

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

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DUGENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes De MASTER ACADEMIQUE

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

Spécialité : **conduite de projet informatique**

Présenté par
ZEGGANE Said
LASSOUAOUI Hakim

Thème
***Conception et réalisation d'une Application
Client/serveur gestion de scolarité pour le
département informatique***

Mémoire soutenu publiquement le...../...../ 20..... devant le jury composé de:

Président :

Encadreur :

Examineur :

Examineur :

∞ Remerciements ∞

Nous tenons vivement à remercier notre promotrice Mme Haddaoui pour avoir accepté d'encadrer ce travail et aussi pour tous les conseils prodigués pour réaliser ce travail.

Nos remerciements s'adressent à tous ceux qui nous ont aidés de près ou de loin dans notre travail et qu'ils trouvent l'expression de notre profonde gratitude

.

Dédicaces

Je dédie ce modeste travail :

En premier lieu à mes très chers parents a qui j'offre toutes mes réussites et tous les mérites. Je leur serais éternellement dévoué,

À mes frères (Achène et Missipsa)

À ma sœur (Thanina)

À qui je souhaite un très bel avenir

À ma belle sœur Soraya et mon petit neveu Elyes

À ma fiancée Fatma et sa famille

À ma grande-mère à qui je souhaite une longue vie, ainsi qu'à mon adorable famille, du plus grand au plus petit.

À mon oncle Embarek et sa femme

À tous mes amis

À mon binôme, Hakim, et à toute sa famille que je remercie de leur hospitalité,

À toutes personnes ayant contribué à la réussite de ce travail.

Z.said

Dédicaces

Je dédie ce modeste travail :

En premier lieu à la mémoire de ma très chère mère .

Et à mon père qui n'a jamais cessé d'être à mes côtés par ces précieux conseils et orientation ainsi que son soutien moral.

A la mémoire de mes grands parents.

A ma grand-mère OUIZA à qui je souhaite longue vie.

A mes frères (Kamel, Sidali et Mourad).

A mes deux sœurs (Taoues et Kenza).

A tous mes oncles, tantes et cousin(e) s sans exception et toute la famille un par un.

A tous mes amis en particulier

(AZDINE GHILES MASSI RAMDANE AMINE NACER ET SALEM).

A mon binôme SAID et à toute sa famille.

A Toutes personnes qui m'ont soutenue tout au long de mes études.

L.Hakim

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre I : Présentation de l'organisme d'accueil.

1.Présentation de l'organisme d'accueil.....	3
2.Missions de l'UMMTO.....	3
3. Mode d'organisation.....	4
4.Présentation de la structure d'accueil	5
4.1. Organisation du Centre de Calcul de l'U.M.M.T.O :	6
I.4.2 Les missions du centre de calcul :	7
5Présentation du champ d'étude.....	7
5.1. L'armature pédagogique de l'UMMTO:	7
5.2 Organisation et mission d'un département:	8
6. Problématique:	9
7. Objectifs:.....	9
8. conclusion:.....	10

Chapitre II : l'architecture client-serveur

1. Introduction:	11
2.Le modèle client /serveur:.....	12
2.1 Les techniques de dialogue client/serveur	12
2.2. Notions de base du modèle client/serveur:.....	12
3. Avantages du modèle client/serveur	13
4.l'architecture client/serveur multi niveaux	14
4.1. Architecture client /serveur a deux niveaux.....	14
4.2. Architecture client/serveur à trois niveaux:	15
4.3. Architecture client/serveur à multi niveaux.....	16
5. Conclusion	17

Chapitre III : Analyse et conception

1. Introduction	182.
Analyse	18
III.2.1 Identification des besoins	18
III.2.2 Identification des acteurs	18
III.2.3 Identification des cas d'utilisations	19
3 Conception	20
A. Le diagramme des cas d'utilisation.....	21
A.1 Le diagramme des cas d'utilisation global.....	22
A.2 diagramme de cas d'utilisation « espace administrateur ».....	23
A.3. Diagramme de cas d'utilisation « Espace Etudiant ».....	24
A.4. Diagramme de cas d'utilisation « Espace enseignant »	25
B. Diagrammes de séquences	26
B.1.diagramme de séquence du cas d'utilisation « ajouter examen »	27B.
2.diagramme de séquence du cas d'utilisation « Ajouter étudiant »	27
B.3.diagramme de séquence du cas d'utilisation « chercher étudiant »	28
B.4.diagramme de séquence du cas d'utilisation « ajouter module ».....	28
C. Diagramme de classes	29
3.1.Le modèle logique	30
4. Conclusion	36

Chapitre IV : Réalisation et mise en œuvre

1. Introduction	38
2. Architecture de l'application	38
3. L'environnement de développement	38
3.1 Langages de programmation	38
3.1.1 PHP.....	39
3.1.2 Structure d'une page PHP	40
3.1.3 Fonctionnement de PHP.....	41
3.2 Serveur Web Apache	41
3.3 Serveur MySQL	42
3.3.1 MySQL	42
3.3.2 Fonctionnalités de MySQL.....	42
IV.4 Les outils de développement	44
4.1 wamp server.....	44

4.2 PhpMyAdmin.....	44
4.2.1 Les fonctions de PhpMyAdmin	45
4.2.2 Utilisation de PhpMyAdmin.....	45
4.3 Autre outils	46
5. Présentation de quelques interfaces de notre plate-forme	47
5.1.1 La page login	48
5.1.2: Interface d'accueil	48
5.1.3 : Interface cherche enseignant.....	49
5.1.4 Interface ajouté examens	50
5.1.5 : Interface consulter la liste des étudiants	51
5.1.6 : Interface ajouter un examen.....	51
5.1.7 : Interface chercher un stage pratique	52
Conclusion générale	53
Bibliographie	

Introduction

Générale

Au cours de ces dernières années, les nouvelles technologies de l'information et de la communication ont connu un bouleversement marqué par l'apparition de l'Internet et par sa croissance exponentielle. L'arrivée de l'Internet marque un nouveau tournant dans l'ère de communication, c'est une amélioration des méthodes traditionnelles dans le sens qu'il n'est pas géographiquement limité et donne la possibilité de communiquer à l'échelle mondiale à tout utilisateur d'ordinateur. Notamment avec l'apparition du web, le service le plus populaire de l'Internet qui ouvre grand les portes dans le grand public sans tenir compte du niveau intellectuel, ni du statut.

C'est dans cette optique que le département d'informatique de l'université envisage de s'investir afin d'exploiter les biens de la technologie Internet en son profit. Pour cela, un travail nous a été confié, consistant en une conception et réalisation d'une application web pour gérer la scolarité du département .

Pour mener à terme notre travail, nous le répartissons de la manière suivante :

- Le premier chapitre s'intitule « Etude préalable » comprend la présentation de l'organisme d'accueil, historique, et mode d'organisation
- Le deuxième chapitre s'intitule « Architecture client/serveur » qui comprend la définition du client/serveur, son fonctionnement, les différents types de client et de serveur,...
- Le troisième chapitre sous le nom « analyse et conception », qui est consacré à l'analyse et la conception de notre application, qui est de réaliser un intranet web pédagogique, pour cela nous avons opté pour le langage UML étant le mieux adapté pour les applications web, pour se faire, nous avons utilisé quelques diagrammes :

-Diagramme de cas d'utilisation pour la représentation des besoins des utilisateurs

-Diagramme de séquence pour la représentation des cas d'utilisation

-Diagramme de classe pour la représentation des objets du système tel que les pages web

Le quatrième chapitre « Réalisation », comporte quand à lui la présentation de l'environnement dont lequel notre application a été réalisé, les outils utilisés et quelques interfaces de notre

Le cinquième chapitre « sécurité »

CHAPITRE

Généralités sur les réseaux, internet & web

1. Présentation de l'organisme d'accueil:

L'Université Mouloud Mammeri de Tizi-Ouzou (U.M.M.T.O) créée par l'ordonnance N° -17-77 du 20 Juin 1977 en tant que centre universitaire (C.U.T.O) dont le premier noyau fut réalisé dans l'ex-asile de vieillards de Oued-Aïssi.

La première rentrée universitaire de septembre 1977 avait accueilli 490 étudiants encadrés par 27 jeunes enseignants dans une structure de quatre départements. Ceux-ci évoluaient dans une salle de cinéma de la ville et un sous-sol de la mairie improvisés alors en amphithéâtres, 07 salles de travaux dirigés, 06 laboratoires et 02 laboratoires de langue.

En 1989, cet important pôle a été élevé au rang d'Université de Tizi-Ouzou (U.T.O.) par le Décret Exécutif N°89-139 du 01/08/89 avec des neuf instituts.

En 1991, l'université de Tizi-Ouzou enrichit son offre de formation par la création du -
Département de Langue et Culture Amazighes. Entre temps, l'université de Tizi-Ouzou fut baptisée à juste titre du nom de l'illustre écrivain Mouloud Mammeri, décédé le 26 février 1989. Elle a dû fonctionner ainsi jusqu'en 1998 où un décret présidentiel remodèle en profondeur l'organisation de l'université algérienne d'une manière générale, y compris dans la gestion de ses aspects pédagogiques.

Actuellement, avec des effectifs progressant rapidement, l'Université Mouloud Mammeri de Tizi-Ouzou ne cesse d'élargir son offre de formation par la création de nouvelles filières et spécialités tant en graduation qu'en post-graduation. Même si elle demeure en phase de construction, l'université Mouloud Mammeri affirme chaque année un peu plus son identité et consolide sa tendance à jouer un rôle de pôle scientifique important tant à l'échelle nationale qu'internationale, grâce notamment aux potentialités qu'elle recèle.

2. Missions de l'UMMTO:

L'une des missions principales de l'université est de contribuer au développement du capital humain et de diffuser le savoir.

Dans un monde en continuelle mutation où les frontières ne constituent plus que des écueils administratifs, la société attend de l'université qu'elle concilie au moins quatre réalités : sa mission de formation et de recherche, sa fonction critique, critique sociale, économique et

éthique s'entend, son positionnement spatial auprès de ses interlocuteurs et partenaires immédiats et de la communauté scientifique internationale, son soutien à l'innovation comme acteur direct et comme partenaire. Devant ces exigences, il est intéressant de s'interroger dans quelle mesure la jeune université algérienne en général et l'université Mouloud Mammeri de Tizi-Ouzou en particulier peut répondre à ces attentes et quelles sont les conditions nécessaires pour y parvenir à moyen et à long termes.

3. Mode d'organisation:

La direction de l'UMMTO est structurée comme suit:

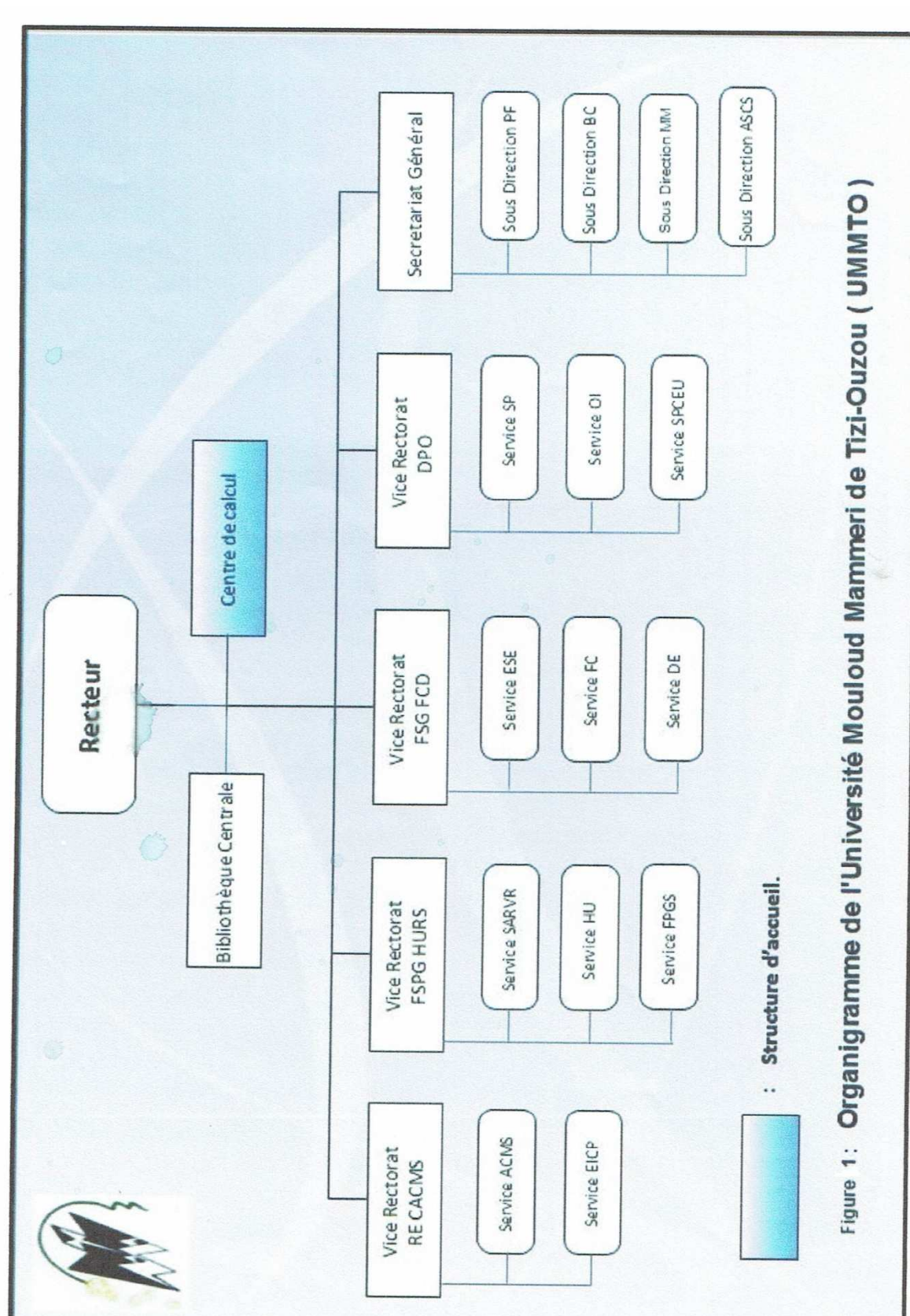
- Recteur
- ✓ Vice Rectorat des Relations Extérieures, Coopération, Animation,

Communication et Manifestations Scientifiques (RE CACMS)

- ✓ Service Animation, Communication et Manifestations Scientifiques (ACMS)
- ✓ Service Echanges Interuniversitaires, Coopération et Partenariat (EICP)
- ❖ Vice Rectorat de Formation Supérieure de Post-Graduation, d'Habilitation Universitaire et de Recherche Scientifique (FSPG HURS)
 - ✓ service de Suivi des Activités de Recherche et de Valorisation de ses Résultats (SARVR)
 - ✓ Service Habilitation Universitaire (HU)
 - ✓ Service Formation Post-Graduation et Post-Graduation Spécialisée (FPGS)
- ❖ Vice Rectorat de Formation Supérieure de Graduation, de Formation Continue et des Diplômes (FSG FCD)
 - ✓ Service Enseignements, Stages et Evaluation (ESE)
 - ✓ Service de Formation Continue (FC)
 - ✓ Service Diplômes et Equivalences (DE)
- ❖ Vice Rectorat de Développement, Prospective et Orientation (DPO)
 - ✓ Service Statistiques et Prospective (SP)
 - ✓ Service Orientation et Information (OI)
 - ✓ Service de Suivi des Programmes de Construction et Equipement de l'Université (SPCEU)
- ❖ Secrétariat Général
 - ✓ Sous Direction de Personnels et Formation (PF)

- ✓ Sous Direction du Budget et Comptabilité (BC)
- ✓ Sous Direction Moyens et Maintenance (MM)
- ✓ Sous Direction des Activités Scientifiques, Culturelles et Sportives (ASCS)
- ❖ Bibliothèque Centrale
- ❖ Centre des Systèmes et Réseaux d'Informations de Communication de
Téléenseignement et Enseignement à Distance (Centre de Calcul)

Cette hiérarchie est modélisée par la figure 1



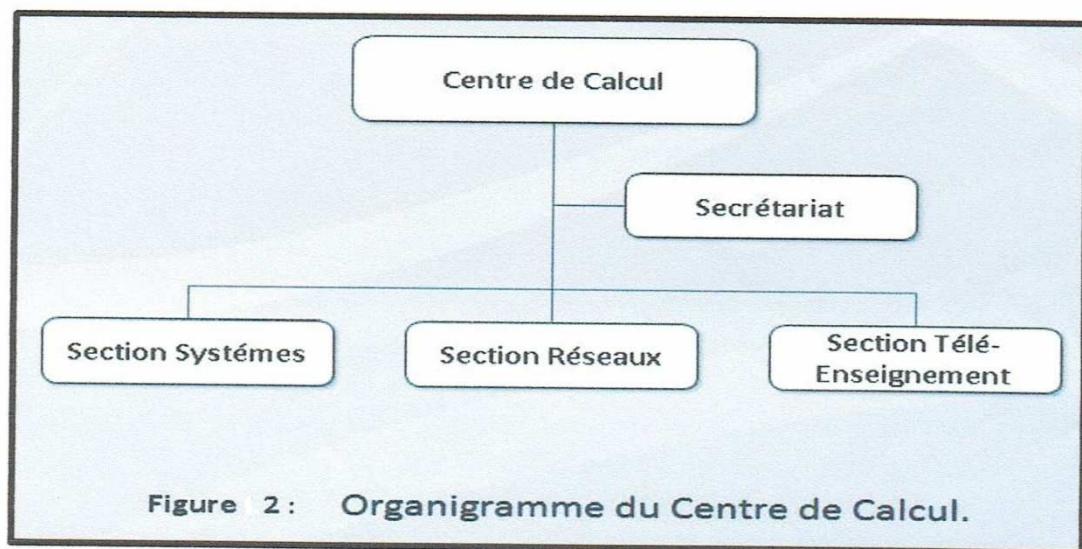
4. Présentation de la structure d'accueil:

4.1. Organisation du Centre de Calcul de l'U.M.M.T.O:

Le Centre de Calcul de l' UMMTO est hiérarchisé comme suit:

- Responsable du Centre de Calcul
 - ❖ Secrétariat
 - ❖ Chef de Section Systèmes
 - ✓ Ingénieurs
 - ✓ Techniciens
 - ✓ Chef de Section Réseaux
 - ✓ Ingénieurs
 - ✓ Techniciens
 - ✓ Chef de Section Télé-enseignement
 - ✓ Ingénieurs
 - ✓ Techniciens

Cette organisation est modélisée par la figure 2



4.2. Les missions du centre de calcul : [Journal Officiel]

Le centre des systèmes et réseaux d'information et de communication, de télé-enseignement et d'enseignement à distance est chargé de:

- ✓ L'exploitation, l'administration et la gestion des infrastructures des réseaux;
- ✓ L'exploitation et le développement des applications informatiques de gestion de la pédagogie;
- ✓ Le suivi et l'exécution des projets de télé-enseignement et d'enseignement distance;
Assurer l'appui technique à la conception et la production de cours en ligne;
- ✓ La formation et l'encadrement des intervenants dans l'enseignement à distance.

5. Présentation du champ d'étude:

5.1. L'armature pédagogique de l'UMMTO:

L'Université Mouloud Mammeri de Tizi-Ouzou contient:

- ✓ Faculté des Sciences
 - ✓ Département Mathématiques
 - ✓ Département Physique
 - ✓ Département Chimie
 - ✓ Département Science de la matière
 - ✓ Département Science et Technologie
 - ✓ Département Mathématique et Informatique
- ✓ Faculté Génie Electrique et Informatique
 - ✓ Département Informatique (Champ d'étude)
 - ✓ Département Electronique
 - ✓ Département Automatique
 - ✓ Département Electrotechnique
- ✓ Faculté Génie de la construction
 - ✓ Département Architecture
 - ✓ Département Génie Civil
 - ✓ Département Génie Mécanique
- ✓ Faculté Médecine
 - ✓ Département Médecine

- ✓ Département Pharmacie
- ✓ Département Chirurgie Dentaire
- ✓ Faculté Science Economique et de Gestion
 - ✓ Département Science Economique
 - ✓ Département Science de Gestion
 - ✓ Département Sciences Commerciales
- ✓ Faculté de Droit
 - ✓ Département Science Juridiques et Administratives
 - ✓ Département Science politiques et Relations internationales
- ✓ Faculté Lettres et Langues
 - ✓ Département Langue et Culture Amazigh
 - ✓ Département Lettre et Langue Arabe
 - ✓ Département Français
 - ✓ Département Anglais
 - ✓ Département Traduction et Interprétariat
- ✓ Faculté Sciences Humaines et Sociales
 - ✓ Département Sciences Humaines
 - ✓ Département Sciences Sociales

Étant donné qu'il y a autant de services scolarité qu'il y a de départements, et que tous ces services fonctionnent de la même manière, alors on va prendre un département parmi ceux-ci, en l'occurrence le département Informatique, il sera le champ d'étude pilote.

5.2. Organisation et mission d'un département: [Journal Officiel]

Un département est structuré comme suit:

- Chef de département: Il est assisté par
 - ❖ Le chef de département adjoint chargé de la scolarité et des enseignements de graduation Il est assisté par :
 - ✓ le chef de service de la scolarité
 - ✓ le chef de service du suivi des enseignements et de l'évaluation.

Il a pour tâches:

- de suivre les opérations d'inscription et de réinscription des étudiants de graduation;

- de veiller au bon déroulement des enseignements;
- de veiller au bon déroulement des examens et épreuves de contrôle des connaissances.
- ❖ Le chef de département adjoint chargé de la post-graduation et de la recherche scientifique : Il est assisté par
 - ✓ le chef de service de la formation supérieure de post-graduation et de la post-graduation spécialisée;
 - ✓ le chef de service du suivi des activités de recherche.

Il a pour tâches:

- de veiller au déroulement des enseignements de post-graduation;
- de veiller au déroulement des formations de post-graduation spécialisée;
- d'assurer le suivi des activités de recherche;
- d'assurer le suivi du fonctionnement du comité scientifique de département.

6. Problématique:

De nos jours, l'outil informatique a touché pratiquement toutes les branches d'activité allant du domaine de gestion jusqu'au champ scientifique. Il est utilisé pour sa capacité à résoudre les problèmes complexes et pour sa fiabilité, ainsi que pour mettre en place des solutions adéquates aux différentes situations de l'entreprise.

En effet les services pédagogiques de l'Université, représente une véritable banque de données sur les étudiants et un suivi permanent de leurs cursus, rendant ainsi leur gestion complexe et pénible.

L'informatisation existant possède de multiples lacunes notamment des bugs qui impliquent parfois la lenteur du travail et de l'information

7. Objectifs:

Dans le but de remédier aux ennuis de ce système , il fallait penser à un moyen fiable, efficace, sûr et surtout compatible avec les moyens des départements.

Les principales caractéristiques de l'application seront:

- Gestion des utilisateurs;
- Gestion des sections;
- Gestion des modules;
- Gestion des examens;
- Gestion des stages pratiques;
- Enregistrement des notes;
- Modification des notes;
- Consultation des notes

L'application est destinée à l'administration, aux enseignants, ainsi qu'à tous les étudiants.

8. conclusion :

Dans ce chapitre nous avons présenter l'organisme d'accueil (UMMTO) et donner certaines informations concernant le fonctionnement de ces administration.

Nous allons dans le prochain chapitre présenter l'architecture client/serveur qui est l'architecture utiliser pour notre travail.

CHAPITRE

Architecture Client/serveur

1. Introduction

De nombreuses applications fonctionnent selon un environnement client/serveur cela signifie que des machines clientes contactent un serveur, une machines généralement très puissante en terme de capacité entrée -sortie, qui leur fournit des services qui sont des programmes fournissant des données telles que les fichiers, une connexion ...

Ces services sont exploités par des programmes, appelés programme clients, s'exécutant sur des machines clientes.

Les applications client/serveur requièrent un canal de communication fiable et accessible entre les systèmes. Pour mettre en œuvre ces applications nous avons besoin d'un réseau qui est chargé de ce lien et qui doit donc être impérativement dimensionné en fonction des exigences de capacité.

L'architecture client /serveur est apparue après un ensemble d'évolutions technologiques, qui ont été enregistrées dans les dernières années : capacité mémoire, performance des processeurs, évolution des réseaux et des différents logiciels.

Cette architecture est utilisée d'une manière très large dans plusieurs domaines.

Par conséquent les applications client/serveur sont distribuées sur plusieurs sites d'où la nécessité de les faire communiquer afin qu'elles coopèrent pour la réalisation d'un travail commun.

- Dans ce chapitre nous allons proposer les différents architectures du modèle client/serveur ainsi les notions de base nécessaires à la compréhension de ce modèle.

2. Le modèle client /serveur [1] :

2.1. Les techniques de dialogue client/serveur :

Le modèle client/serveur est un mode de dialogue entre deux processeurs, le premier Appelé <<client >>et le second appelé <<serveur>>. La communication client/serveur se fait par la fourniture de service par un processeur serveur à un processeur client, cela est du a la transmission d'une requête à un processus serveur qui exécute l'opération demandée et envoi une réponse en retour . Le programme serveur tourne en permanence attendant des requêtes, et peut répondre à plusieurs clients en même temps.

2.2 Notions de base du modèle client/serveur :

- Client : c'est une entité (processus, programme, ordinateur ...) qui demande l'exécution d'une opération à une autre entité par l'envoi d'un message et attendant la réponse à cette opération par un message en retour .
- Serveur : c'est une entité (processus, programme, ordinateur ...) qui effectue une opération demandé par un client et lui transmette la réponse.
- Requête : Message transmis par un client à un serveur décrivant l'opération à exécuter.
- Réponse : Message transmit par un serveur à un client suite à l'exécution d'une opération demandée par ce client.

La figure si dessous illustre ces notions :

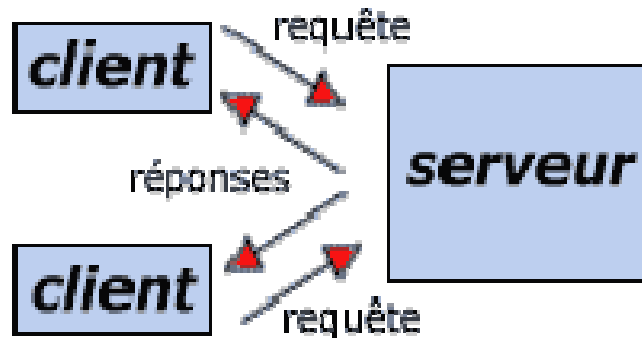


Figure 3 : dialogue client/serveur

3. middleware [1] :

Le middleware est l'un des composants clé de l'architecture client/serveur , c'est un logiciel assurant la modélisation entre client et serveur dans le cadre d'architecture de système hétérogène .En d'autre terme c'est un ensemble de service logiciels construit au dessus d'un Protocol de transport afin de permettre l'échange des requêtes et des réponses associés entre clients et serveur de manière transparente , permettant de cacher l'hétérogénéité des composants mis en jeu (réseau , SGBD ...).

Avantages du modèle client/serveur [2]:

Les principaux avantages du modèle client/serveur sont :

- Flexibilité : il est facile de modifier un système ou un composant.
- Exploitation : il est facile à utiliser.
- Mise à l'échelle : il est facile de mettre à l'échelle le système ou le composant selon les dimensions du problème à résoudre.
- Mieux maîtriser le système d'information : pour mieux maîtriser un système d'information en le rendant plus intégré et évolutif il faut adopter des systèmes ouverts et éviter de s'enfermer sur un constructeur. l'intégrité et l'évolutivité

passent par l'architecture client/ serveur qui permet l'intégration des données et de traitement commun de l'entreprise au sein du serveur relationnel et l'évolutivité des applications développées sur les clients .

- Réutilisation des applications existantes dans l'entreprise :

Ce dernier point est un élément déterminant pour le choix d'une architecture client/serveur.

Prendre en compte les évolutions des technologies : le modèle client/serveur apporte une modularité des composants matériels et logiciels. ceci permet d'intégrer plus facilement les évolutions technologiques dans le système d'information de l'entreprise ce qui est très important.

Malgré tous ces avantages, le modèle a quelques inconvénients parmi lesquels :

- Un maillon faible : le serveur est le seul maillon faible d'une application client/serveur, étant donné que toute application architecturée autour de lui.

4. l'architecture client/serveur multi niveaux : [1]

Les application client/serveur se différencient par la façon dont la communication se partage entre le client et le serveur .le modèle orienté serveur place plus de fonctionnalités sur le serveur . Tandis que dans le modèle orienté client, le gros de l'application tourne du côté client.

Dans le modèle client/serveur on distingue deux architectures :

Architecture client/serveur à deux niveaux et architectures client /serveur a trois niveau.

4.1. Architecture client /serveur a deux niveaux [2] :

L'architecture a deux niveaux ou a 2-tiers caractérise les systèmes client/serveur dans lesquels le client demande un service au serveur qui le lui rend directement sans faire appel à un autre serveur intermédiaire .cette architecture est simple et populaire grâce a sa souplesse et à la facilité de création d'outils frontaux permettant aux utilisateurs de créer leur application . Comme l'illustre la figure ci-dessous.

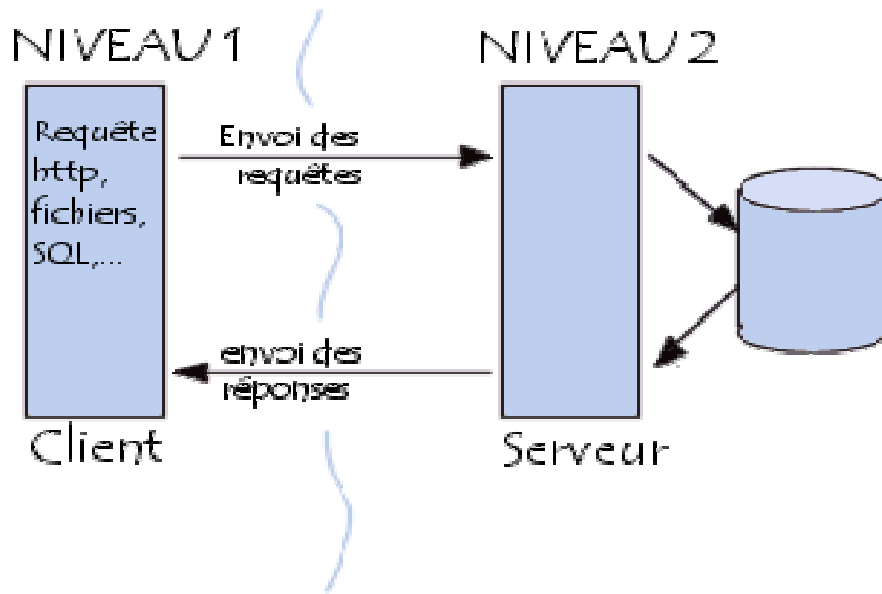


Figure 4.architecture client/serveur à deux niveaux.

4.2 Architecture client/serveur à trois niveaux [2]

L'architecture à trois niveaux est aussi appelée architecture à trois tiers. Elle est partagée généralement entre :

- Un client : qui demande la ressource.
- Un serveur d'application (middleware) : qui est chargé de fournir la ressource en

faisant appel à un autre serveur.

- Un serveur secondaire : qui fournit un service au premier serveur qui est généralement un serveur de base de données.

Les applications de cette architecture à trois niveaux sont plus faciles à déployer et à gérer sur le réseau, elle minimise les charges sur le réseau en créant des niveaux de service.

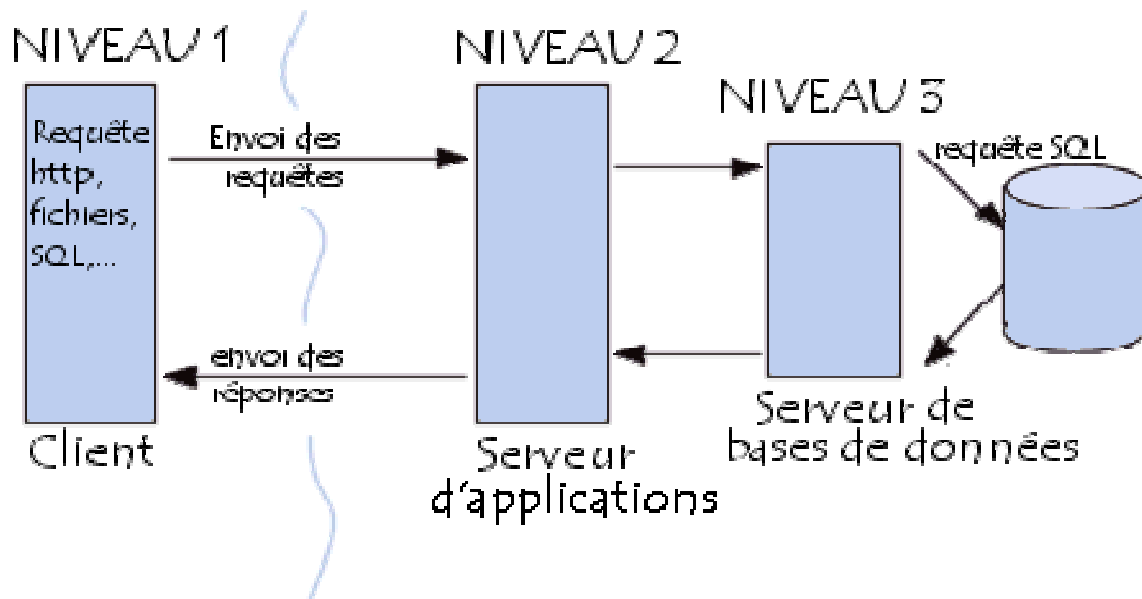


Figure 5 : Architecture client/serveur à trois niveaux.

4.3 Architecture client/serveur à multi niveaux [2] :

L'architecture à trois niveaux devient de plus en plus à N niveaux, cela est du au fait que le niveau intermédiaire est implémenté sous forme d'une collection de composants. Chaque composant se charge d'une fonction relativement petite exécuté suite à une demande d'un client, cette dernière peut nécessiter plusieurs composants. D'où le modèle à trois niveaux est en fait à N niveaux.

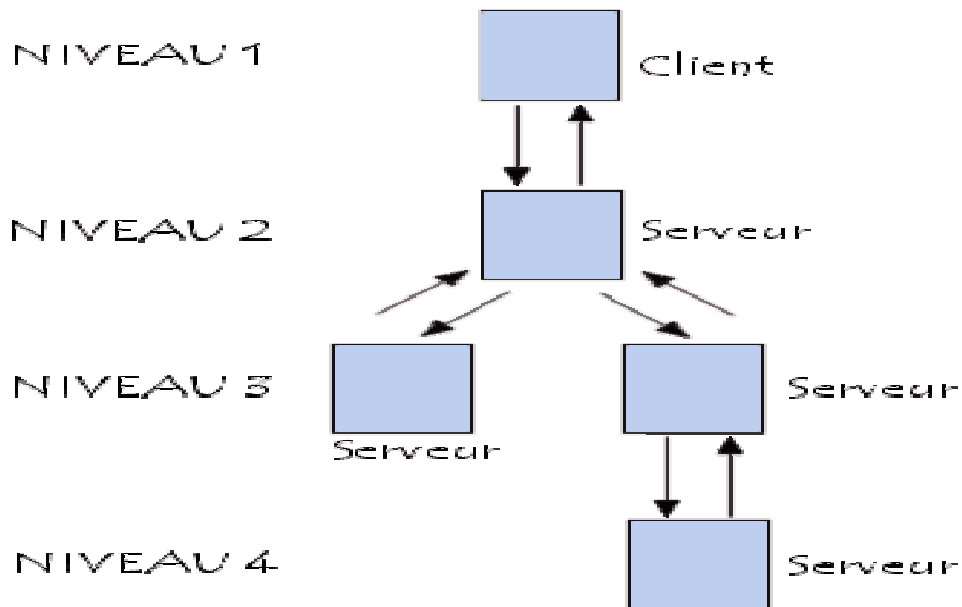


Figure 6 : Architecture client/serveur multi niveaux.

Conclusion

l'architecture client/serveur est une approche qui vise à répartir un système informatisé sur des machines distantes, grâce à des matériels et des protocoles standards, l'internet et bien précisément le web sont bâtis sur cette architecture.

Les applications client/serveur sont au cœur de fonctionnement de l'entreprise et avec l'internet, les serveurs peuvent accepter des requêtes arrivant de millions de PC connectés.

Notre travail consiste à réaliser une application client/serveur pour la gestion de scolarité, dans notre cas on s'intéresse à la faculté génie électrique département informatique.

Le prochain chapitre fera l'objet de l'analyse et conception de notre application

CHAPITRE

Analyse & Conception

1. Introduction :

Pour introduire la partie pratique nous allons présenter la démarche que nous avons adoptée pour mener à bien notre projet. Notre démarche s'appuie sur l'approche objet en utilisant UML (Unified Modeling Language), car cette approche présente un grand nombre d'avantages qui sont essentiellement : La stabilité de la modélisation par rapport aux entités du monde réel, la construction itérative, la réutilisation des composants et la facilité de compréhension des représentations abstraites complexes. UML n'est, toutefois, qu'un langage de modélisation et il doit être accompagné d'un processus qui devra guider la modélisation, étape par étape, jusqu'à la réalisation.

2. Analyse :

Cette partie comprend l'identification des besoins fonctionnels du système, des acteurs et leurs interactions avec le système ainsi que les cas d'utilisation.

2.1 Identification des besoins :

Notre application a pour objectif la réalisation et l'implémentation d'une nouvelle application web pour l'automatisation des procédures de gestion de la scolarité via le réseau internet afin de faciliter les tâches qui se faisaient manuellement auparavant.

2.2. Identification des acteurs :

Définition : Un acteur représente un ensemble de rôles joué par des entités externes (utilisateur humain, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Dans le cas de notre application, nous avons trois acteurs qui interagissent avec le système :

- ✓ Etudiant : un étudiant peut consulter ses notes, voir la liste des enseignants, voir les modules enseignés au département, les types de diplômes et les filières
- ✓ Enseignant : enseignant du département d'informatique peut chercher un étudiant ou un enseignant, ajouter un examen et une évaluation, ainsi modifier en cas d'erreur d'insertion
- ✓ Administrateur : la personne chargée de l'administration de la base de données.

2.3. Identification des cas d'utilisation :

Définition

Un cas d'utilisation représente un ensemble de séquences qui sont réalisées par le système en réponse à une action d'un acteur et qui produisent un résultat observable. Les cas d'utilisation décrivent ce que le système devra faire sans spécifier comment le faire.

Le tableau suivant récapitule les cas d'utilisation de chaque acteur de notre application :

Acteurs	Cas d'utilisations (taches)
Etudiant	T0 : accéder au site. T1 : identification T2 : naviguer dans le site. T3 : consulter la liste des étudiants, ses notes, la liste des enseignants, voir les spécialités, les modules, stages pratique, examen, bulletins, section T4 : chercher un étudiant, un enseignant, stage pratique
Enseignant	T5: idem qu'un étudiant T6 : ajouter un examen, notes,

Administrateur	T9 : idem qu'un enseignant T10 : modifier un étudiant, un enseignant, une spécialité, un module, section, stage pratique T11 : supprimer un étudiant, un enseignant, une spécialité, un module, section, stage pratique T12 : module, filière, section, stage pratique
----------------	---

3. La conception :

Le processus de conception de notre projet se caractérise par deux niveaux : le niveau applicatif et le niveau données. Le niveau applicatif s'appuie essentiellement sur quelques diagrammes de l'extension du langage de modélisation UML pour les applications web. A cet effet nous avons adopté la démarche suivante :

- Après l'identification des différents acteurs ainsi que les cas d'utilisation qui sont mis en œuvre par ces acteurs, le diagramme de cas d'utilisation est élaboré.
- Chaque cas d'utilisation se traduit par un ou plusieurs scénarios. Chaque scénario fait l'objet d'une description sous forme graphique à l'aide d'un diagramme de séquence et un digramme d'activité.
- Une identification des classes est fournie par la synthèse des diagrammes de séquence, ainsi le diagramme de classe sera élaboré.

Le niveau données concerne l'organisation conceptuelle, logique et physique des données manipulées. Durant la partie analyse nous avons pu identifier les données nécessaires et indispensables au bon fonctionnement de l'application et à travers la conception du niveau applicatif nous allons dégager les classes significatives, dès lors on peut élaborer la conception de la base de données.

La figure donne la représentation graphique de la démarche de modélisation adoptée pour concevoir l'application:

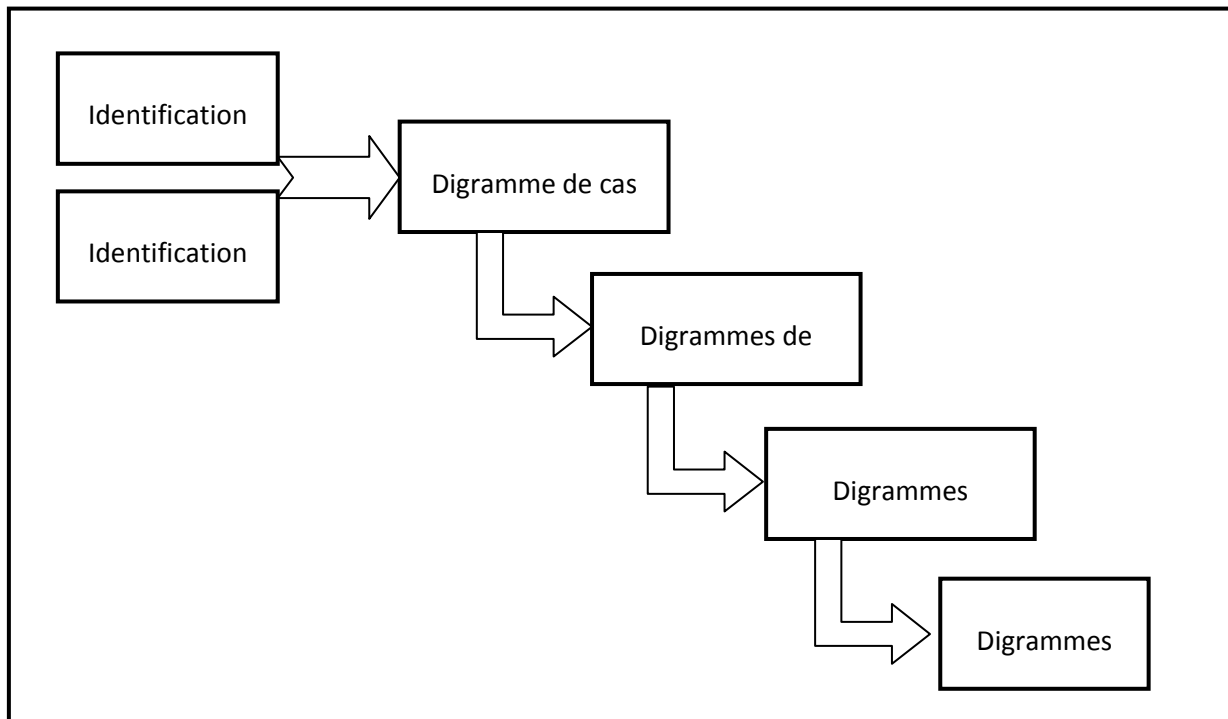


Figure 7 : La démarche adoptée pour la modélisation [4]

A. Diagramme des cas d'utilisation :

Lors de la phase d'analyse nous avons pu identifier les acteurs ainsi que les cas d'utilisation associés à ces derniers. Ce qui nous donne l'opportunité d'élaborer le diagramme des cas d'utilisation.

Définition : Les diagrammes de cas d'utilisation permettent de représenter un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils présentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système.

- La relation d'inclusion (include) : Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations.

- La relation d'extension (Extend) : Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions.

A.1 diagramme du cas d'utilisation global :

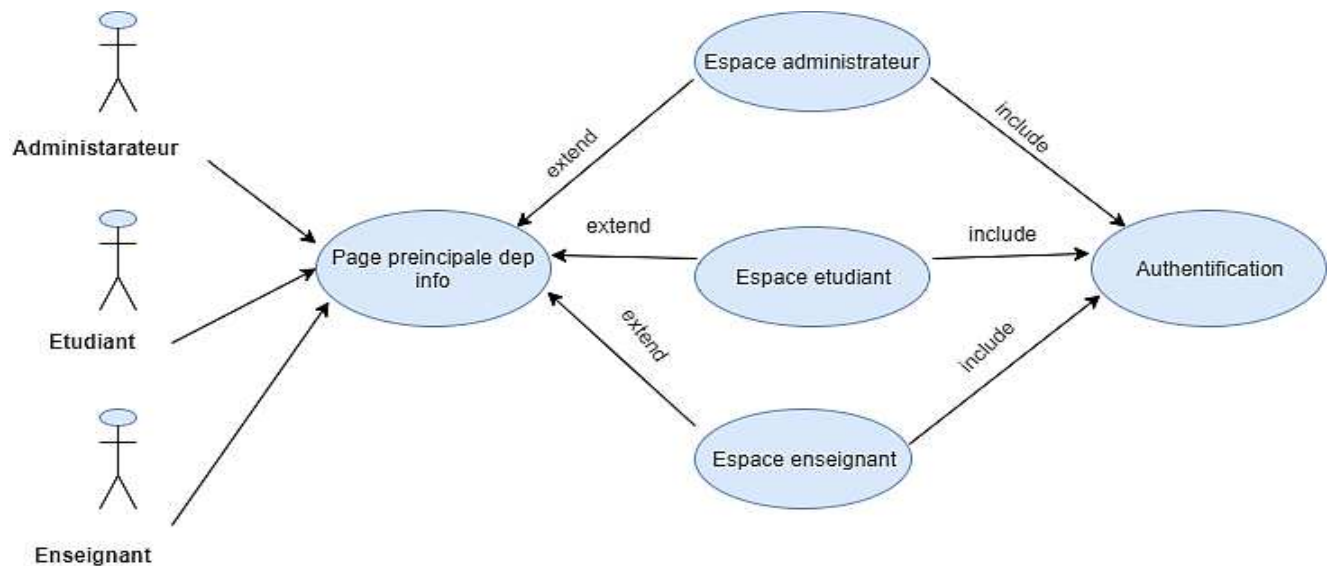


Figure 8 : diagramme du cas d'utilisation global

A.2 diagramme de cas d'utilisation « espace administrateur »



Figure 9 : diagramme de cas d'utilisation « espace administrateur »

A.3. Diagramme de cas d'utilisation « Espace Etudiant »

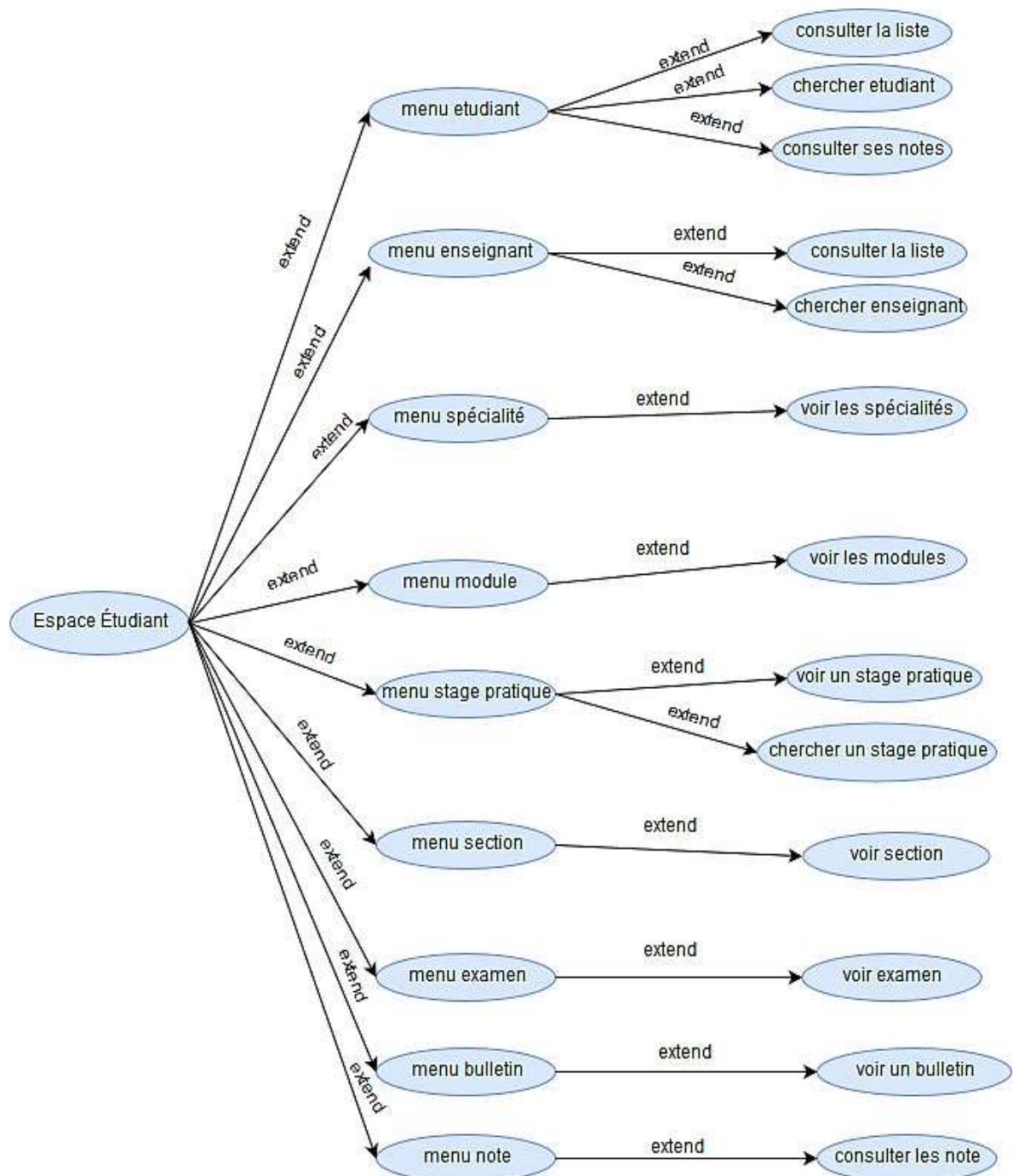


Figure 10 : Diagramme de cas d'utilisation « Espace Etudiant »

A.4. Diagramme de cas d'utilisation « Espace enseignant »

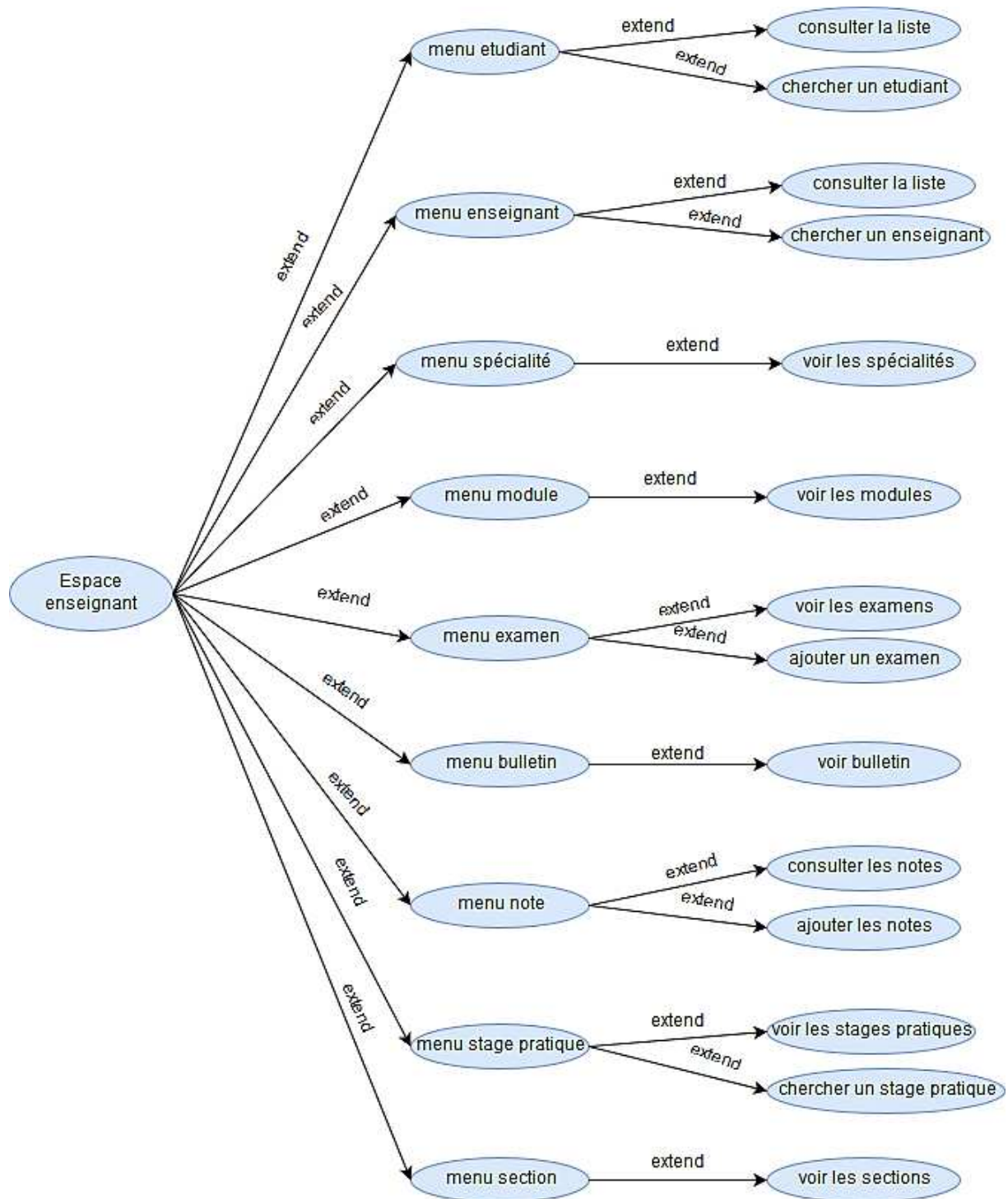


Figure 11 : Diagramme de cas d'utilisation « Espace enseignant »

B. Diagrammes de séquences :

Après avoir décrit les cas d'utilisation avec des scénarios, nous allons traduire ces derniers en diagrammes de séquences et en diagrammes d'activités.

- **Un diagramme de séquences :** indique l'interaction entre plusieurs partenaires de communication, également appelés lignes de vie. Les principales informations contenues dans les diagrammes de séquences sont les messages échangés entre les lignes de vie. Un diagramme de séquences met toujours l'accent sur l'ordre chronologique des messages.

- **objet interface :**

Représente l'interface entre l'acteur et le système tel que les pages web ou écrans de saisie, c'est une description des opérations visibles.

- **objet entité :**

Représente les concepts métier. Il est très souvent persistant, est décrite dans un cas d'utilisation et se trouve dans un autre cas d'utilisation tel que operateur, enregistrement,...

- **objet contrôle :**

C'est un objet actif, tel qu'il possède un flot de contrôle .un objet actif peut activer un objet passif pour le temps d'une opération, en lui envoyant un message.il dirige les activités des entités et interfaces.ces objets sont obtenus en extrayant les verbes des cas d'utilisations.

B.1.diagramme de séquence du cas d'utilisation « ajouter examen »

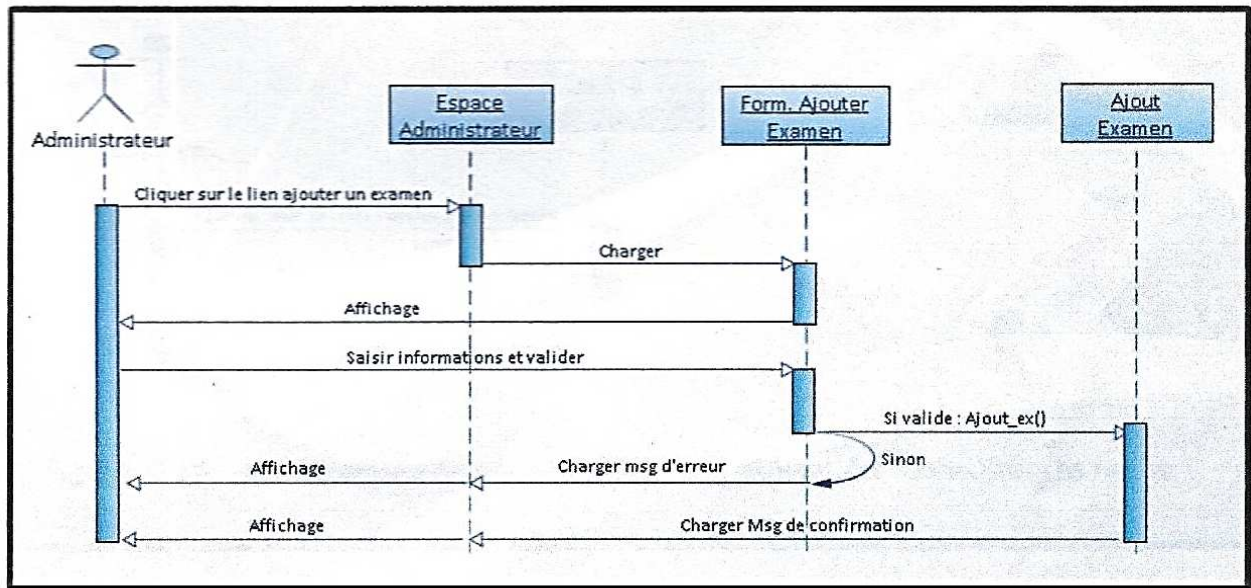


Figure 12 : diagramme de séquence du cas d'utilisation « ajouter examen »

B.2.diagramme de séquence du cas d'utilisation « Ajouter étudiant »

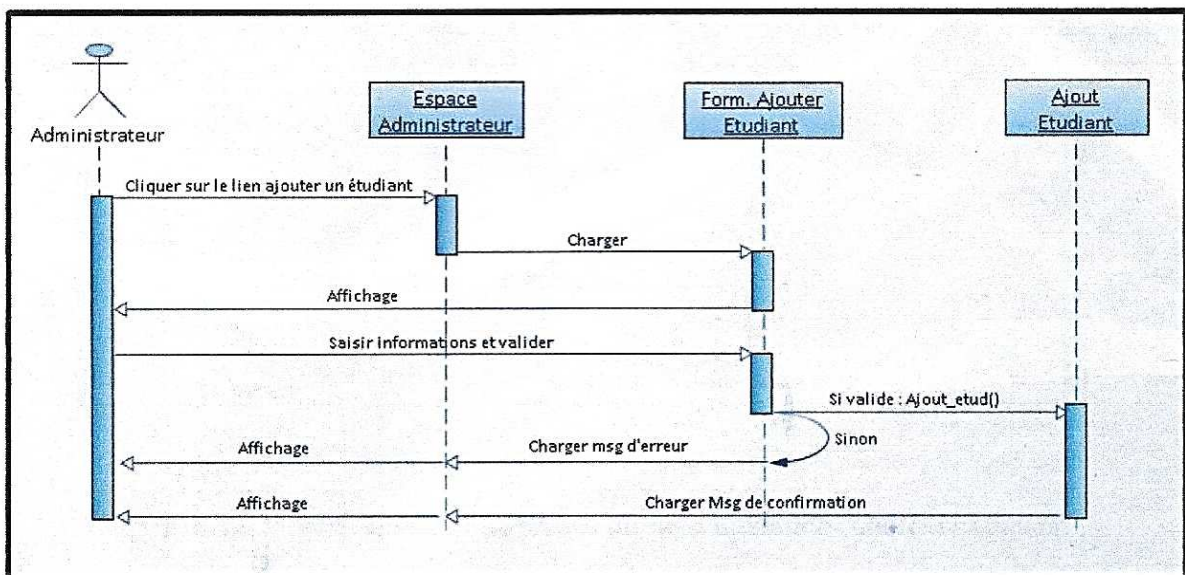


Figure 13 : diagramme de séquence du cas d'utilisation « Ajouter étudiant »

B.3.diagramme de séquence du cas d'utilisation « chercher étudiant »

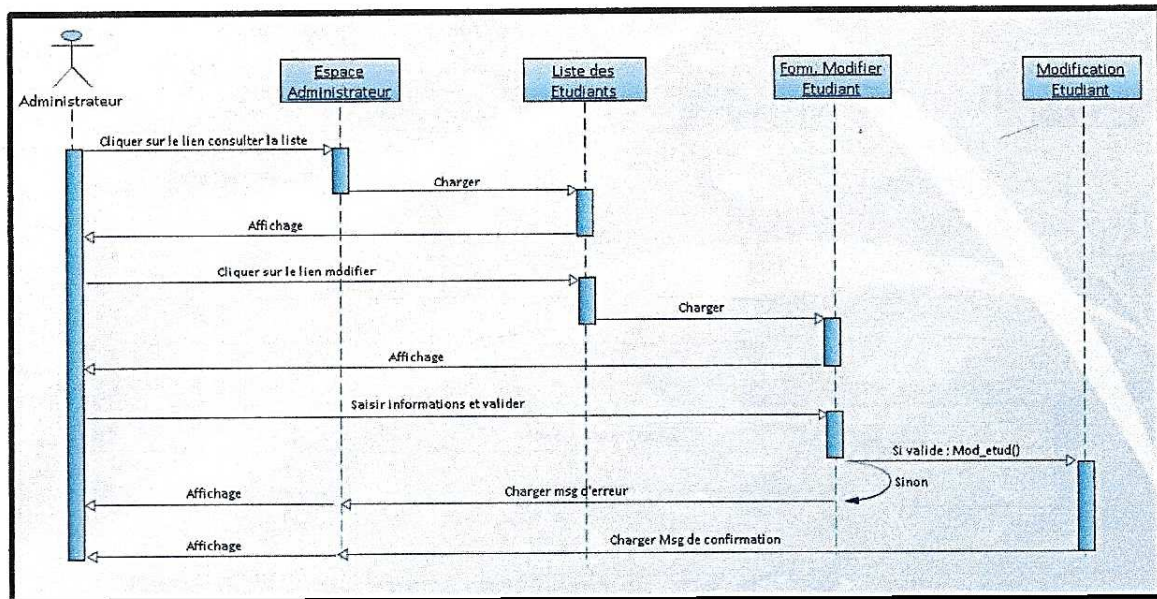


Figure 14 : diagramme de séquence du cas d'utilisation « chercher étudiant »

B.4.diagramme de séquence du cas d'utilisation « ajouter module »

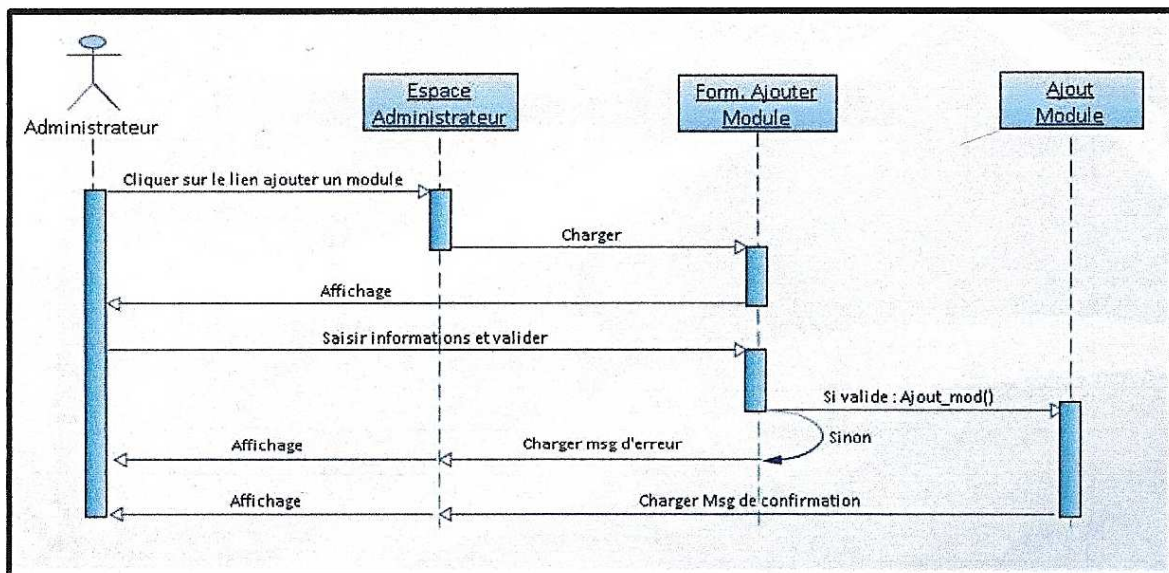


Figure 15 : diagramme de séquence du cas d'utilisation « ajouter module »

C. Diagramme de classes :

Dans cette partie, nous allons passer à la modélisation de l'aspect statique de notre application, c'est-à-dire nous allons modéliser l'intérieur de notre système. Pour ce faire, nous allons utiliser les diagrammes de classes.

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet. Alors que les diagrammes précédents (diagrammes de séquences et diagrammes d'activités) montrent le système du point de vue dynamique, le diagramme de classe en montre la structure interne. Il contient principalement des classes reliées par des associations et chaque classe contient des attributs et des opérations.

Après avoir élaboré les diagrammes de séquences et les diagrammes d'activités pour les cas d'utilisation déjà décrits, nous allons élaborer des diagrammes de classes généraux puis détaillés pour chacun d'eux.

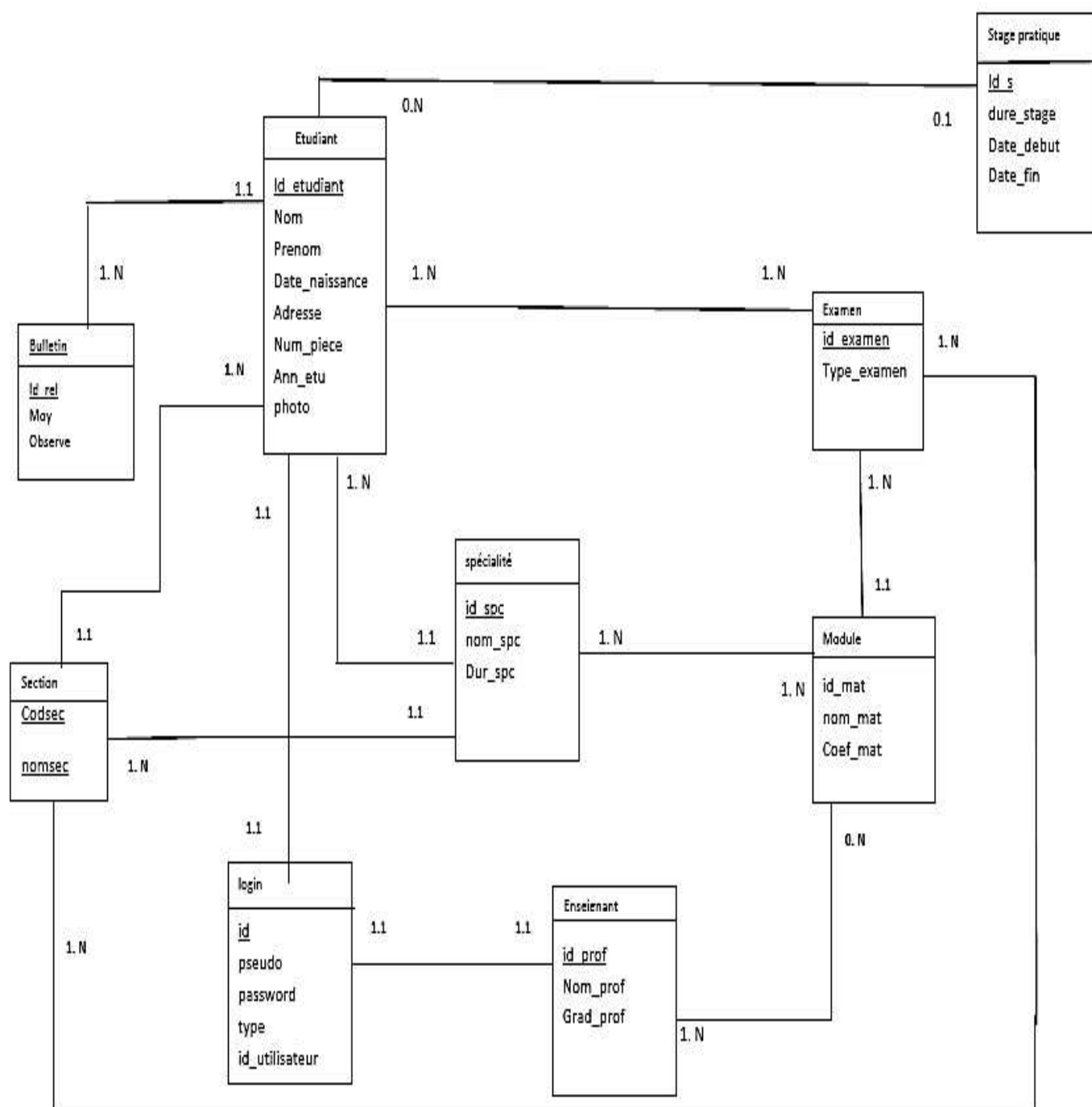


Figure 16 : diagramme de classe global

3.1. Le modèle logique :

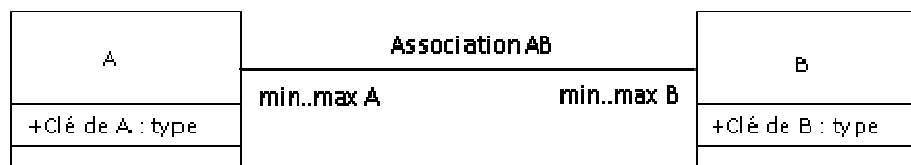
Afin de d'assurer la cohérence de la base de données lors du passage du modèle conceptuel (le digramme de classes) au modèle logique, un certain nombre de règles doivent être respectées. Ces règles sont les suivantes :

A- Les classes : La règle est assez simple, « chaque classe devient une relation ».

Chaque attribut de la classe devient un attribut de la relation. Concernant la clé, soit il existe une clé pour la classe qui deviendra la clé de la relation, ou bien, on ajoute une clé artificielle à la relation.

B- Traduire les associations : L'objectif est de mémoriser les liens entre les objets des classes reliées par l'association. La solution à cette mémorisation dépend de la cardinalité de l'association, plus exactement de ses maximums.

Soient les deux classes A et B suivantes, qui sont reliées par une association AB :



Le tableau suivant examine les cas possibles concernant le maximum des cardinalités de A et de B

MaxB \ MaxA	1	>1
1	Ajouter la clé de B dans la relation de A comme attribut ou Ajouter la clé de A dans la relation de B comme attribut.	Ajouter la clé de B dans la relation de A comme attribut.
>1	Ajouter la clé de A dans la relation de B comme attribut.	Créer une relation AB ayant comme attribut la clé de A et la clé de B.

En se basant sur les règles citées précédemment pour le passage du modèle

conceptuel au modèle logique, nous obtenons les relations suivantes :

Etudiant(id_etudiant, id_spc*, id_rel*,codsec*, Nom, Prénom, Date_naissance, Adresse, Num_piece+, Ann_etu, photo).

Enseignant (id_prof, Nom_prof, Grad_prof).

Passer un examen (id_passer, id_etudiant, id_examen)

Avoir (id_avoir, id_spc, id_mat)

Section (Codsec, nomsec).

Examens (id_examen, Type_examen).

Avoir2 (id_avoir2, codsec, id_examen)

Stage pratique (id_s, duree_stage, Date_debut,Date_fin),

Bulletins (id_rel,Moy, Observe).

Spécialité (id_spc, nom_spc, Dur_spc).

Module (id_mat, id_examen*, nom_mat, Coef_mat).

Enseigne (id_enseigne, id_prof, id_mat)

Login (id,pseudo, password, type, id_utilisateur)

Table étudiant

<i>Champ</i>	<i>Signification</i>	<i>Type</i>	<i>Clé</i>
Id_etudiant	Code etudiant	int(11)	PRIMAIRE
Nom	Nom de l'étudiant	varchar(25)	
Prenom	Prénom de l'étudiant	varchar(25)	
Date	Date de naissance	text	
Adresse	Adresse perssonel	text	
Num_piece	Numéro de la pièce d'identité	int(11)	
Ann_etu	question	text	
Nb_rep	Nombre de réponse	int(11)	
Photo	Photo d'identité	text	
Id_spc	Code section	Int(11)	SECONDAIRE
Id_rel	Code relevé de notes	Int(11)	SECONDAIRE
Codsec	Code section	Int(11)	SECONDAIRE

Table enseignant

<i>Champ</i>	<i>Signification</i>	<i>Type</i>	<i>Clé</i>
Id_prof	Code enseignant	int(11)	PRIMAIRE
Nom_prof	Nom de l'enseignant	varchar(25)	
Prenom_prof	Prenom de l'enseignant	varchar(25)	
Grad_prof	Grade de l'enseignant	text	

Table passer examen

<i>Champ</i>	<i>Signification</i>	<i>Type</i>	<i>Clé</i>
Id_passer	Code passer examen	int(11)	PRIMAIRE
Id_etudiant	Code etudiant	int(11)	SECONDAIRE
Id_examen	Code examen	int(11)	SECONDAIRE

Table avoir

Champ	Signification	Type	Clé
<u>Id_avoir</u>	Code avoir	int(11)	PRIMAIRE
Id_spc	Code spécialité	int(11)	SECONDAIRE
Id_mat	Code module	int(11)	SECONDAIRE

Table section

Champ	Signification	Type	Clé
Codsec	Code section	int(11)	PRIMAIRE
<u>Nomsec</u>	Nom de la section	varchar(15)	

Table examen

Champ	Signification	Type	Clé
Id_examen	Code examen	int(11)	PRIMAIRE
Type_examen	Type de l'examen	varchar(15)	

Table avoir2

Champ	Signification	Type	Clé
<u>Id_avoir2</u>	Code avoir2	int(11)	PRIMAIRE
Codsec	Code section	int(11)	SECONDAIRE
Id_examen	Code examen	int(11)	SECONDAIRE

Table stage pratique

Champ	Signification	Type	Clé
Id_s	Code enseignant	int(11)	PRIMAIRE
Duree_stage,	Nom de l'enseignant	varchar(25)	
Date_debut	Prenom de l'enseignant	varchar(25)	
Date_fin	Grade de l'enseignant	text	

Table bulletins

<i>Champ</i>	<i>Signification</i>	<i>Type</i>	<i>Clé</i>
Id_rel	Code bulletin	int(11)	PRIMAIRE
Moy	Moyenne de l'étudiant	Float	
Observe	Observation	text	

Table spécialité

<i>Champ</i>	<i>Signification</i>	<i>Type</i>	<i>Clé</i>
Id_spc	Code spécialité	int(11)	PRIMAIRE
Nom_spc	Nom de la spécialité	Varchar(25)	
Dur_spc	Durée de spécialité	text	

Table module

<i>Champ</i>	<i>Signification</i>	<i>Type</i>	<i>Clé</i>
Id_mat	Code module	int(11)	PRIMAIRE
Id_examen	Code examen	Int(11)	SECONDAIRE
Nom_mat,	Nom du module	varchar(25)	
Coef_mat	Coefficients module	Int(5)	

Table enseigne

<i>Champ</i>	<i>Signification</i>	<i>Type</i>	<i>Clé</i>
<u>Id_enseigne</u>	Code enseigne	int(11)	PRIMAIRE
Id_prof	Code enseignant	int(11)	SECONDAIRE
Id_mat	Code module	int(11)	SECONDAIRE

Table login :

<i>Champ</i>	<i>Signification</i>	<i>Type</i>	<i>Clé</i>
<u>id</u>	Code enseigne	int(11)	PRIMAIRE
pseudo	Nom utilisateur	Varchar(20)	
password	Mot de passe	Varchar(20)	
type	le type de l'utilisateur	text	
id_utilisateur	Le code de l'utilisateur	int(11)	

4. Conclusion :

Ce chapitre est consacré à l'analyse et à la conception de l'application avec le langage UML. Nous avons présenté le processus de conception de notre application en deux niveaux, le niveau applicatif et le niveau de données. En premier lieu, nous avons commencé l'analyse et la conception par le niveau applicatif qui concerne les fonctionnalités et les traitements de l'application, ensuite nous sommes passés au niveau données qui nous a permis d'avoir le modèle logique de la base de données.

Le chapitre suivant sera donc la traduction de cette conception en un ensemble de scripts, en ce qui concerne partie applicative, et en une base de données physique pour la partie données.

CHAPITRE

Réalisation & Mise en œuvre

1. Introduction

On s'intéressera dans ce chapitre à l'implémentation de future application, et cela on décrivant d'abord l'environnement et les outils de développement de notre application, en suit on présentant un exemple des différentes interfaces de cette dernière ainsi que les fonctionnalités qu'elles regroupent.

2. Architecture de l'application

L'application comporte trois niveaux, analogue à l'architecture trois tiers du modèle client serveur. L'architecture de notre application peut être représentée graphiquement comme suit :

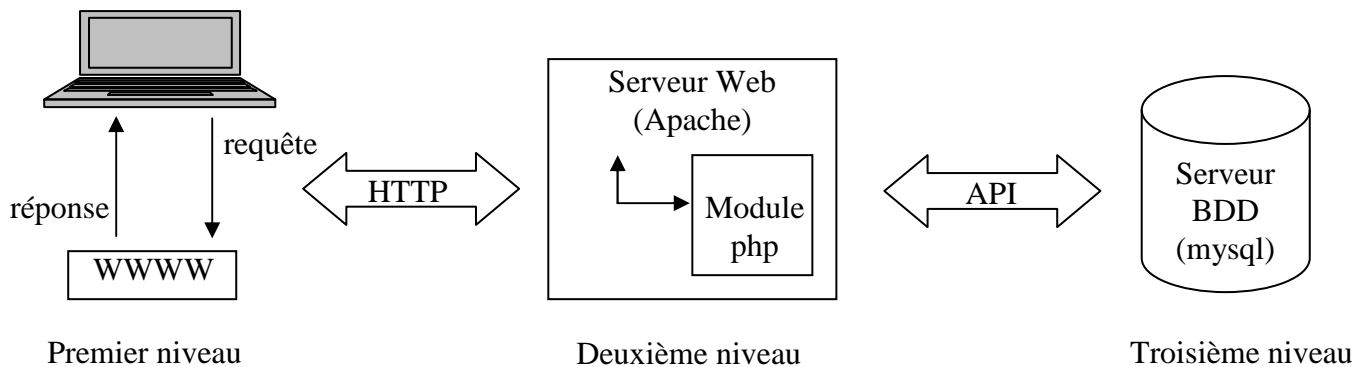


Figure 17 : l'architecture du déploiement de l'application

Premier niveau : il comprend le navigateur qui interprète et affiche les interfaces utilisateur relatives aux différents services de l'application ;

Deuxième niveau : le deuxième niveau est constitué d'un serveur Web Apache doté de module PHP (l'interpréteur PHP est installé comme module Apache) ;

Troisième niveau : le troisième niveau est composé d'un serveur de base de données (MYSQL) performant souple et disposant d'un jeu de commande SQL large.

3. L'environnement de développement

3.1 Langages de programmation

Pour satisfaire les besoins de l'application à réaliser, nous avons utilisé les langages suivants :

- ✓ **Le langage HTML** : pour la création des pages statiques de l'application qui constitueront l'interface utilisateur.
- ✓ **Le langage de requête SQL** : pour l'élaboration des requête d'interrogation de la base de données.
- ✓ Le langage de script côté client **java script** : pour les contrôles des informations au niveau client.
- ✓ **Le langage de script PHP** : pour la partie dynamique du site. Il constitue le langage pilier de l'application, c'est-à-dire, que c'est sur lui que reposent le dynamisme de l'application, c'est la raison pour laquelle nous allons mieux le détailler dans ce qui suit :

3.1.1 PHP [15]

PHP (Hypertext Preprocessor , Pré processeur Hypertexte PHP) est un langage de script côté serveur, c'est à dire que les scripts sont exécutés avant que la page ne soit envoyée au navigateur, Ses principaux avantages sont :

- PHP est facile à utiliser, il permet de créer avec un minimum d'efforts des pages Web dynamiques destinées aux applications Internet multimédia et de E-commerce.
- PHP est multi plate forme, il existe pour les différentes versions de Windows, Unix et Linux, ainsi que pour de nombreux serveurs Web dont APACHE et IIS.
- PHP est libre, donc « OPEN SOURCE », on peut ajouter de nouvelles fonctionnalités sans attendre une nouvelle version.
- PHP utilise le moteur de scripts Zend, qui améliore la rapidité de traitement

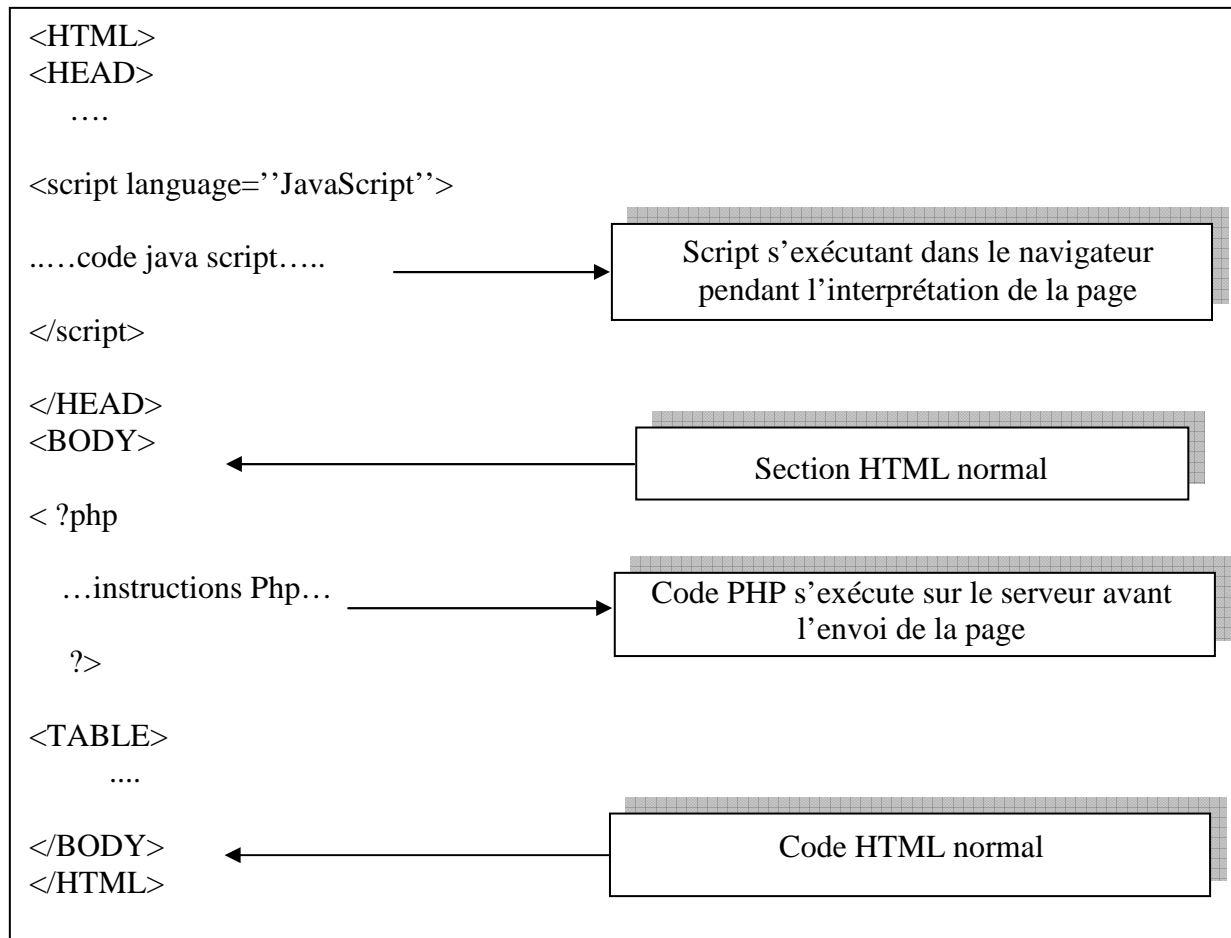
- PHP a été conçu pour fonctionner sur le Web, la connexion et l'interrogation d'une base de données sont extrêmement simple (peuvent être accomplies en deux ou trois lignes de code).
- La plus grande qualité et le plus important avantage du langage PHP est le support d'un grand nombre de bases de données (plus de 20). PHP possède des fonctions ODBC, permettant de s'interfacer de manière conviviale à toute base de données ODBC.
- Avec plus de 1200 fonctions utilisables dans des applications très variées, PHP couvre pratiquement tous les domaines en rapport avec les applications Web.

3.1.2 Structure d'une page PHP [15]

Le code PHP fait partie intégrante de la page HTML, il est inclus entre :

- `< ? PHPinstruction ?>`
- `< ? phpinstruction ?>`
- `< ?instruction ?>`
- `< %instruction %>`
- `< script langage= « PHP » >`
.....instruction.....
`< / script>`

Ce code permet au navigateur de passer en mode PHP.



- Les autres langages de script coté client comme le Java script sont exécutées par le navigateur, alors que le code Php est exécuté sur le serveur. Si on a un script Php sur un serveur, le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat.

3.1.3 Fonctionnement de PHP [15]

Le serveur Web reconnaît à l'extension des fichiers, différente de celle des pages HTML simples, si le document appelé par le client comporte du code PHP.

- Le serveur Web lance l'interpréteur PHP ;
- L'interpréteur PHP traduit le document demandé et exécute le code source de la page ;
- Les commandes figurant dans la page sont interprétées et le résultat prend la forme d'une page HTML publiée à la place du code source dans le même document ;

- La page modifiée est envoyée au client pour y être affichée par le navigateur.

De cette façon la page Web est créée dynamiquement, c'est à dire au moment même où le client y accède. Cela permet donc de modifier la page avant qu'elle ne soit envoyée au client en fonction de dialogue avec l'utilisateur.

3.2 Serveur Web Apache

L'ancêtre d'apache est le serveur libre développé par le (National Center for Supercomputing Applications) de l'université de l'Illinois. L'évolution de ce serveur s'est arrêtée lorsque le responsable a quitté le NCSA en 1994, les utilisateurs ont continué à corriger les bugs et à créer des extensions qu'ils distribuaient sous forme de "patches" (bouts de programmes ajoutés par les utilisateurs des NCSA pour étendre les fonctionnalités d'Apache) d'où le nom "a patchee Server" la version 1.0 de Apache a été disponible le 1 décembre 1995.

Nous choisissons et présentons ce serveur par les avantages qu'il offre et que nous allons détailler ci-après :

- c'est un serveur gratuit (peut être télécharger à partir du site du groupe Apache à l'adresse '<http://www.apache.org>').
- un niveau élevé de performances pour des besoins matériels modeste.
- extensible, modulaire et configurable.
- robuste.
- très portable contrairement à IIS (Internet Information Services) de Microsoft qui tourne seulement sous Windows, le serveur Web apache dispose d'une version pour chaque plate forme (Linux, Windows,...).

3.3 Serveur MySQL [15]

3.3.1 MySQL

MySQL un véritable serveur de bases de données SQL multiutilisateur et multi traitement. Cela permet d'établir des connexions rapides et d'utiliser la même mémoire cache pour plusieurs requêtes.

MYSQL est une configuration Client –Serveur ce qui consiste en un serveur démon Mysql, différents programmes clients et des librairies.

MYSQL est basé sur une bibliothèque de gestion de donnée éprouvée depuis de nombreuses années et faisant appel à des index d'arbres binaires. Grâce à cela, le cœur du système peut afficher une performance remarquable, tout particulièrement dans les accès indexés.

Les principaux objectifs de MYSQL sont la rapidité, la robustesse, et la facilité d'utilisation.

3.3.2 Fonctionnalités de MySQL

La liste suivante décrit quelques fonctionnalités importantes de MYSQL :

- **Multitraitement** : MySQL est multitraitement en utilisant les threads du noyau. Il peut utiliser plusieurs CPU
- **Langues** : Le serveur peut fournir au client les messages d'erreurs en plusieurs langues.
- **APIs** : Les applications de bases de données MySQL peuvent être écrites en C, C++, Eiffel, JAVA, PERL, PHP, PYTHON et TCL.
- **Multi plateformes** : Prise en charge de plus de 20 plates-formes de système d'exploitation Win95/98, NT UNIX et LINUX.
- **Tables** : MySQL stocke chaque table sous forme de fichier distinct dans le répertoire de la base de données. La taille maximale d'une table comprise entre 4Go est la taille maximale de fichier acceptée par le système d'exploitation. Le mélange des tables de différentes bases de données est supporté dans une même requête.
- **Système de droits flexibles et sécurisé** : Système de droits flexible et sécurisé de mots de passe, et qui autorise une vérification faite sur l'hôte : le serveur détermine l'identité du client grâce à l'hôte depuis lequel il se connecte et le nom d'utilisateur qu'il spécifie. Puis, l'associer avec les droits d'utilisation des commandes :select, insert, update et delete sur cette base (le serveur vérifie chaque requête émise pour voir

si elle est autorisée). Les mots de passe sont sécurisés depuis que la gestion des mots de passe est cryptée entre le client et le serveur.

- **Rapide, Robuste et Convivial :** MySQL est de trois à quatre fois plus rapide que la plupart des autres bases de données commerciales : les fonctions SQL sont implémentées à travers des classes de bibliothèques extrêmement optimisées. En général, il n'y a plus d'allocation mémoire après une requête d'initialisation.

Les développeurs de MySQL (TCX) utilisent MySQL depuis 1996 dans un environnement de plus de 40 bases de données contenant 10000 tables, dont plus de 500 contiennent plus de 7 millions d'enregistrements. C'est environ 100 Giga Octets de données critiques.

Sa gestion est aisée puisqu'elle ne nécessite pas des compétences poussées en administration de bases de données

- **Economique :** MySQL est un SGBD, « OPEN SOURCE ».

4. Les outils de développement

4.1 wamp server

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs à travers une icon (icône près de l'horloge de Windows).

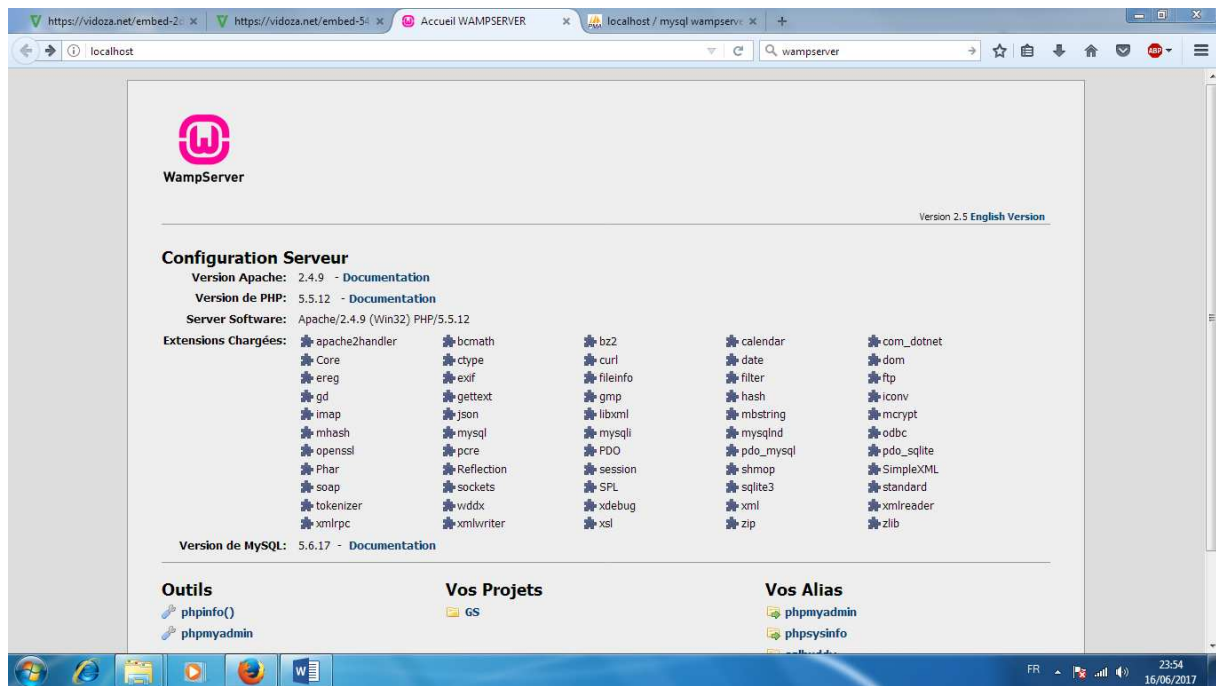


Figure 18 : Page d'accueil Wamp Server

4.2 PhpMyAdmin

PhpMyAdmin est un utilitaire rendant plus conviviale l'administration de base de données MySQL. Il consiste en un ensemble de scripts PHP permettant d'administrer des bases de données MySQL en passant par un navigateur web.

4.2.1 Les fonctions de PhpMyAdmin

- Création et suppression de base de données
- Création, modification, copie et suppression de table
- Edition, ajout et suppression de champ
- Exécution des commandes SQL.
- Création d'index
- Chargement de fichier dans des tables

4.2.2 Utilisation de PhpMyAdmin

Pour accéder à PhpMyAdmin, il faut d'abord vérifier qu'Easyphp.exe est lancé et que le serveur fonctionne, après on peut accéder à partir de "l'administration". Pour ouvrir l'administration il suffit de faire un clic droit sur l'icône et sélectionner "administration". Une page Web apparaît, au milieu de celle-ci il y a un bouton PhpMyAdmin avec un simple clic là-dessus, la page d'accueil de PhpMyAdmin s'affiche dans la fenêtre du navigateur, accompagner d'un champ de sélection de base de donnée présente sur l'hôte MySQL par défaut.

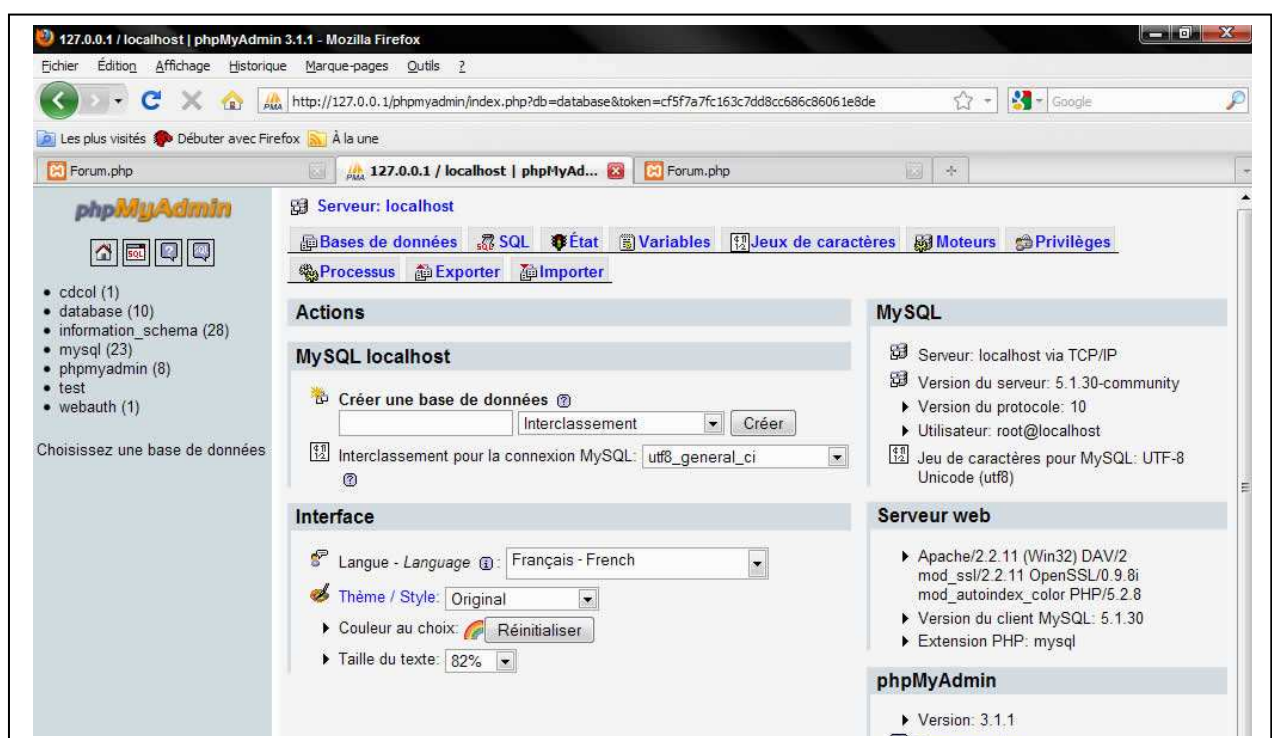


Figure 19 : Administration de MySQL à partir de PhpMyAdmin

Pour afficher le contenu de la base personnelle par exemple, il faut cliquer sur la ligne correspondante à la base de données personnelle dans la partie gauche de la fenêtre du navigateur. Toutes les tables de cette base de données seront affichées :

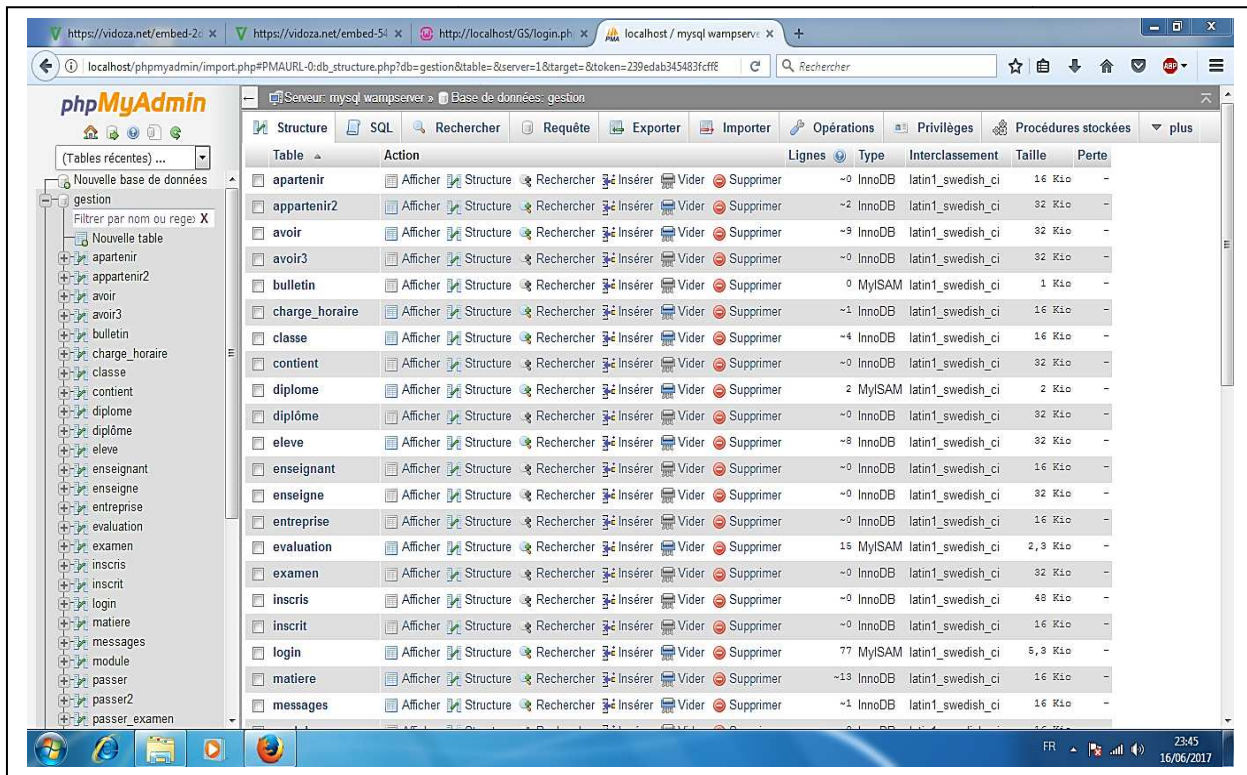


Figure 20 : Accès à la base de données “ database ” à partir de PhpMyAdmin

4.3 Autre outils

- ✓ **Macromedia Dreamweaver 8:** est un éditeur de page HTML, convivial et simple à utiliser.
- ✓ **Macromedia Fireworks 8:** est un éditeur d'image, il a été utilisé pour la création des objets graphiques de notre application tel que les bannières les boutons et le logo.

5. Présentation de quelques interfaces de notre plate-forme :

5.1.1 La page login

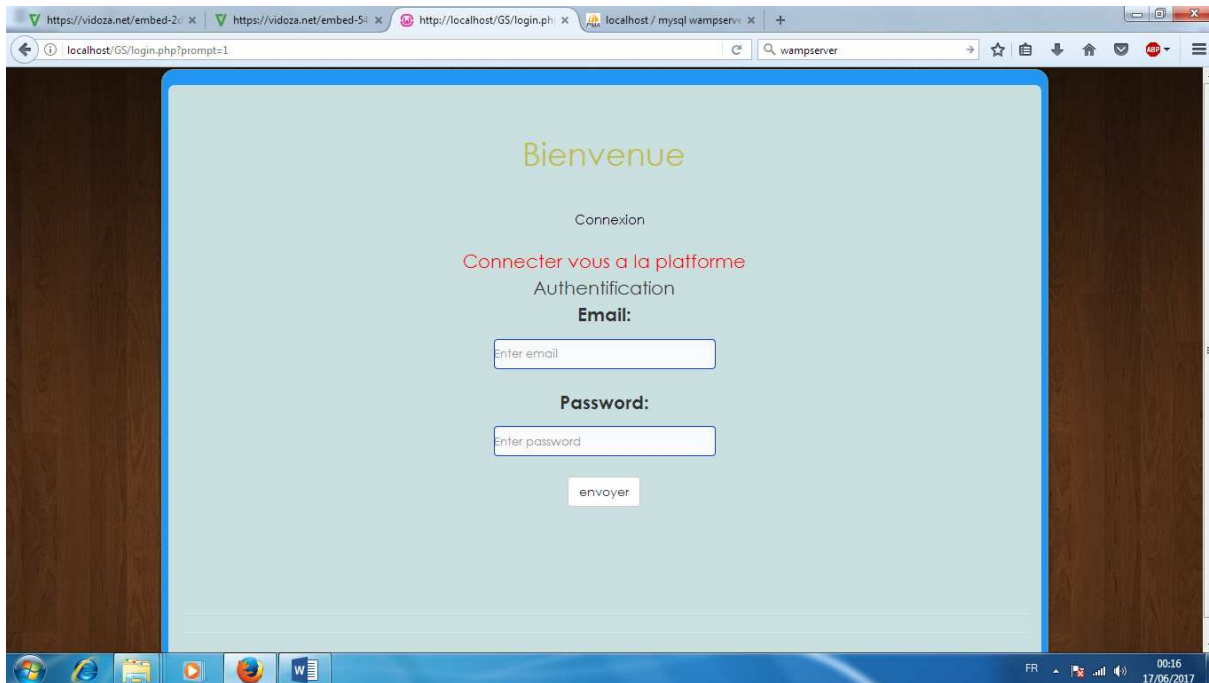


Figure 21 : page login

5.1.2: Interface d'accueil :

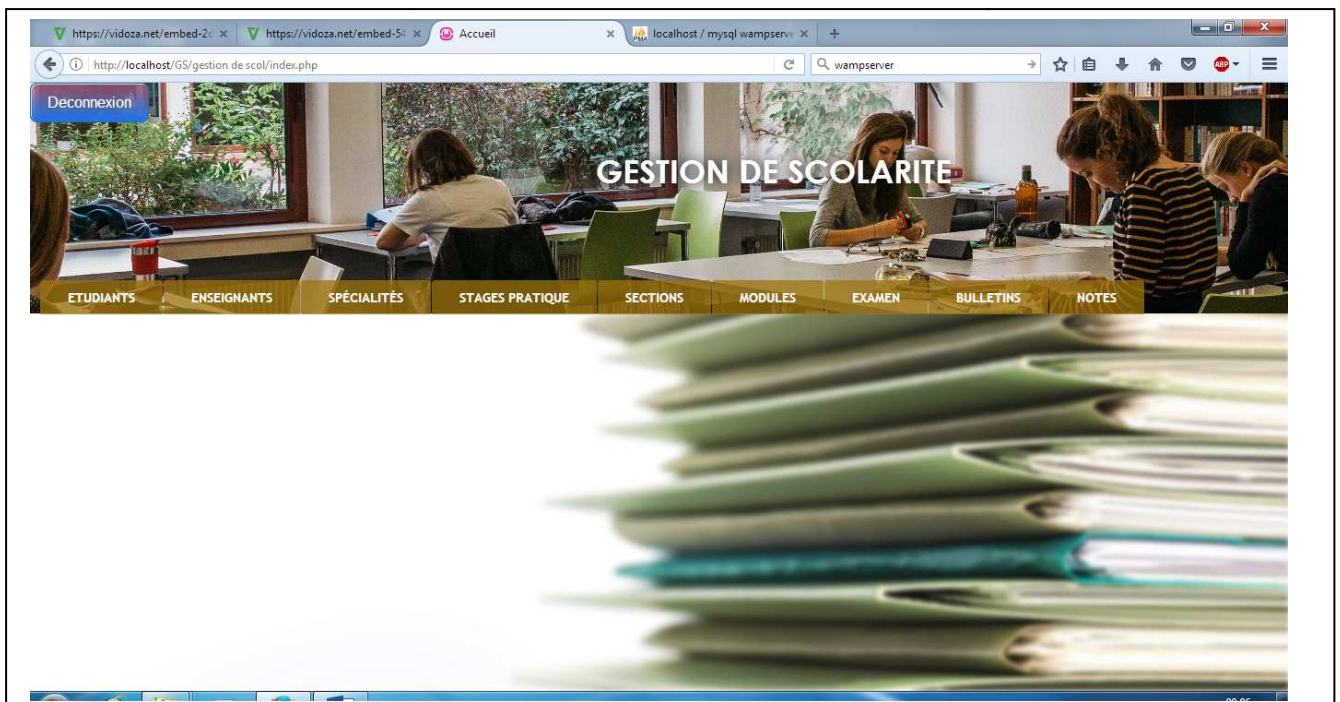


Figure 22 : page Interface d'accueil

5.1.3 : Interface recherche enseignant :

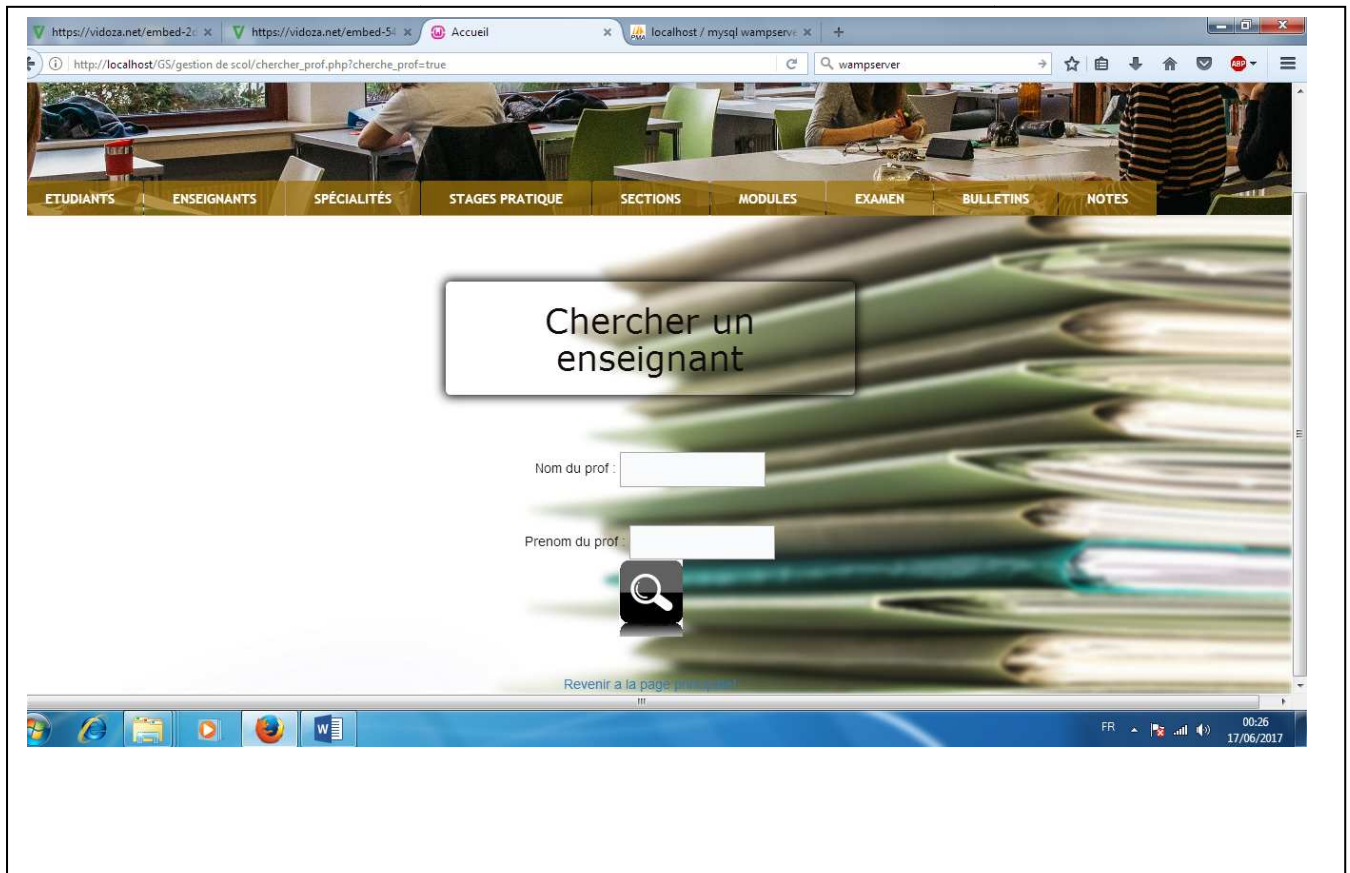


Figure 23 : Interface recherche enseignant

5.1.4 Interface ajouté examens :

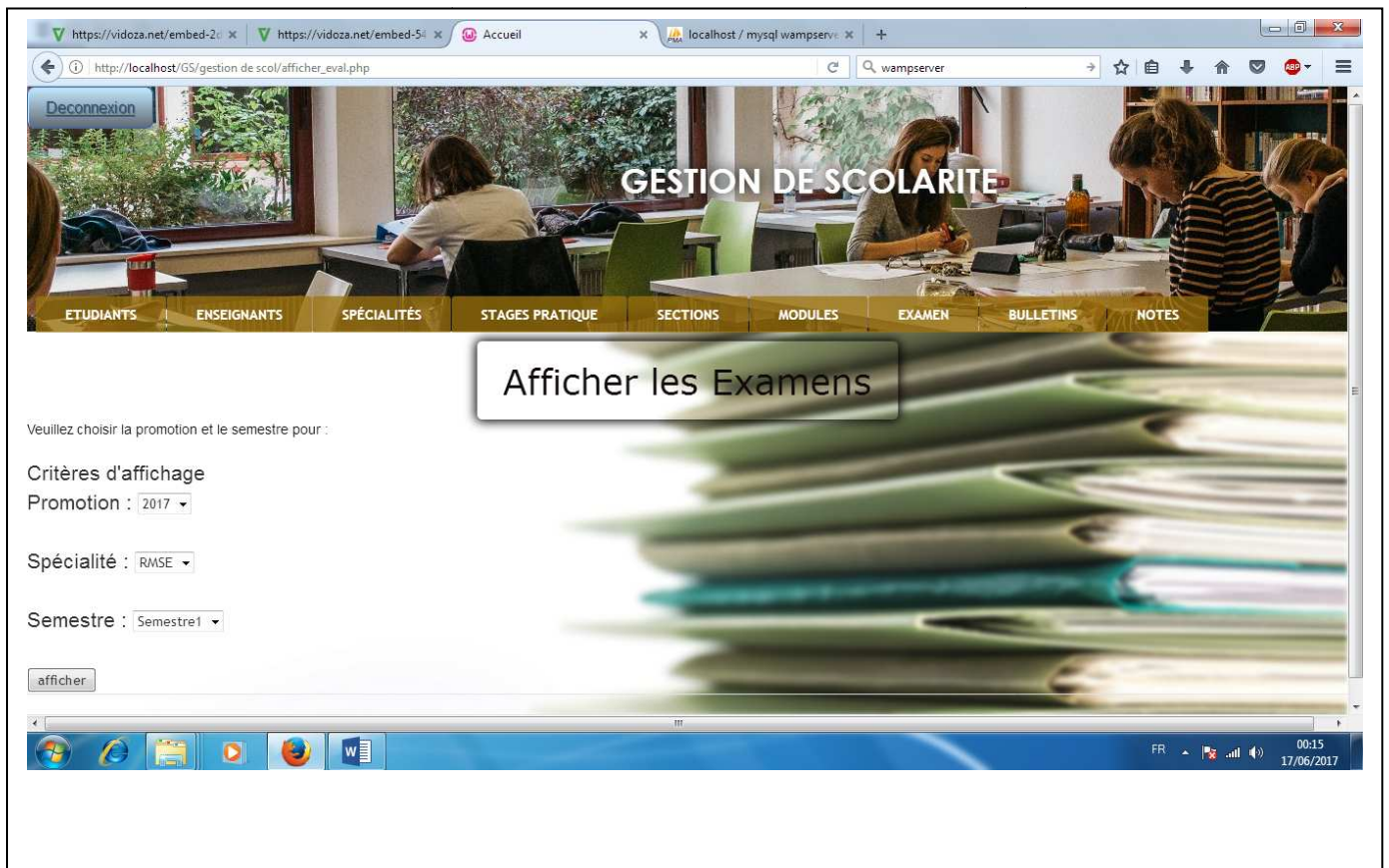


Figure 24 : Interface ajouter examens

5.1.5 : Interface consulter la liste des étudiants :

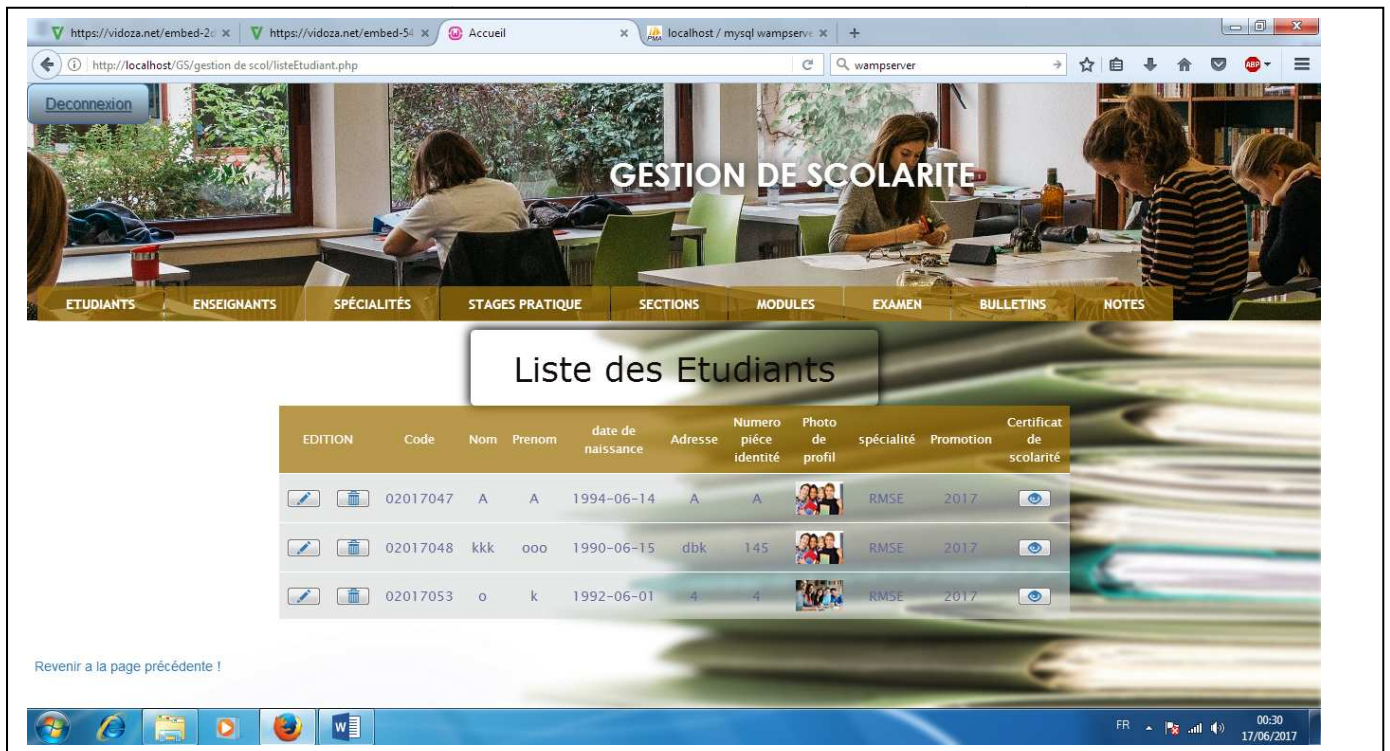


Figure 25 : Interface consulter la liste des étudiants

5.1.6 : Interface ajouter un examen :

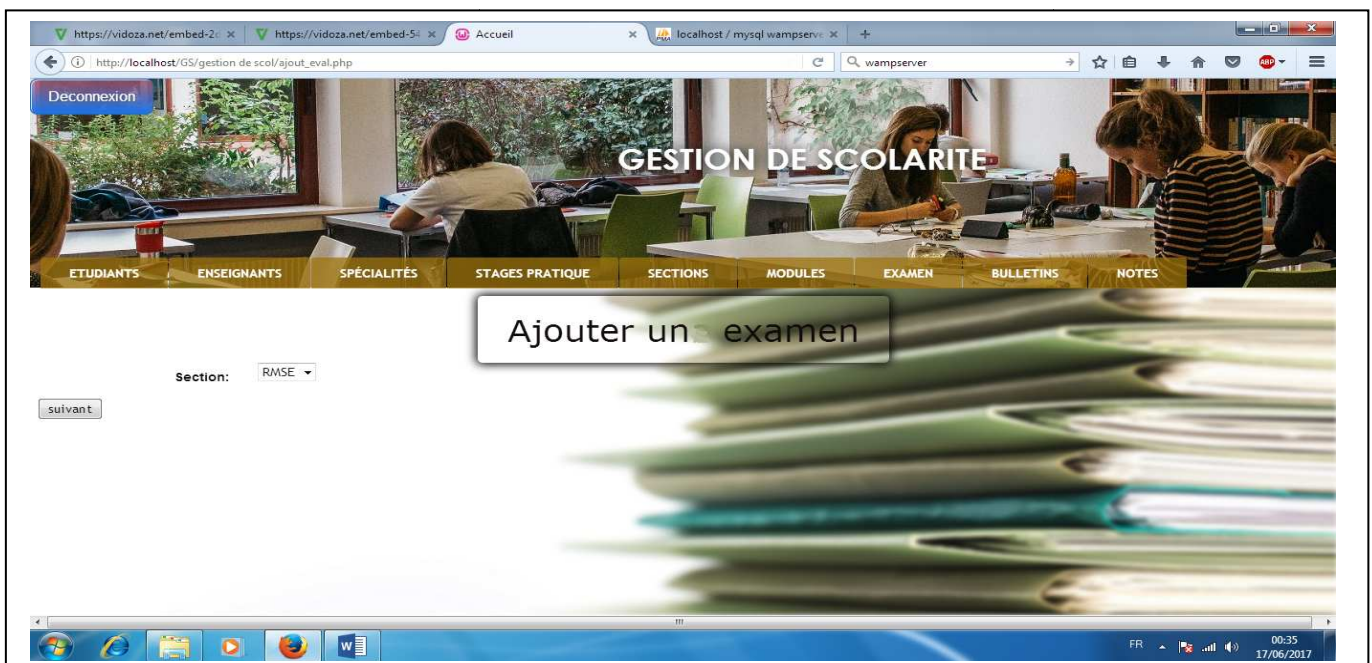


Figure 26 : Interface ajouter un examen

5.1.7 : Interface chercher un stage pratique :

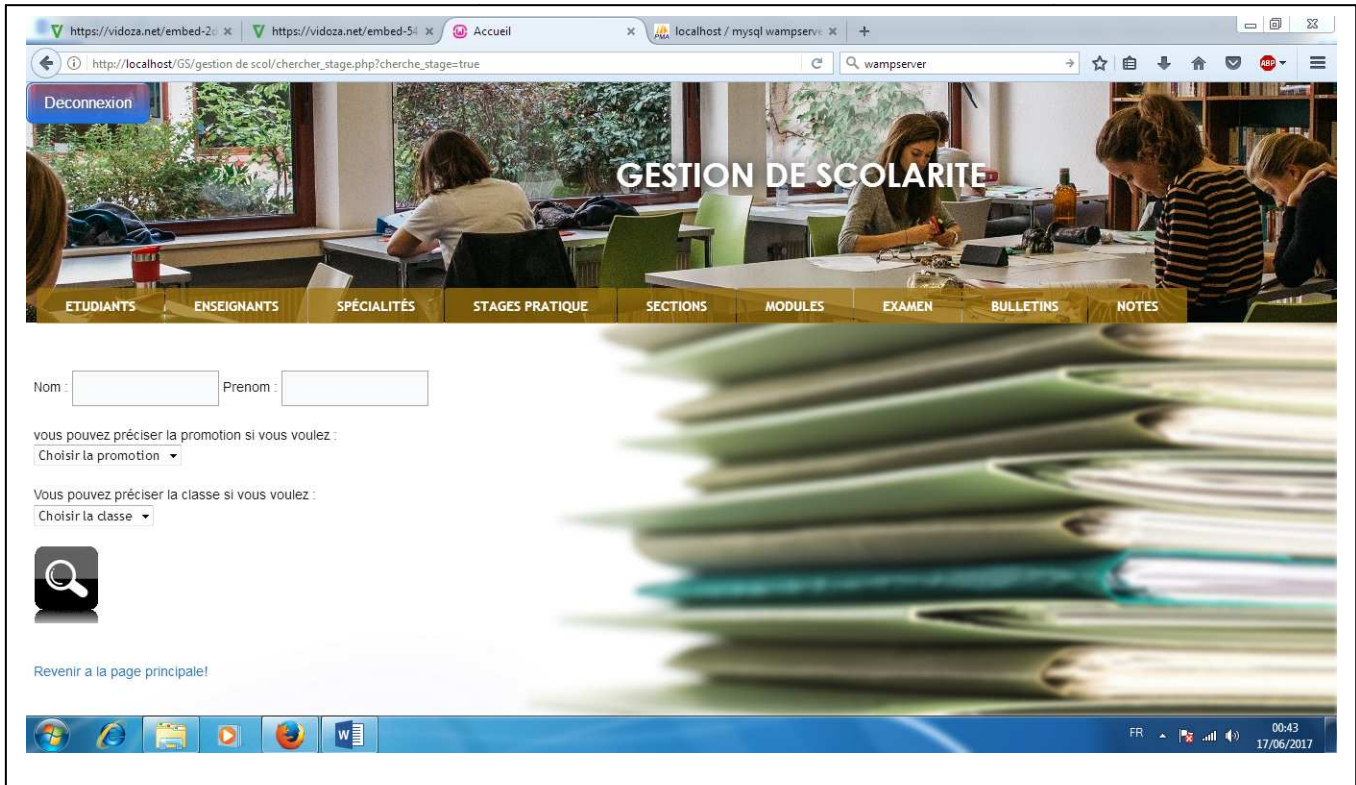


Figure 27 : Interface chercher un stage pratique

4.6 Conclusion

Dans ce chapitre, nous avons présenté en premier lieu l'environnement de développement de notre application puis quelques interfaces essentielles de notre site.

Conclusion

Générale

Conclusion Générale

Toute organisation, quelque soit sa vocation et son caractère, doit se mettre au diapason de la progression technologique et faire face par l'automatisation de ses structures et la formation de son personnel afin d'améliorer son rendement et son service.

L'étude et le développement de ce projet informatique nous ont permis, outre la familiarisation avec le domaine professionnel, d'acquérir de nouvelles compétences sur le plan pratique et technique , à savoir côté application Client/serveur, les outils de développement tels que Macromedia Dreamweaver et l'utilisation du serveur Web Apache, côté bases de données avec à la clé l'usage du SGBD MySQL, côté conception orientée objet faite à l'aide du langage UML et côté réalisation avec l'introduction des langages de programmation tels que XHTML, CSS, PHP, JavaScript et SQL.

L'application à laquelle nous avons abouti, traite la gestion de scolarité qui adopte l'architecture Client/Serveur. Elle offre aux différents utilisateurs plus de souplesse et d'efficacité dans l'exécution de leurs tâches.

Quoique notre application offre plusieurs services et rend aisé la tâche de gestion de scolarité, quelques améliorations et extensions peuvent être apportées. En guise de perspectives, les possibilités d'évolutions de ce projet sont nombreuses. Pour clore ce mémoire, nous espérons avoir répondu aux attentes et aux exigences de l'université, et que l'application informatique réalisée sera d'une grande utilité pour l'Université Mouloud Mammeri de Tizi-Ouzou.

Annexe

La notation UML

I. Description

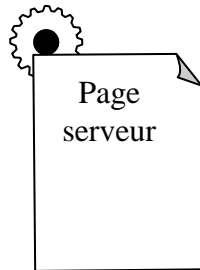
L'extension d'UML pour le Web définit un ensemble de stéréotype, d'étiquettes et de contraintes, qui rend possible la modélisation Web.

II. Stéréotypes

II.1. classe

II.1.1. Page serveur « Server page »

Icône



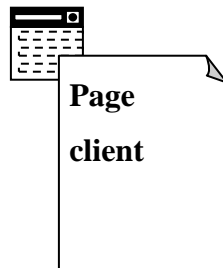
Description : Représente une page Web possédant des scripts qui interagissent avec des ressources serveur telles que les bases de donnée, ces scripts sont exécuté par le serveur.

Contraintes : Les pages serveur ne peuvent avoir de relation qu'avec des objets sur le serveur.

Etiquettes : Moteur de script qui peut être un langage ou le moteur qui doit être utilisé pour exécuter ou interpréter cette page.

II.1.2. Page client « client page »

Icône



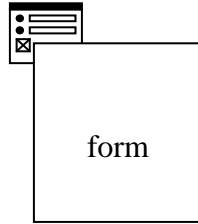
Description : Une instance d'une page client est une page Web formatée en HTML. Les pages clients peuvent contenir des scripts interprétés par les navigateurs lorsque celles-ci sont restituées par ces derniers. Les fonctions des pages clients correspondent aux fonctions des scripts de la page web.

Contrainte : Aucune.

Etiquettes

- Titre (Title) : Titre de la page tel qu'il est affiché par le navigateur.
- Base (Base) : URL de base pour dé-référencer les URL relatives.
- Corps (Body) : ensemble des attributs de la balise <body>, qui définit des caractéristiques par défaut du texte et de l'arrière-plan.

II.1.3. Formulaire

Icône

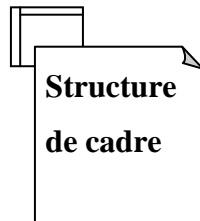
Description : Une classe stéréotypée « form » est un ensemble de champs de saisie faisant partie d'une page client. A une classe formulaire correspond une balise HTML <form>, les attributs de cette classe correspondent aux éléments de saisie d'un formulaire HTML (zone de saisie, zone de texte, boutons d'option.).

Un formulaire n'a pas d'opérations, puisqu'il peut les encapsuler. Toute opération qui interagit avec le formulaire appartient à la page qui la contient.

Contraintes : Aucune.

Etiquettes

GET ou POST – Méthodes utilisées pour soumettre les données à l'URL de l'attribut action de la balise HTML <form>.

II.1.4. structure de cadres « frameset »**Icône**

Description : Une structure de cadres est un conteneur de plusieurs pages Web. La zone d'affichage rectangulaire est divisée en cadres rectangulaires inscrits. A chaque cadre peut être associé un nom unique de cible « Target ».

Le contenu d'un cadre peut être une page Web ou une structure de cadre.

Une classe stéréotypée « frameset » est directement associée à une structure de cadre de page Web par la balise HTML <frameset>.

Une structure de cadre est une page client qui peut posséder des opérations et des attributs.

Contraintes : Aucune

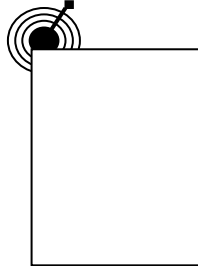
Etiquettes

Rangées (rows) : valeur de l'attribut rows de la balise HTML <framset>. C'est une chaîne de pourcentages séparés par des virgules, définissant les hauteurs relatives des cadres.

Colonnes (cols) : valeur de l'attribut cols de la balise HTML <frameset>. C'est une chaîne de pourcentages séparés par des virgules, définissant les largeurs des cadres.

II.1.5. cible « target »

Icône



Description : Une cible est une zone nommée dans la fenêtre du navigateur dans laquelle des pages Web peuvent être affichées. Le nom de la classe stéréotypée est celui de la cible. Habituellement, une cible est le cadre d'une structure de cadre définie dans une fenêtre ; cependant, une cible peut être une toute nouvelle instance de navigateur : une fenêtre. Une association « targeted link » spécifie la cible où une page Web doit être affichée.

Contraintes : Pour chaque client du système le nom de la cible doit être unique. Par conséquent sur un même client, il ne peut exister qu'une seule instance d'une même cible.

Etiquettes : Aucune.

II.1.6. Objet Java Script « Java Script object »

Icône : Aucune.

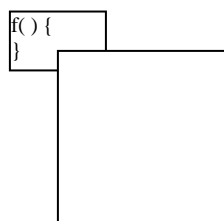
Description : Sur un navigateur compatible Java Script, il est possible de simuler des objets personnalisés à l'aide de fonctions Java Script. Les objets Java Script ne peuvent exister que dans le contexte de pages client.

Contraintes : Aucune.

Etiquettes : Aucune.

II.1.7. Objet ScriptClient « ClientScript object »

Icône



Description : Un Objet ScriptClient est un ensemble qui regroupe des scripts client particuliers dans un fichier. Lequel est inclus dans une requête distincte du navigateur client. Ces objets regroupent souvent des lots de fonctions couramment utilisées au travers d'une application ou d'une entreprise.

Contraintes : Aucune.

Etiquettes : Aucune.

II.2. association

II.2.1. Lien « link »

Icône : Aucune

Description : Un lien est un pointeur d'une page client vers une autre page. Dans un diagramme de classes, un lien est une association entre une page client et une autre page client ou une page serveur. A un lien correspond une balise ancre HTML.

Contraintes : Aucune.

Etiquettes : Paramètres (paramètres) : liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

II.2.2. Lien cible « targeted link »

Description : Similaire à une association lien. Un lien cible est un lien dont la page associée est affichée dans une cible. A un lien cible correspond une balise ancre HTML, dont l'attribut target prend la valeur de la cible.

Icône : Aucune.

Contraintes : Aucune.

Etiquettes : Paramètres (Paramètres) : liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

Nom de la cible (target name) : nom de la cible ou la page vers laquelle pointe le lien qui doit être affichée.

II.2.3. Contenu de cadre « frame content »

Icône : Aucune.

Description : Une association contenue de cadre est une association d'agrégation qui traduit l'appartenance d'une page ou d'une cible à un cadre.

Une association contenue de cadre peut aussi pointer vers une structure de cadre, aboutissant dans ce cas, à des cadres imbriqués.

Contraintes : Aucune.

Etiquettes : Rangée (Row) : entier qui indique la rangée du cadre dans la structure de cadre auquel appartient la page, ou la cible associée.

Colonne (Col) : entier qui indique la colonne du cadre dans la structure de cadre auquel appartient la page, ou la cible associée.

II.2.4. soumet « submit »

Icône : Aucune.

Description : submit est une association qui se trouve toujours entre un formulaire et une page serveur. Les formulaires soumettent les valeurs de leurs champs au serveur, par l'intermédiaire de pages serveur, pour qu'il les traite. Le serveur Web traite la page serveur, qui accepte et utilise les informations du formulaire.

Contraintes : Aucune.

Etiquettes : Paramètres (parameters) : une liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

II.2.5. Construit « build »

Icône : Aucune.

Description : La relation « build » est une relation particulière qui fait le pont entre les pages client et les pages serveur. L'association « build » identifie quelle page serveur est responsable de la création d'une page client. C'est une relation orientée, puisque la page client n'a pas connaissance de la page qui est à l'origine de son existence.

Une page serveur peut construire plusieurs pages client, en revanche, une page client ne peut être construite que par une seule page serveur.

Contraintes : Aucune.

Etiquettes : Aucune.

I.2.6. Redirige « redirect »

Icône : Aucune

Description : Une relation « redirect », est une association unidirectionnelle avec une autre page web, peut être dirigée à partir d'une page client ou serveur ou vers une page client ou serveur.

Contraintes : Aucune.

Etiquettes

Délai (delay) : délai que doit observer une page client avant de rediriger vers la page destination. Cette valeur correspond à l'attribut Content de la balise <META>

II.2.7. IIOP (Internet Inter-ORB Protocol)

Description : IIOP est un type spécial de relation entre objets sur le client et objets sur le serveur. IIOP est un mécanisme de communication client/serveur différent de HTTP. On pourra trouver ce type de relation entre des JavaBeans sur le client et des EJB (Entreprise JavaBeans) sur le serveur.

Icône : Aucune.

Contraintes : Aucune.

Etiquettes : Aucune.

II.2.8. RMI (Remote Methode Invocation)

Icône : Aucune.

Description : RMI est un mécanisme qui permet à des applets Java, ou à des JavaBeans, d'envoyer des messages à des JavaBeans situés sur d'autres machines.

On pourra trouver ce type de relation entre des JavaBeans, ou des applets, sur le client et des EJB (Entreprise JavaBeans) sur le serveur.

Contraintes : Aucune.

Etiquettes : Aucune.

II.3. Attribut

II.3.1. Élément de saisie « input element »

Icône : Aucune.

Description : Un élément de saisie correspond à la balise <input> d'un formulaire HTML.. Les étiquettes associées à cet attribut stéréotypé, correspondent aux attributs de la balise <input>.

Les attributs obligatoires de la balise HTML <input> sont renseignés de la manière suivante : l'attribut name prend la valeur du nom de l'élément de saisie et l'attribut value prend celle de sa valeur initiale.

Contraintes : Aucune.

Etiquettes

Type (Type) : Le type de l'élément de saisie : texte, numérique, mot de passe, case à cocher, bouton d'option, bouton SUBMIT ou bouton RESET.

Taille (size) : définit la largeur visible allouée à l'écran en caractères.

- **Longueur Max (Maxlength)** : nombre maximal de caractères que peut saisir l'utilisateur.

II.3.2. Sélection d'éléments « select element »

Icône : Aucune.

Description : Contrôle de saisie employé dans les formulaires, il permet à l'utilisateur de sélectionner une ou plusieurs valeurs dans une liste. La plupart des navigateurs restituent ce contrôle par une liste d'options ou une liste déroulante.

Contraintes : Aucune.

Etiquettes

- **Taille (Size)** : définit le nombre d'éléments qui doivent être affichée simultanément.
- **Multiple (Multiple)** : valeur booléenne qui indique que plusieurs éléments peuvent être sélectionnés conjointement.

II.3.3. Zone de texte « texte area element »

Icône : Aucune.

Description : C'est un contrôle de saisie, employé dans les formulaires, qui permet l'écriture de plusieurs lignes de texte.

Contraintes : Aucune.

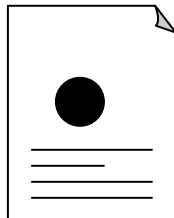
Etiquettes

- **Ligne (rows)** : nombre de lignes de texte visibles.
- **Colonnes (cols)** : largeur visible du texte en largeurs de caractères moyennes.

II.4. composant

II.4.1. page web « web page »

Icône



Description : Un composant page est une page web. Il peut être requis d'après son nom par un navigateur. Un composant page peut contenir des scripts client ou serveur.

Le plus souvent, le composant page est un fichier texte accessible au serveur Web, mais il peut également être un module compilé, chargé et exécuté par le serveur web. Dans les

deux cas, le serveur Web produit, à partir du composant page, un document au format HTML, qui est renvoyé en réponse à la requête du navigateur.

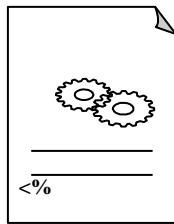
Contraintes : Aucune

Etiquettes

chemin (path) : chemin requis pour spécifier la page Web sur le serveur web. Cette valeur doit être relative au répertoire racine du site de l'application web.

II.4.2. Page ASP « ASP page »

Icône



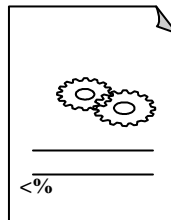
Description : Une page ASP est une page Web qui implémente du code ASP coté serveur. Ce stéréotype n'est pertinent que dans un environnement d'application basé sur les ASP (Active Server Page) de Microsoft.

Contraintes : Aucune.

Etiquettes : Identique à celles de la page web.

II.4.3. Page JSP « JSP page »

Icône



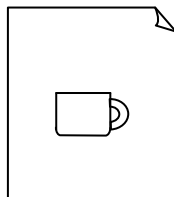
Description : Une page JSP est une page Web qui implémente du code JSP coté serveur. Ce stéréotype n'est pertinent que dans un environnement d'application basé sur les JSP (Java Server Pages).

Contraintes : Aucune.

Etiquettes : Identique à celles de la page web.

II.4.4. Servlet « Servlet »

Icône



Description : Une servlet est un composant Java. Ce stéréotype n'est pertinent que dans un environnement d'application compatible avec les servlets de Sun.

Contraintes : Aucune.

Etiquettes : Identique à celles de la page web.

II.4.5.bibliothèque de scripts « script library »

Icône



Description : Une bibliothèque de scripts est un composant qui propose un ensemble de sous-routines. Ou de fonctions, pouvant être inclus à d'autres composants pages web.

Contraintes : Aucune.

Etiquettes : Identique à celles de la page web.

III. Règles de cohérence sémantique

- **Réalisation de composant**

En principe, les composants pages Web peuvent réaliser les classes stéréotypées « serveur page », « client page », « form », « JavaScript object », « clientScript object », « frameset » et « target ». quand l'environnement de développement associé(ASP ou JSP) est en place, les pages web peuvent réaliser, au lieu des pages « server page », selon le cas, des classes stéréotypées « ASP page » ou « JSP page ».

- **Généralisation**

Tous les éléments de modélisation impliqués dans une même généralisation doivent être du même stéréotype.

- **Association**

Une page client peut avoir au plus une relation « build » avec une page serveur, mais une page serveur peut avoir plusieurs relations « build » avec différentes pages client. En plus des combinaisons standard d'UML, les combinaisons de stéréotypes présentées au Tableau ci-dessous sont permises.

De \ A	« client page »	« server page »	« frameset »	« target »	« form »
« client page »	« link » « redirect » « targeted link »	« link » « redirect » « targeted link »	« link » « redirect » « targeted link »	«dependenc»	«aggregation»
« server page»	« build » « redirect »	« redirect »	« build » « redirect »		
« frameset »	« frame content »		«frame content »	«frame conte»	
« target »					
« Form »	«aggregated by»	«submit»			

Tableau A- 1 : Combinaisons valides d'associations de stéréotypes.

Bibliographie

[1] mémoire ingénieur « Conception et réalisation d'un site web interactif pour la vente de véhicules Toyota. Cas « concessionnaire Toyota Haddad ». Réalisé par :Melle Berkal Farida & Mr Benyahia Ramdane. Proposé et dirigé par : Mr Samir Hameg. Promotion 2004/2005.

[2] [wikipedia](#)