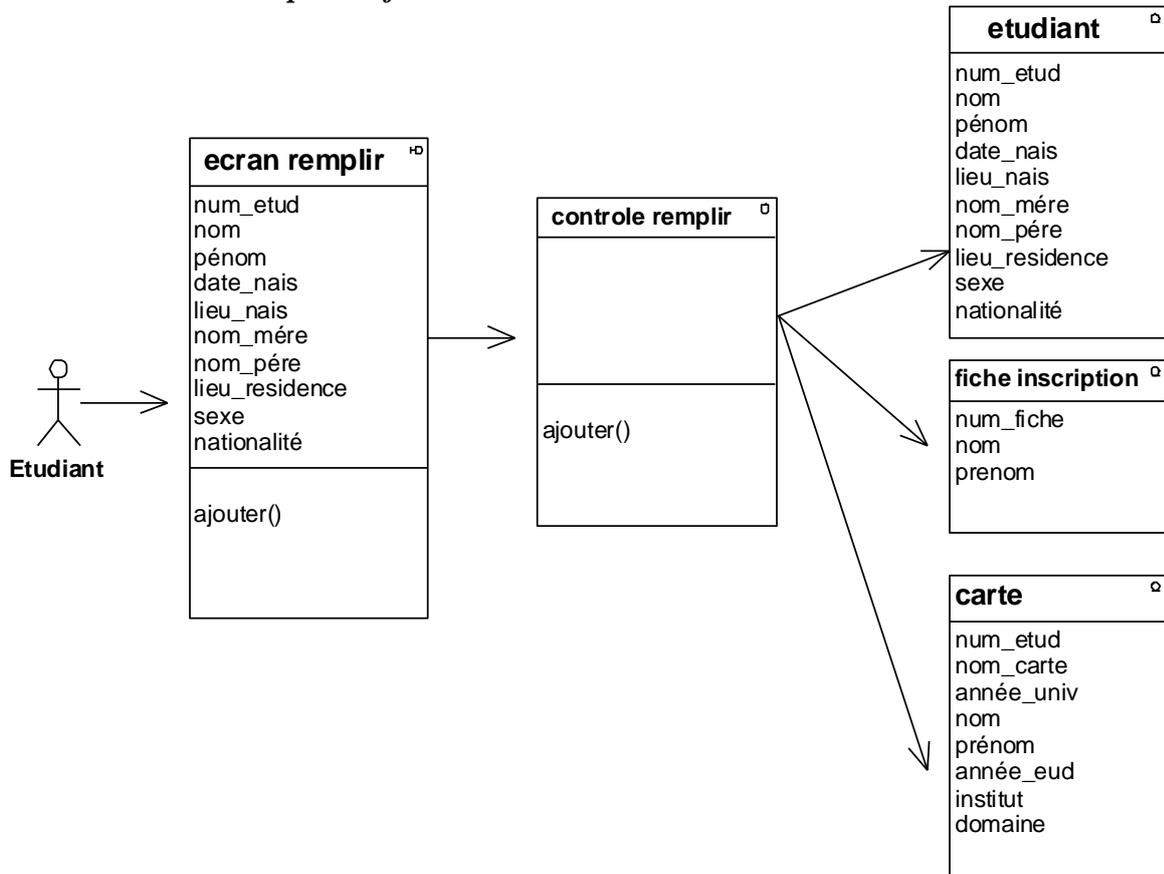


Figure 3.30 : diagramme de classe participant du cas d'utilisation: remplir formulaire.

8.3.4 modifier

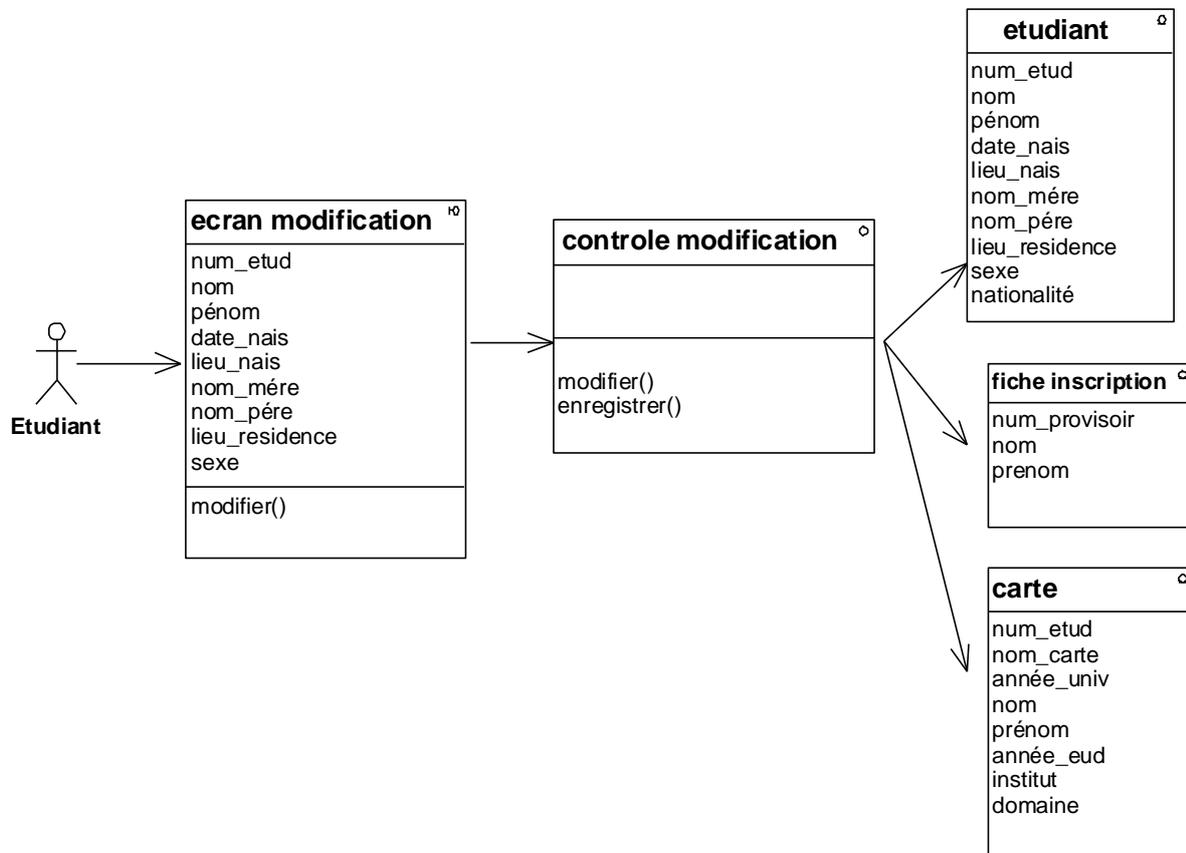


Figure 3.31 : diagramme de classe participant du cas d'utilisation: modifier.

8.3.5 enregistrer

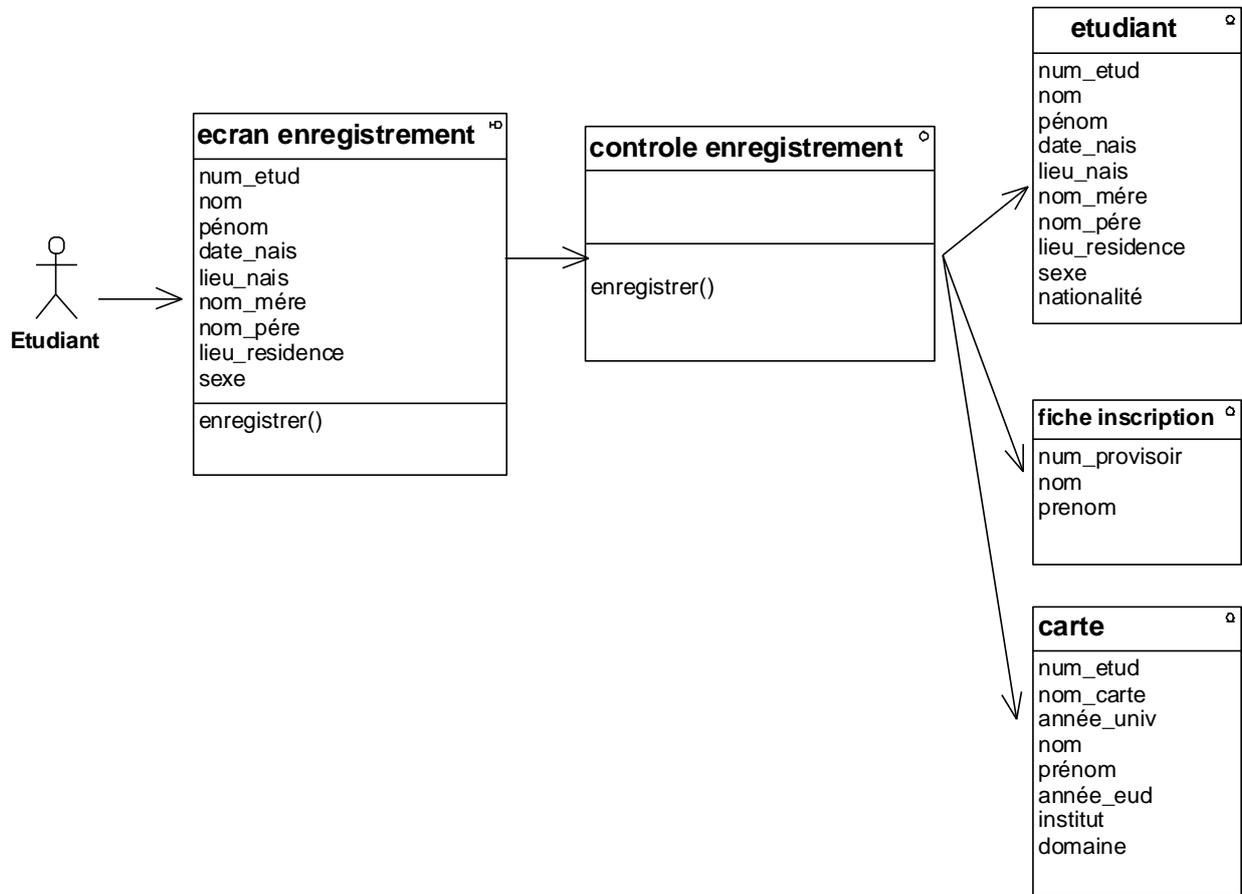


Figure 3.32 : diagramme de classe participant du cas d'utilisation: enregistrer.

8.3.6 valider

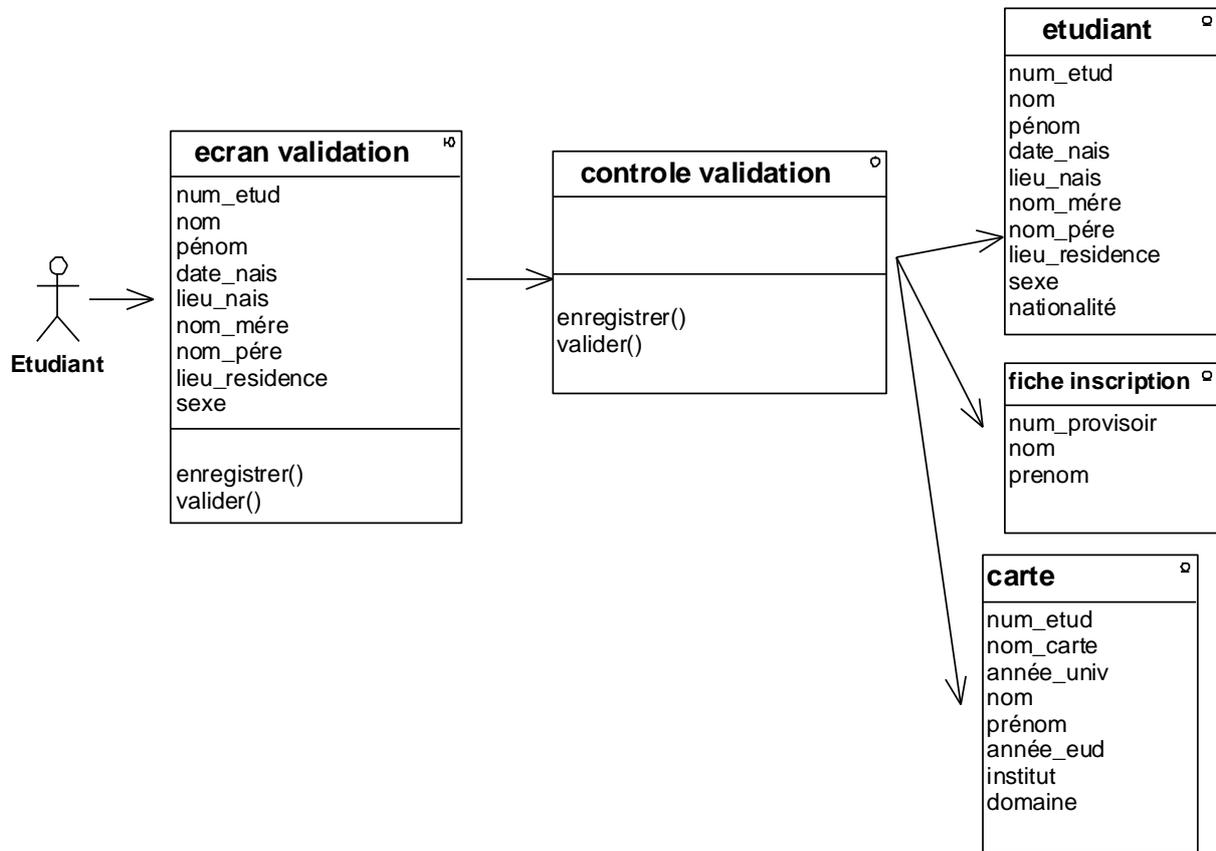


Figure 3.33 : diagramme de classe participant du cas d'utilisation: valider.

9. Les diagrammes de classe conceptions

9.1 Administrateur

9.1.1 s'authentifier

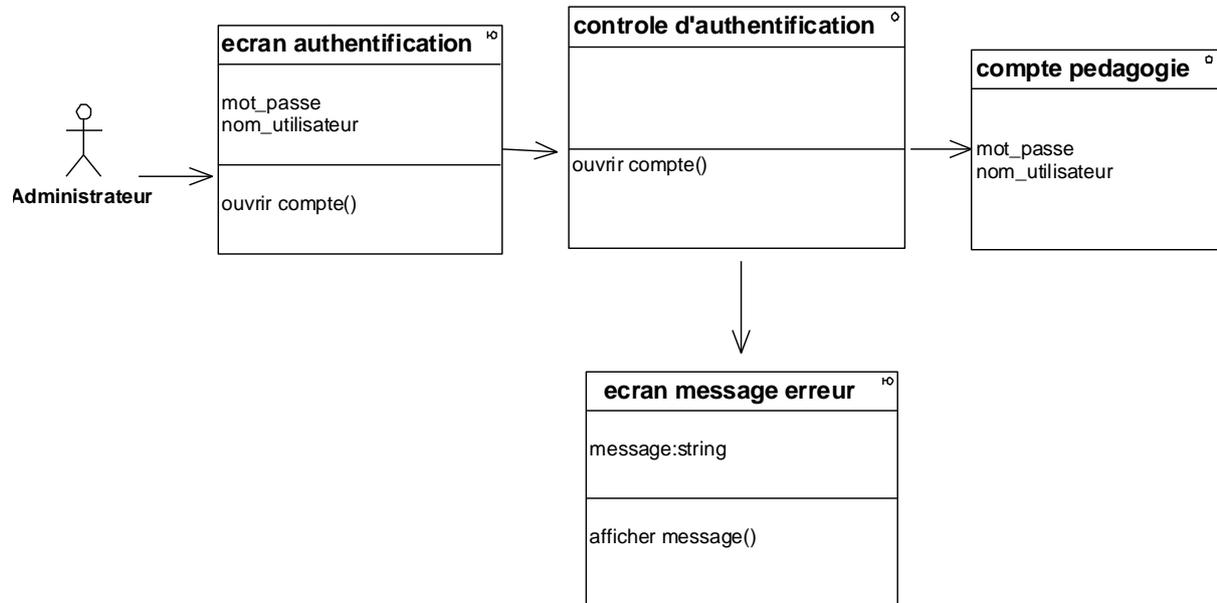


Figure 3.34 : diagramme de classe de conception du cas d'utilisation: authentifier.

9.1.2 créer compte pédagogie

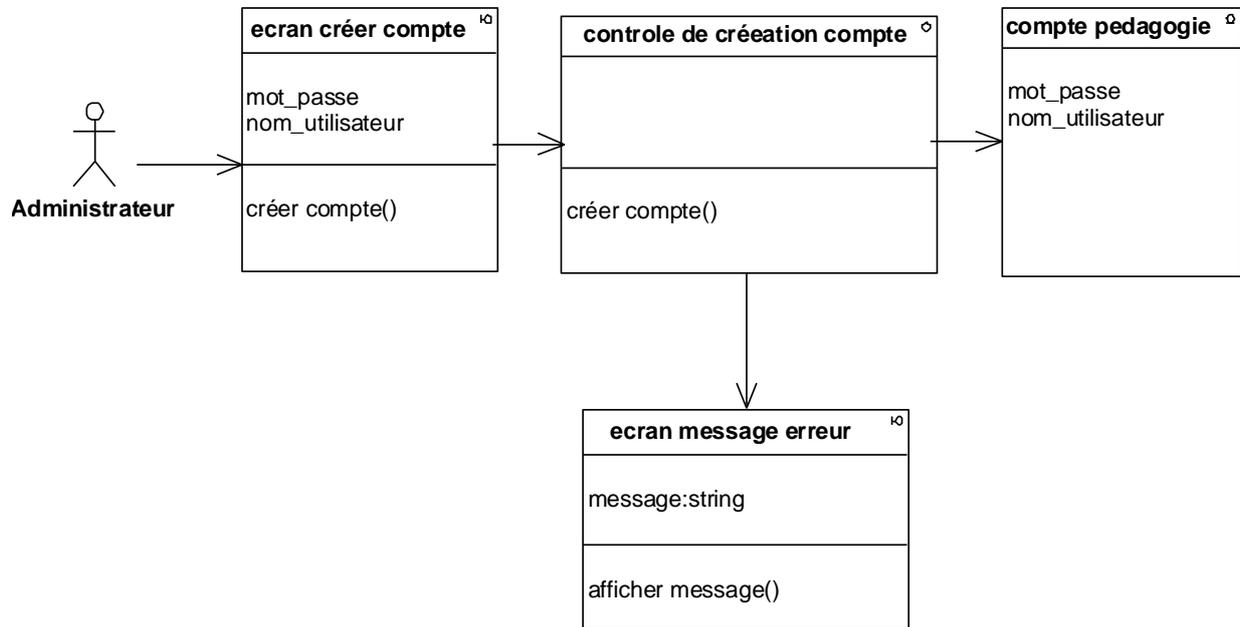


Figure 3.35 : diagramme de classe de conception du cas d'utilisation: créer compte pédagogique.

9.2 Agent pédagogie

9.2.1 Rechercher

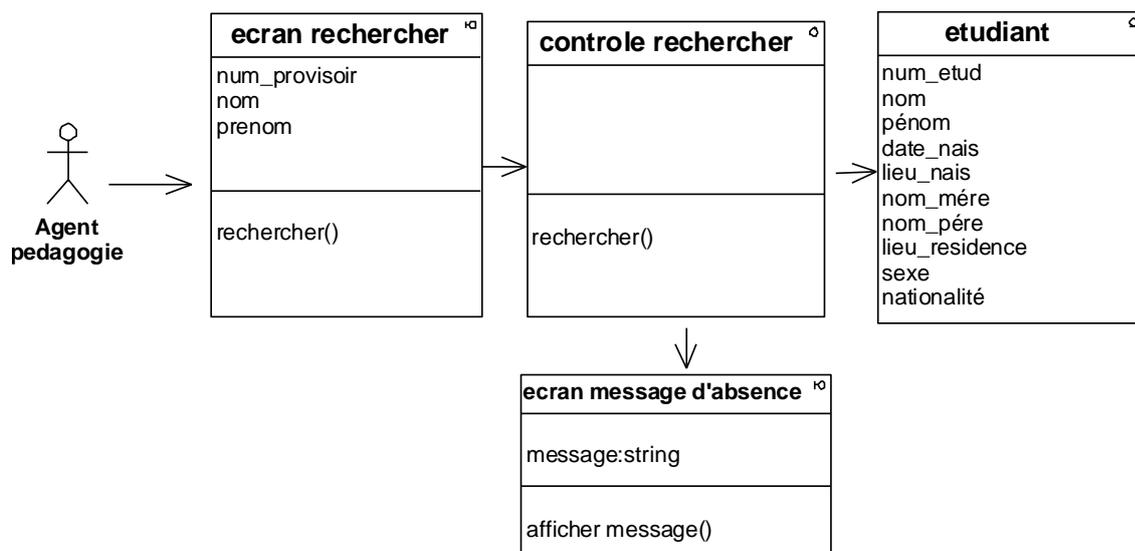


Figure 3.36 : diagramme de classe de conception du cas d'utilisation: rechercher.

9.2.2 supprimer compte

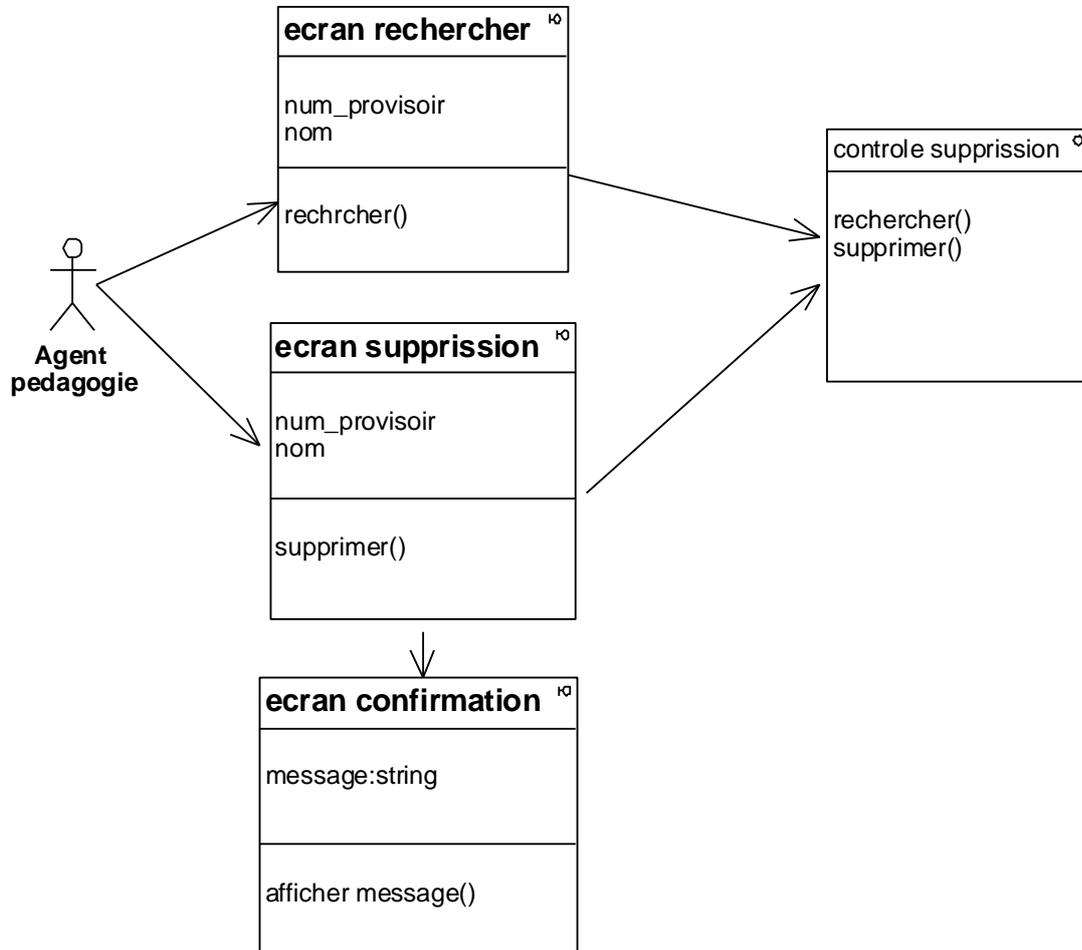


Figure 3.37 : diagramme de classe de conception du cas d'utilisation: supprimer compte étudiant.

9.3 Etudiant

9.3.1 créer compte étudiant

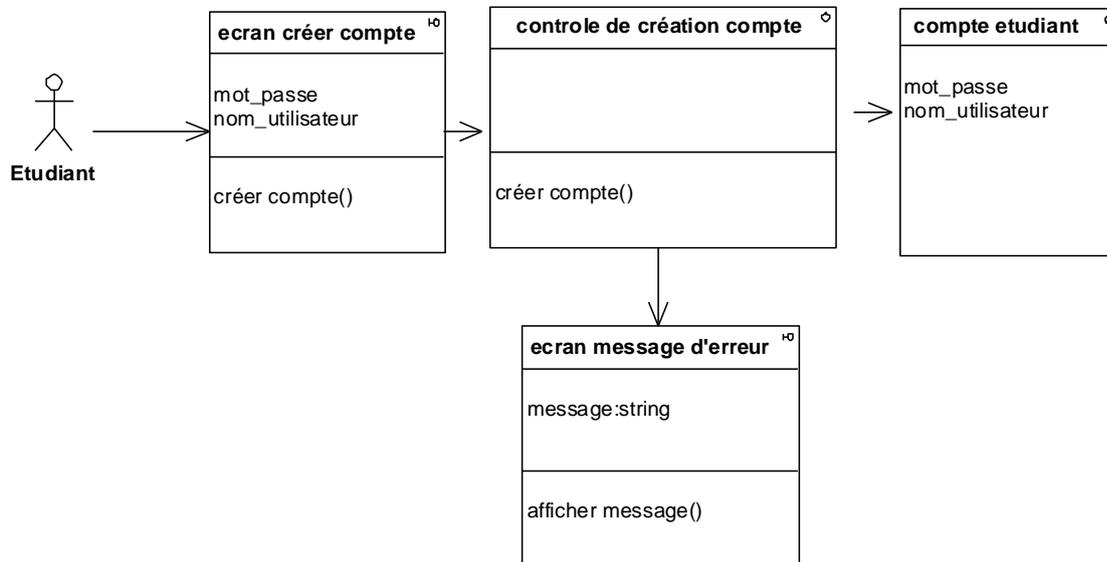


Figure 3.38 : diagramme de classe de conception du cas d'utilisation: créer compte étudiant.

9.3.2 s'inscrire

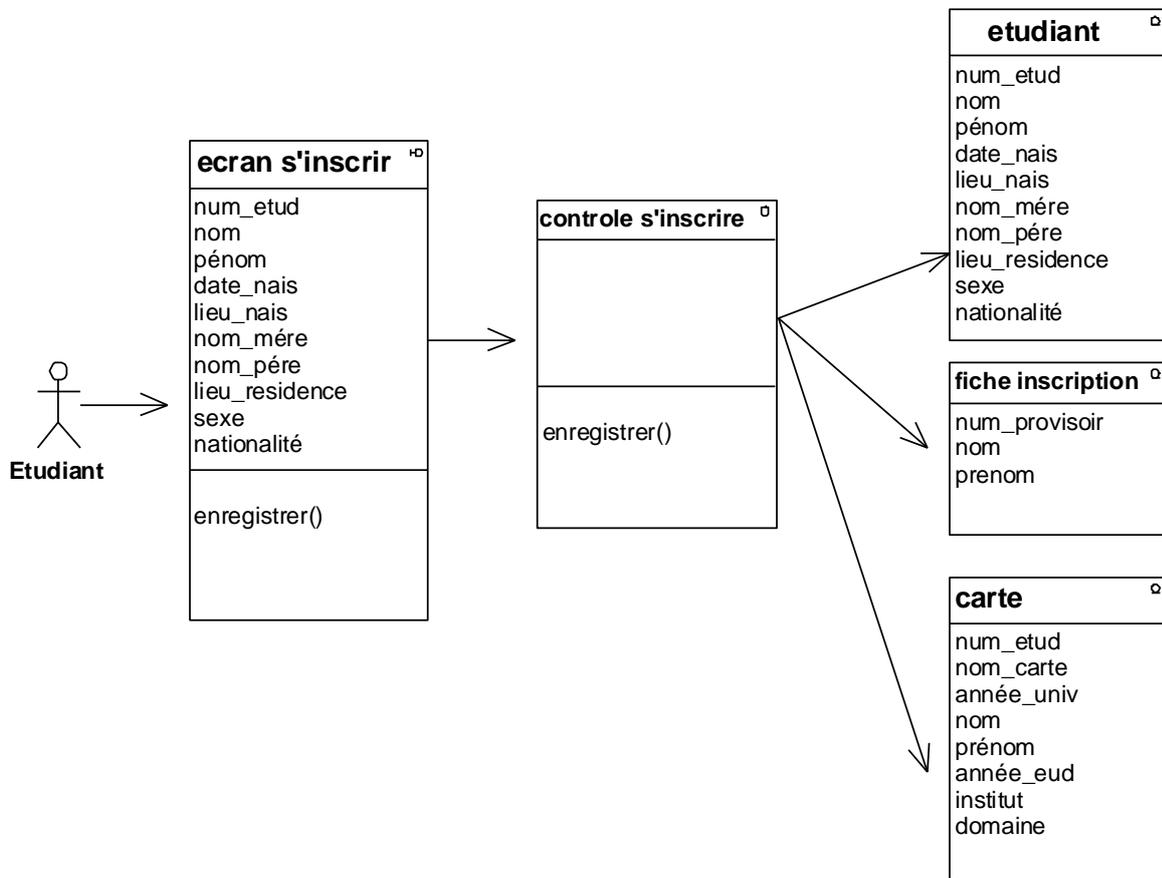


Figure 3.39 : diagramme de classe de conception du cas d'utilisation: inscrire.

9.3.3 remplir les formulaires :

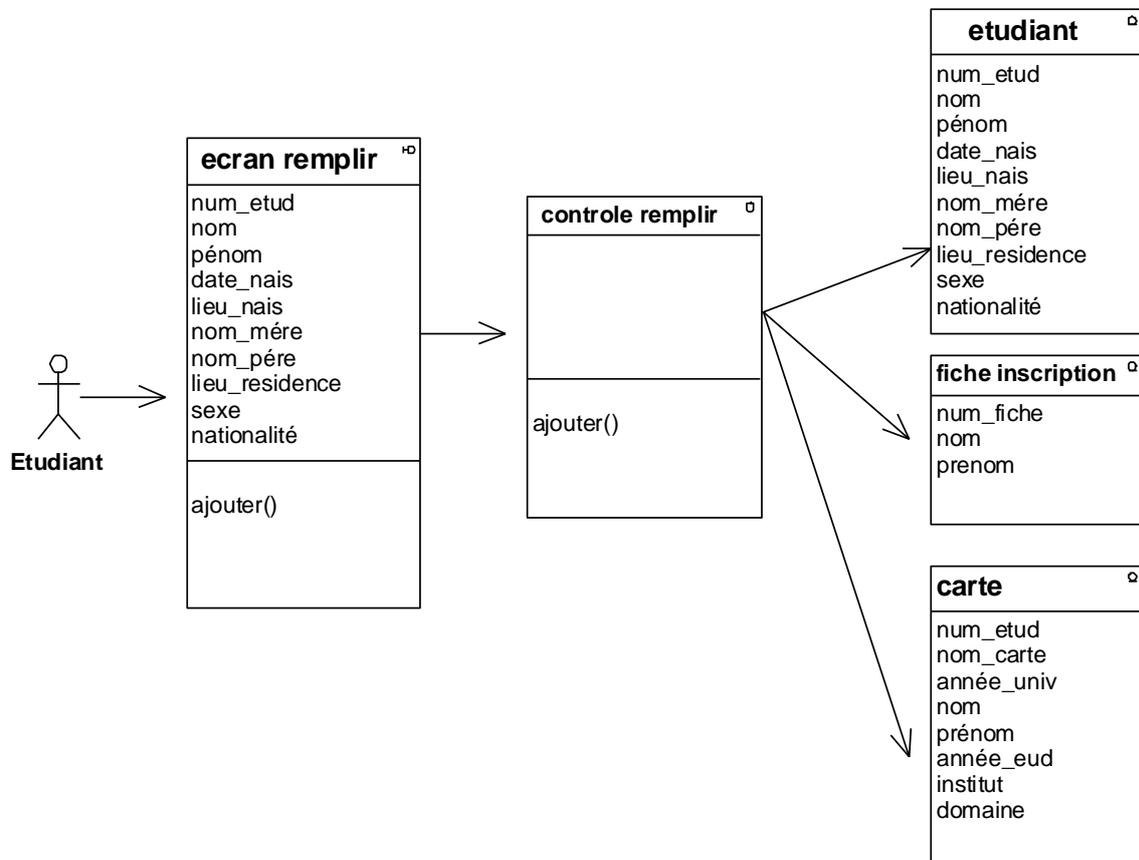


Figure 3.40 : diagramme de classe de conception du cas d'utilisation: remplir formulaire.

9.3.4 modifier

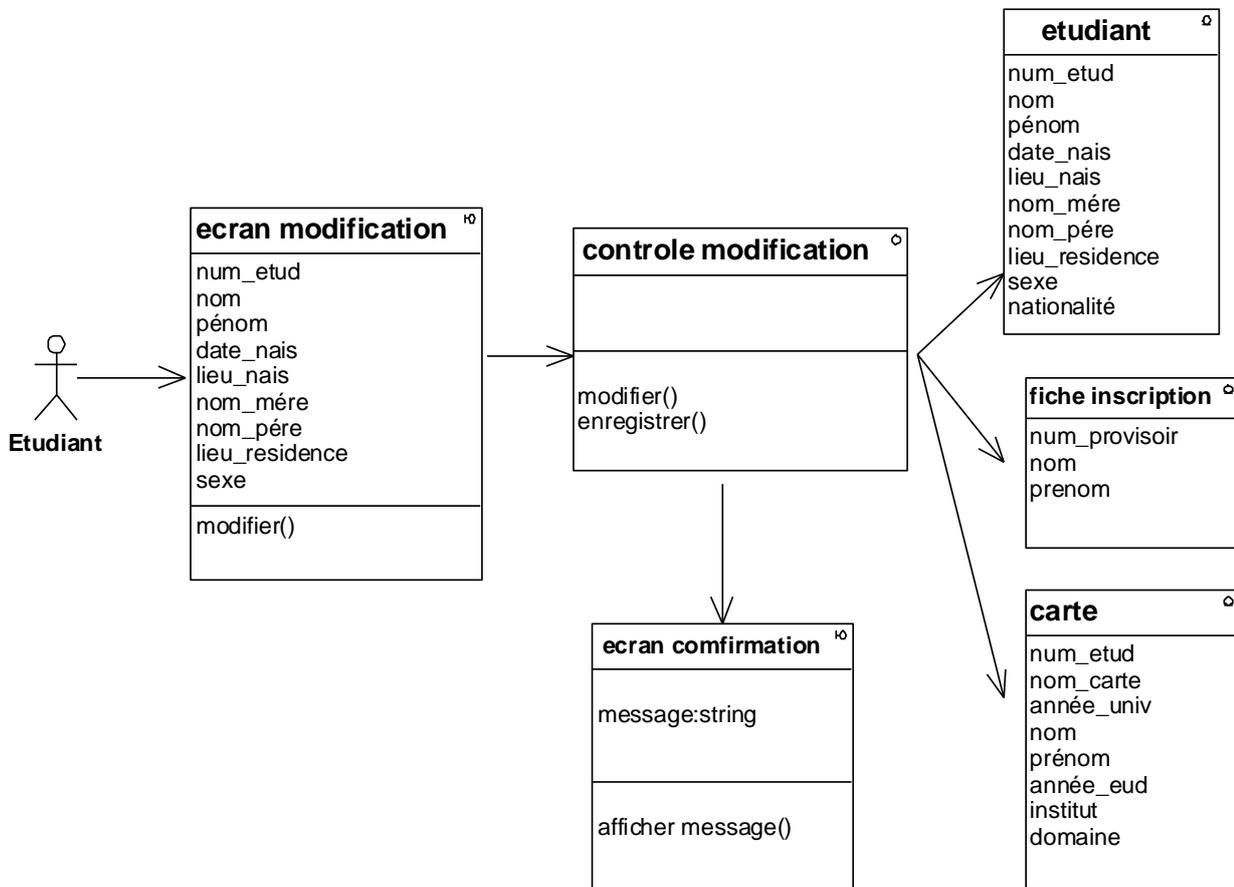


Figure 3.41 : diagramme de classe de conception du cas d'utilisation: modifier.

9.3.5 enregistrer

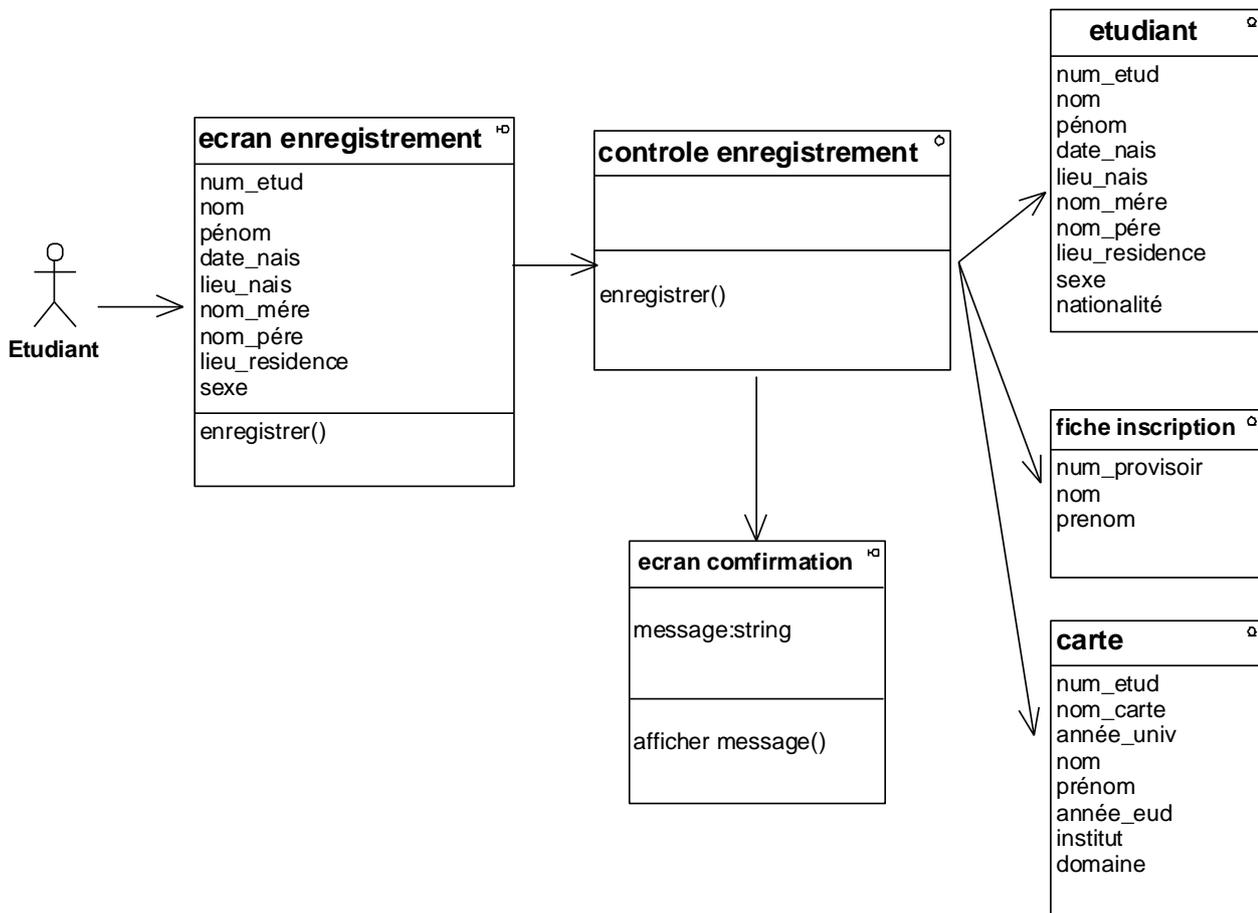


Figure 3.42 : diagramme de classe de conception du cas d'utilisation: enregistrer.

9.3.6 valider

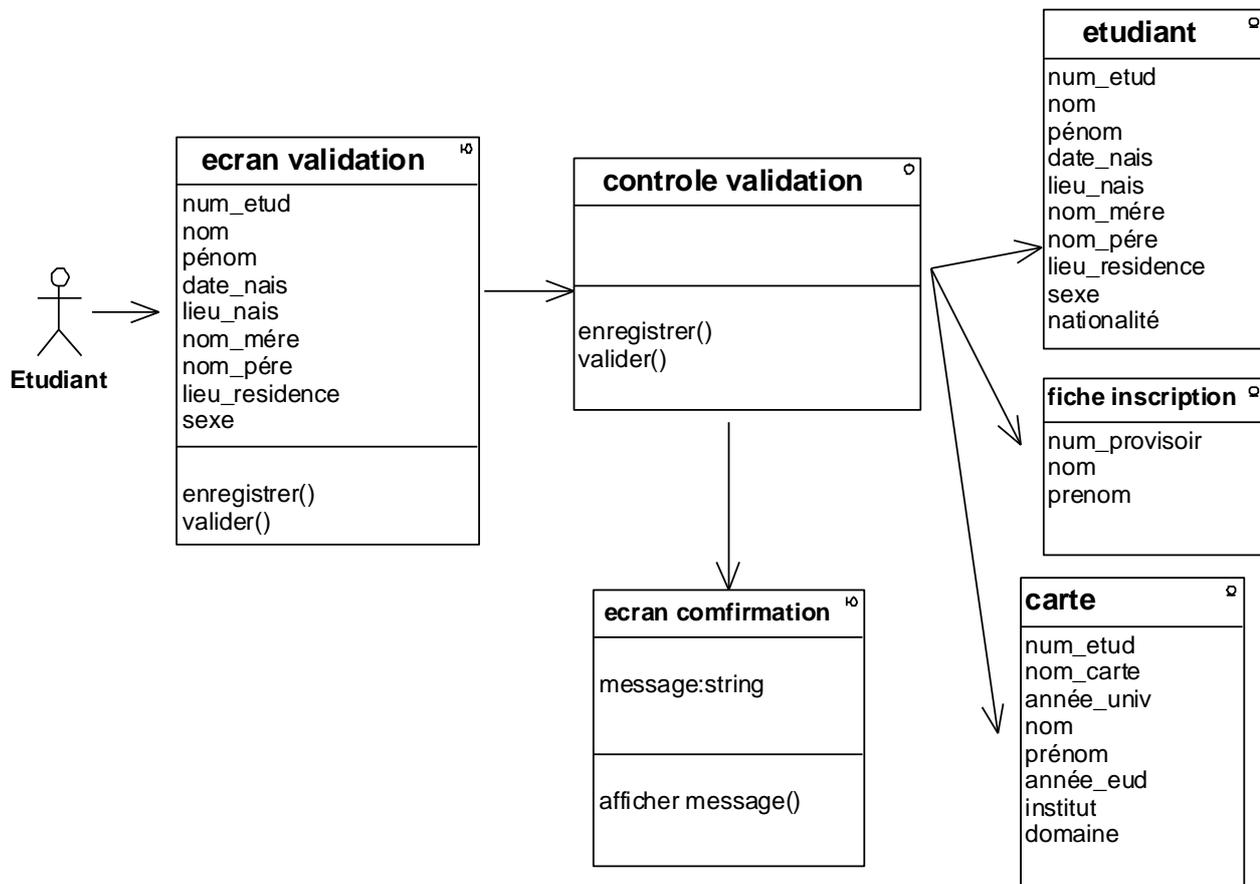


Figure 3.43 : diagramme de classe de conception du cas d'utilisation: valider.

10. CONCLUSION

Dans ce chapitre, nous avons utilisé le langage de modélisation UML pour les applications web, en suivant une démarche simplifiée de développement pour réaliser l'analyse et la conception de notre application. L'analyse nous a permis de recueillir les besoins fonctionnels, de les modéliser, de spécifier leur réalisation grâce aux diagrammes de Cas d'utilisation, d'activité et de séquence.

L'analyse du domaine nous a permis de dresser le modèle de classes métier.

1. Introduction

Nous allons citer dans ce chapitre la liste des outils ainsi les langages de programmation Web qu'on a réuni pour l'implémentation et la réalisation de notre site en expliquant le rôle de chaque outil.

2. Outils de travail

2.1 L'environnement de développement

2.1.1 Mozilla Firefox

Firefox est un navigateur Web, un logiciel qui permet d'afficher des pages internet. Créé et développé par la fondation Mozilla, il est gratuit et connaît un certain succès puisqu'il est le navigateur le plus populaire après Internet Explorer, son concurrent de chez Microsoft. Firefox n'intègre pas de client e-mail mais le chargement des pages est plus rapide et surtout beaucoup plus respectueux des standards, pour voir enfin les pages comme les webmasters l'ont pensées. Ses nombreuses extensions permettent également de personnaliser le logiciel à sa guise, avec de nombreux gadgets plus ou moins bien pensés. Firefox a véritablement tout pour plaire à ceux qui souhaitent avant tout naviguer sur le net, dans les meilleures conditions. Firefox a depuis longtemps prouvé qu'il était plus qu'une alternative au navigateur propriétaire de Microsoft.



2.1.2 Notepad++ :

Notepad++ est un éditeur de texte générique codé en C++, qui intègre la coloration syntaxique de code source pour les langages et fichiers C, C++, Java, C#, XML, HTML, PHP, Java Script, ASP, VB/VB Script, SQL, Objective-C, CSS, Pascal, Perl, Python, Assembleur, Lisp, et VHDL ; ainsi que pour tout autres langages informatiques, car ce logiciel propose la possibilité de créer ses propres colorations syntaxiques pour un langage quelconque. Ce logiciel a pour but de fournir un éditeur léger (aussi bien au niveau de la taille du code compilé que des ressources occupées durant l'exécution) et efficace. Il est également une alternative au bloc-notes de Windows (d'où le nom). [6]



2.1.3 Serveur Apache et EasyPHP :

Apache est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows. EasyPHP est téléchargeable, il regroupe les applications suivantes :



- Le serveur web Apache
- Le serveur de bases de données MySQL
- Le serveur d'application PHP
- L'outil phpMyAdmin permettant de gérer des bases MySQL

Apache possède de nombreuses fonctionnalités dont la possibilité de définir une configuration spécifique à chaque fichier ou répertoire partagé, ainsi que de définir des restrictions d'accès grâce aux fichiers htaccess. [1]

2.1.4 Pacestar UML Diagrammer 6.02 :

Un programme qui fournit un ensemble complet d'outils de modélisation graphique, d'analyse et de conception dans le développement de logiciels basés sur les modèles UML, COM, OMT.

2.2 Les langages de programmation

2.2.1 Coté client

a- HTML

HTML « Hyper Texte Markup Langage » est un langage permettant de décrire les différents composants d'un document : Définir des titres, construire des tableaux, mettre en formes des textes. Le HTML est un langage simple rédigé sous forme de texte et qui demande un simple éditeur de texte.

Le lien Hypertexte est un des principaux atouts du langage HTML, un lien permettant à l'utilisateur d'accéder rapidement à un autre emplacement du document ou à un autre page sur Internet

Voici un exemple sur un code HTML :

```
<Html>
<Head>
<Title>Le titre de la page</Title>
</Head>
<Body>
<h1>Mon premier titre</h1>
<p>Mon premier <b>paragraphe</b></p>
</Body>
</Html>
```

b- Java Script

Java Script est un langage de programmation qui peut être inclus dans des pages HTML destinées aux navigateurs Web les plus courants. Grâce à ce langage, il est possible d'écrire des pages interactives. Java Script ne doit pas être confondu avec Java, qui est un langage beaucoup plus complexe permettant de créer des applications autonomes.

c- CSS (feuilles de style en cascade)

Le langage informatique CSS (Cascading Style Sheets : feuilles de style en cascade) sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien supportés par les navigateurs web dans les années 2000. L'un des objectifs majeurs de CSS est de permettre la stylisation hors des documents. Il est par exemple possible de ne décrire que la structure d'un document en HTML, et de décrire toute la présentation dans une feuille de style CSS séparée. Les styles sont appliqués au dernier moment, dans le navigateur Web des visiteurs qui consultent le document. Cette séparation fournit un certain nombre de bénéfices, permettant d'améliorer l'accessibilité, de changer plus facilement de structure et de présentation, et de réduire la complexité de l'architecture d'un document. Ainsi, les avantages des feuilles de style sont multiples :

1. La structure du document et la présentation peuvent être gérées dans des fichiers séparés.

2. La conception d'un document se fait dans un premier temps sans se soucier de la présentation, ce qui permet d'être plus efficace.
3. Dans le cas d'un site web, la présentation est uniformisée : les documents (pages HTML) font référence aux mêmes feuilles de styles. Cette caractéristique permet de plus une remise en forme rapide de l'aspect visuel.
4. Le code HTML est considérablement réduit en taille et en complexité, puisqu'il ne contient plus de balises ni d'attributs de présentation. [5]

2.2.2 Coté serveur

a- PHP

PHP (Personal Home Page) est langage de script coté serveur, il constitue une extension utilisée sur les serveurs Internet, permettant de créer des pages Web dynamiques, destinées aux applications Internet multimédia ou de e-commerce. [11]

La syntaxe du PHP s'inspire largement du langage C, avec quelques traits de parenté des langages Perl et Java. C'est un langage multi plates formes: porté sur de nombreuses stations UNIX. [10]

PHP est un langage facile à utiliser, caractérisé par une grande richesse fonctionnelle notamment vis-à-vis de la connexion à des bases de données.

b- MySQL

MYSQL est un Système de Gestion de Base de Données (SGBD) fonctionnant sous Linux et Windows. Depuis la version 3.23.19, MYSQL est sous Licence GPL (aussi bien sous Linux que Windows), ce qui signifie qu'il peut être utilisé gratuitement.

MYSQL consiste en un ensemble de programmes qui sont chargé de gérer une ou plusieurs bases de données, et qui fonctionnent selon aune client/serveur.

3. Passage du modèle objet au modèle relationnel

L'utilisation d'un SGBDR impose un enchaînement de représentation entre la structure des classes et la structure des données relationnelles, les deux structures ayant des analogies, les équivalences exprimées au tableau sont utilisées pour en réaliser le rapprochement. Une classe définie une structure de données à laquelle souscrivent des instances ; elle correspond donc à une table du model relationnel :

Chaque attribut donne lieu à une colonne, chaque instance stocke les données dans une ligne et son ID sert de clé primaire. Certain attributs de type complexe ne correspondent à aucun des types de SQL ; on rencontre fréquemment ce cas pour les attributs représentant une structure de données. Un type complexe peut être conçu :

- Soit avec plusieurs colonnes, chacune correspondent à un champ de la structure.
- Soit avec table spécifique d'une clé étrangère pour relier les instances aux valeurs de leurs attributs complexe.

Model d'objet	Model relationnel
Classe	Table
Attribut de type simple	Colonne
Attribut de type composé	Colonne ou clé étrangère
Instance	T-uplet
ID	Clé primaire
Association	Clé étrangère
Héritage	Clé primaire identique sur plusieurs tables

Tableaux 4.1: Table du passage du modèle objet au modèle relationnel

Liste des relations

Compte (num_compte, nom_utilisateur, mot_passe, type, #id_personne, #num_provésor).

Personne (id_personne, nom, prénom, #num_compte).

Etudiant (num_provésor, num_etud, nom, prénom, date_naiss, lieu_naiss,

Nom_mère, nom_père, lieu_résidence, sexe, nationalité, #num_compte, #num_bac, #num_fiche).

Carte (num_carte, num_etud, num_carte, année_univ, nom, prénom,

Année_etud, Institut, domaine, #num_provésor).

Attestation_bac (num_bac, filiere_bac, wilaya_bac, moy_bac, année_bac, #num_provésor).

Fiche_inscription (num_fiche, nom, prénom, #num_provésor).

4. Le plan du site

Notre site contient une page principale « Accueil » composée de trois pages :

1. Page étudiant :
 - Remplir formulaire.
 - Modifier.
2. Page administrateur :
 - créer compte pédagogie.
3. Page Agent pédagogie :
 - Rechercher.
 - Supprimer compte pédagogie.

5. Manuel d'utilisation du site

Nous présentant dans cette partie comment utiliser notre application. En suite on donne une description générale de son contenu et son mode de fonctionnement.

Page d'accueil

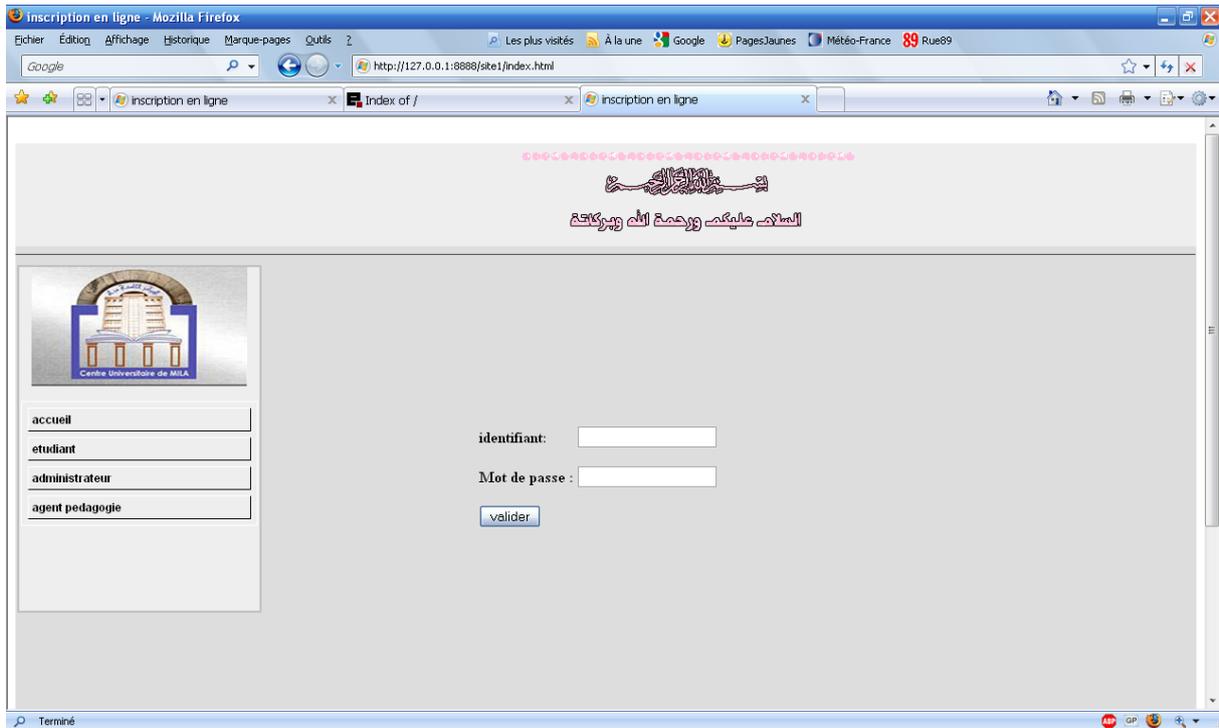


Figure 4.1 : Page d'accueil

Dans cette page un internaute peut s'authentifier et consulter les différents services du site : La page contient les trois pages : Etudiant, Administrateur et Agent pédagogique.

Page créer compte

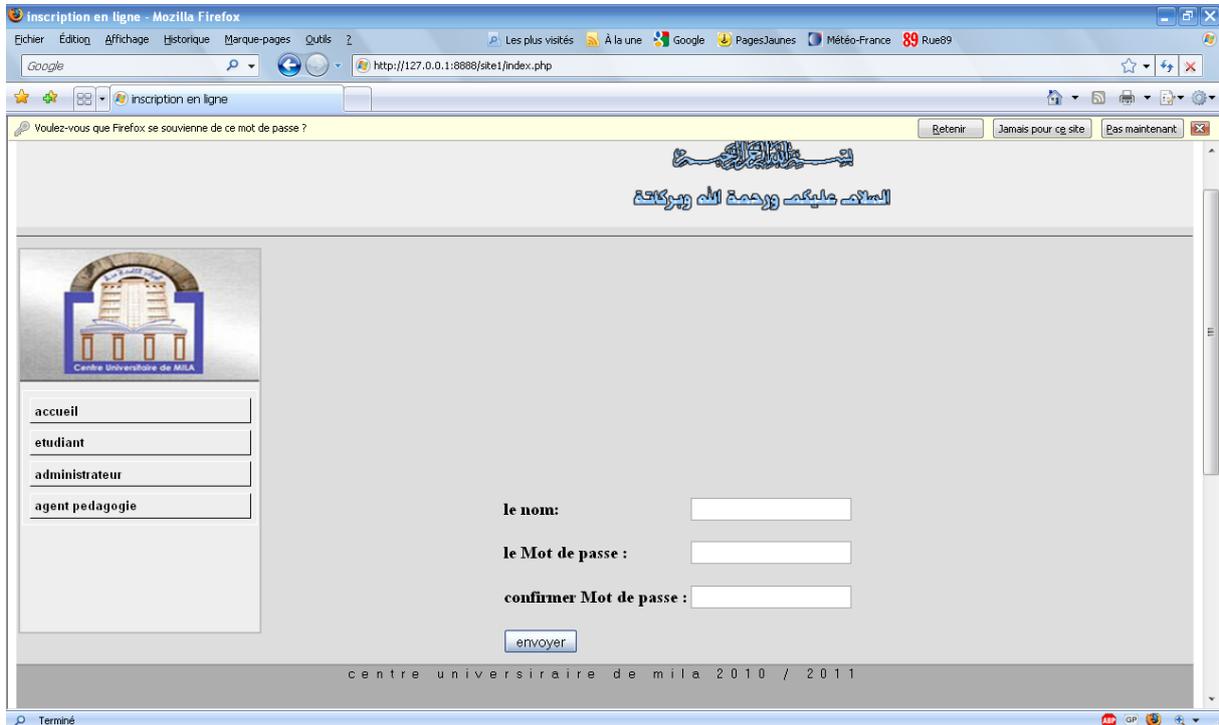


Figure 4.2 : Page créer compte.

Cette page permet à l'étudiant de créer un compte.

Page étudiant

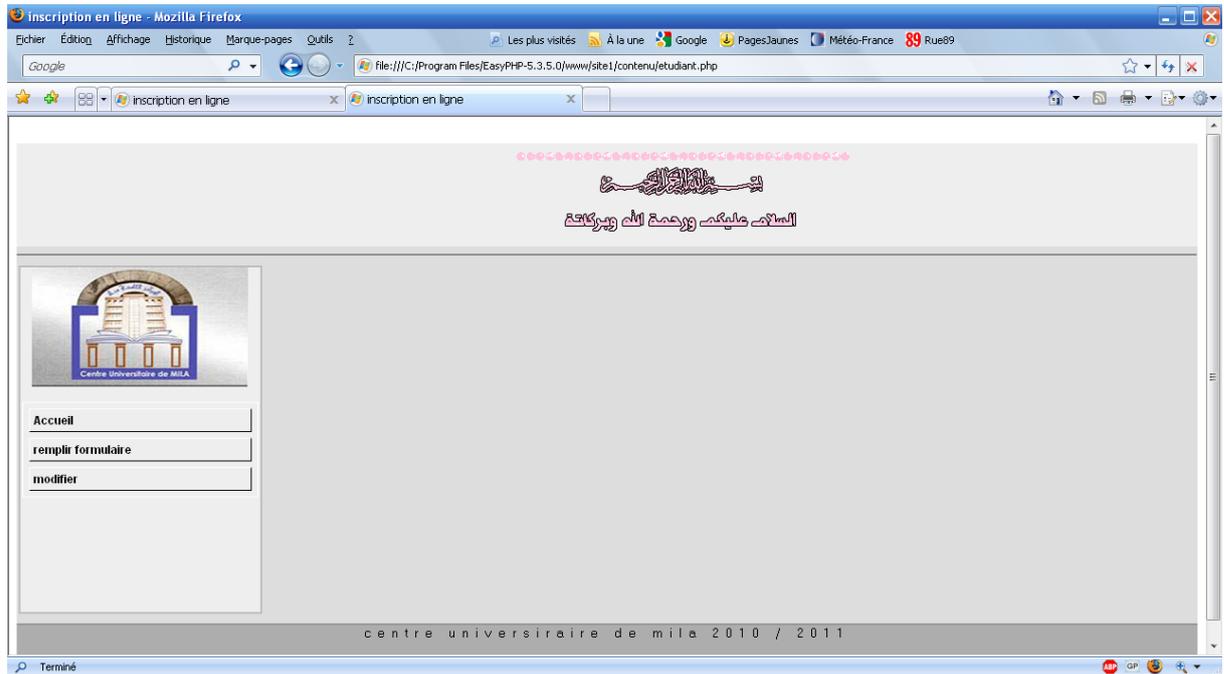
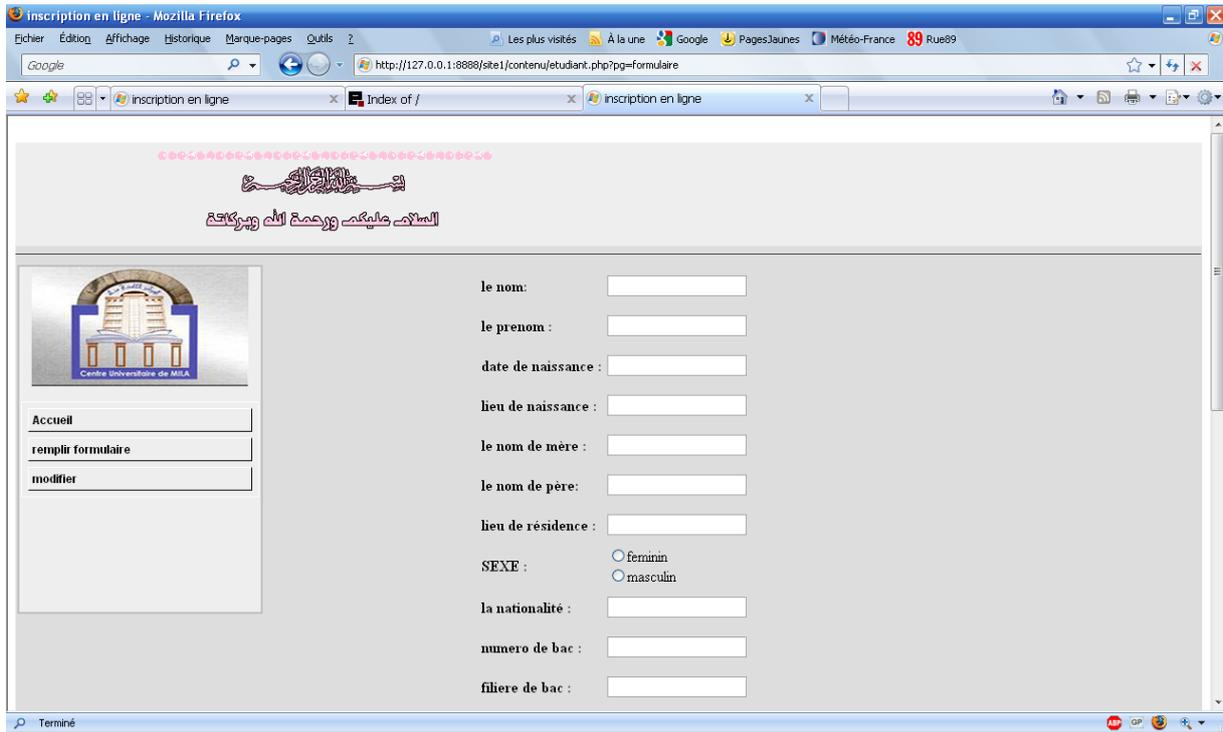


Figure 4.3 : Page d'étudiant

Dans cette page l'étudiant peut : remplir ou modifier les formulaires qui contiennent ses informations.

Page remplir formulaires



The screenshot shows a web browser window titled "Inscription en ligne - Mozilla Firefox". The address bar displays "http://127.0.0.1:8888/site1/contenu/etudiant.php?pg=formulaire". The page content includes a header with Arabic text and a logo for "Centre Universitaire de MILA". A sidebar on the left contains three buttons: "Accueil", "remplir formulaire", and "modifier". The main content area contains a registration form with the following fields:

- le nom:
- le prenom:
- date de naissance:
- lieu de naissance:
- le nom de mère:
- le nom de père:
- lieu de résidence:
- SEXE: féminin masculin
- la nationalité:
- numero de bac:
- filiere de bac:

Figure 4 .4 : Page remplir formulaire.

Cette page permet à l'étudiant de taper et enregistrer ses informations.

Page d'agent pédagogie

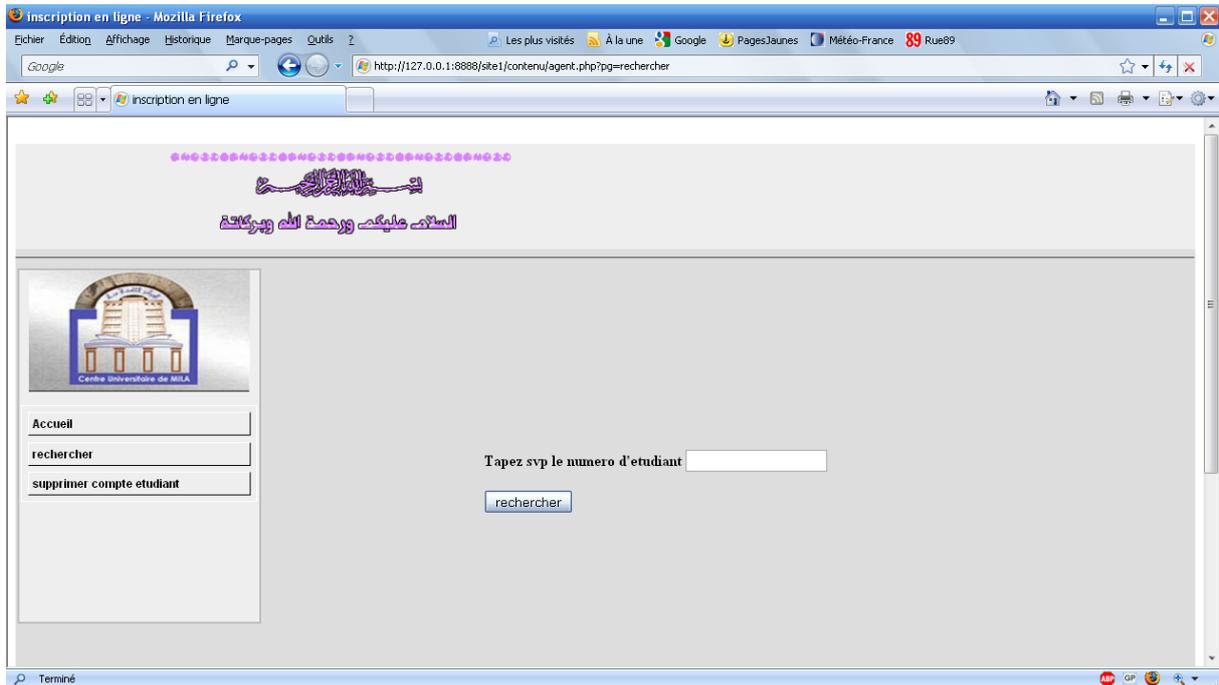


Figure 4.5 : Page Agent pédagogie.

Dans cette page l'agent peut rechercher l'étudiant ou supprimer son compte.

Page administrateur

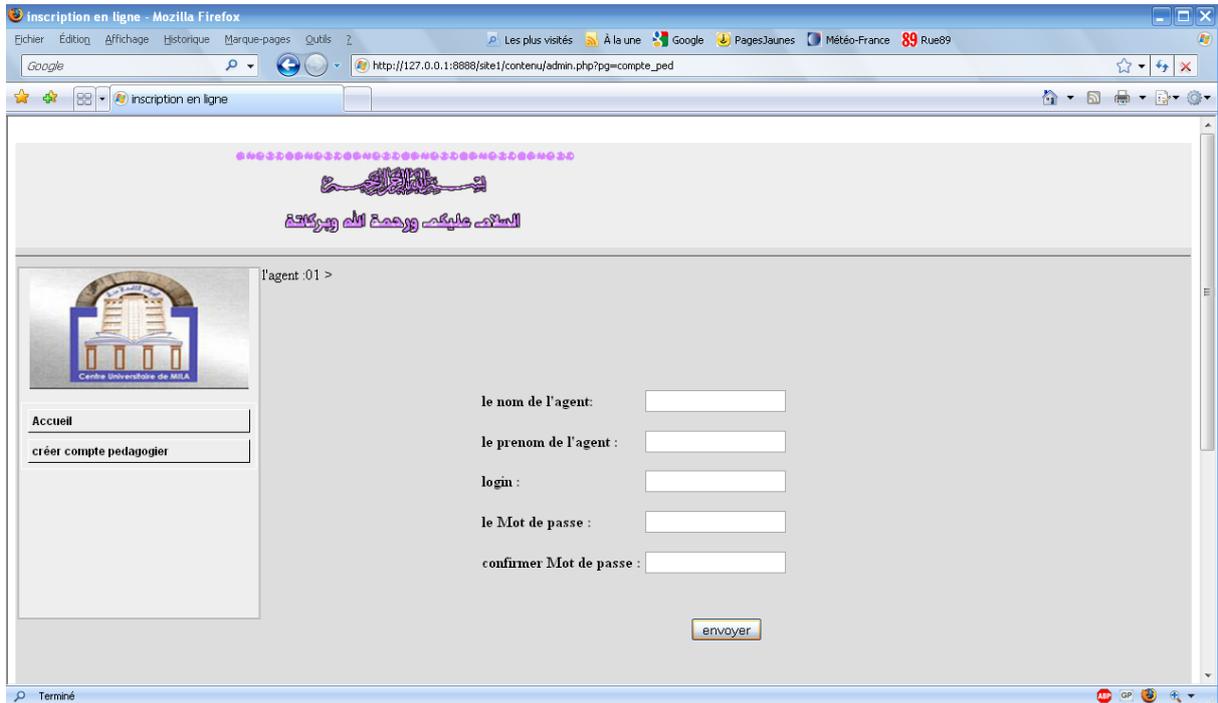


Figure 4.6 : page administrateur.

Cette page permet à l'administrateur de créer les comptes pédagogie.

Conclusion Générale

Le Web est avant tout un outil de communication et de transfert d'information. Très vite, se ressent le besoin de faire connecter des données avec ce moyen de communication et de les mettre sur le réseau.

On a tenté à travers ce projet de présenter une application client/serveur pour l'inscription en ligne des étudiants au niveau du centre universitaire de Mila.

Pour réaliser effectivement ce travail, on a eu besoin d'un ensemble varié d'outils de modélisation et de programmation pour les sites dynamiques.

Pour mener la phase d'identification de besoins, d'analyse et de conception, nous avons suivi une démarche basée sur le langage de modélisation UML.

Quant à la phase d'implémentation, nous avons tiré profit des langages PHP et HTML qui sont considérés comme les langages les plus importants dans le monde d'implémentation des applications Web. En effet le langage PHP assure la possibilité de communiquer facilement avec la base de données MySQL.

BIBLIOGRAPHIQUES

Les sites Internet

[1] : www.commentcamarche.net

[2] : fr.wikipedia.org/wiki/Page_Web_dynamique.htm

Les mémoires de fin d'étude

[3] : Dib Abderrahim, Sahli younes « Système Multi agents pour le commerce électronique » , Université Mentouri de Constantine 2003.

[4] : Chahdane karima, Benhammada lamyia : « Conception et réalisation d'un site Dynamique Pour Une Agence Commerciale d'ET - Etude de cas Gestion des demandes et réclamations des abonnées » Université Mentouri de Constantine 2010

[5] : Atmane Bilel, Mebiroki Ali : « Conception et réalisation d'un site web dynamique pour Le journal EL ACIL » Université Mentouri de Constantine 2010

[6] : djouambi Mohamed, lemeurs abd elbassit, fredji younes « Conception et réalisation d'un site web dynamique de centre universitaire de mila » , Université Mentouri de Constantine 2008

[7] : Bouziane Youssef, Latreche Hakim, « conception et réalisation d'un système pour le vote électronique», université Mentouri de Constantine promotion 2007/2008.

Les livres

[8] : M. Grimaldi ; modélisation UML diagrammes structurels ; février 2007

[9] : Cours Formulaire PHP

[10] : GA. Lelerer et R.Stoll ; Grand Livre PHP & MySQL ; Traduction: Samy Boutayeb ; 1^{ère} édition août 2000

[11] : Mattzandastr ; Ouvrage [8]: PHP 4 ; Publié par Compus press; édition 2000