

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



Université Mouloud MAMMARI de Tizi-Ouzou
Faculté de Génie électrique et Informatique
Département Informatique



Mémoire

De fin de cycle

*En vue de l'obtention du diplôme
de MASTER en informatique
Spécialité : Conduite de Projets Informatiques*

Thème

*Conception et réalisation d'une application
Web pour la gestion de la scolarité avec
PYTHON.*

Cas : DEPARTEMENT INFORMATIQUE

Proposé et dirigé par :
M^{me} BOURKACHE Gh.

Réalisé par :
M^r. BENABI NAZIM YACINE

2013/2014

Je tiens à exprimer ma profonde gratitude à ma promotrice, madame BOURKACHE Ghenima pour m'avoir encadré durant cette année,

Je tiens, également, à lui exprimer ma profonde reconnaissance pour le temps précieux qu'elle m'a consacré ainsi que pour ses conseils judicieux et généreusement prodigués.

Que les membres du jury trouvent ici mes plus vifs remerciements pour avoir accepté d'honorer par leur jugement mon travail.

Un grand merci aussi à toute personne qui de près ou de loin a contribué à ce que ce modeste travail voit le jour.

Table des matières

Table des matières	2
Table des figures	5
I Partie Théorique du projet	9
1 Les technologies Web, Internet et réseaux	10
1.1 Les réseaux	11
1.1.1 Les objectifs des réseaux	11
1.1.2 Avantages des réseaux [1]	11
1.1.3 Vocabulaire (notion) de base	12
1.1.4 Réseaux d'ordinateurs	12
1.1.5 Classification des réseaux [2]	12
1.1.5.1 Réseau local (LAN : Local Area Network)	13
1.1.5.2 Réseau métropolitain (MAN : Metropolitan Area Network)	13
1.1.5.3 Réseau étendu (WAN : Wide Area Network)	13
1.1.6 Architecture des réseaux [3]	14
1.1.6.1 Modèle de référence OSI	14
1.1.6.2 Architecture TCP / IP [3]	16
1.2 Le client - Serveur	16
1.2.1 Définition	16
1.2.2 Présentation de l'architecture d'un système client/serveur	16
1.2.3 Notions de base	16
1.2.4 Fonctionnement d'un système Client/Serveur	17
1.2.5 L'évolution de l'architecture Client/Serveur	18
1.2.5.1 Le client/serveur de première génération	18
1.2.5.2 Le client/serveur de deuxième génération [4]	18
1.2.5.3 Le client/serveur de troisième génération [4]	19
1.2.6 Les caractéristiques du Client-Serveur	19
1.2.7 Avantages de l'architecture client/serveur	20
1.2.8 Inconvénients du modèle client/serveur	20
1.3 Internet	20
1.3.1 Définition	20
1.3.2 Historique	21
1.3.3 Les principaux services de l'Internet	21
1.3.3.1 E-MAIL (la messagerie électronique)	21
1.3.3.2 IRC (Internet Relay Chat)	21
1.3.3.3 NEWS	21
1.3.3.4 FTP	21
1.3.3.5 TELNET	22
1.3.3.6 WAIS et GOPHER	22
1.3.3.7 WWW ou Web	22
1.3.3.8 DNS (Domain Name System)	22
1.3.3.9 Multimédia	23

1.3.4	Internet et le Client/Serveur	23
1.4	Le Web	24
1.4.1	Historique	24
1.4.2	Facteurs de succès du Web	24
1.4.3	Principaux termes du Web	25
1.4.4	Types de sites Web [5]	26
1.4.5	Les sites Web statiques [6]	27
1.4.6	Sites Web dynamiques	27
2	Le système LMD	29
2.1	Présentation du système LMD	30
2.1.1	Les objectifs de la réforme	30
2.1.2	Organisation des enseignements en LMD	30
2.2	Historique de la réforme LMD [8]	32
2.3	Evaluation et progression [9]	33
2.3.1	La progression dans les études [9]	34
2.3.2	La progression dans les études de Licence [9]	34
2.3.3	La progression dans les études de Master [9]	35
2.4	L'algorithme général pour les deux système de progression	35
2.4.1	Premier système de progression	35
2.4.2	Deuxième système de progression	36
2.5	Forme présentielle des études et Forme Tutorial des études	36
2.5.1	Forme présentielle des études	36
2.5.2	Forme tutorial des études	36
2.6	Présentation de département informatique de l'Université Mouloud Mammeri (UMMTO)	37
2.6.1	Historique [10]	37
2.6.2	Les formations [10]	37
2.6.2.1	Licence	37
2.6.2.2	Master	38
3	Analyse et Conception	39
3.1	Rappel sur UML	40
3.2	Identification des acteurs	41
3.3	Identification des cas d'utilisation	41
3.4	Spécification des scénarios :	42
3.4.1	Définition d'un scénario :	42
3.5	Spécification des cas d'utilisation (Use case)	44
3.5.1	Cas d'utilisation "S'authentifier"	44
3.5.2	Cas d'utilisation "Edition des PV Modulaire"	44
3.5.3	Cas d'utilisation "Edition des relevés de notes"	44
3.5.4	Cas d'utilisation "Changer le mot de passe"	44
3.6	Diagramme des cas d'utilisation globale de l'application gestion de la scolarité LMD	45
3.7	Diagramme de séquence détaillée cas utilisation	46
3.7.1	Diagramme de séquence détaillé cas utilisation « S'authentifier »	47
3.7.2	Diagramme de séquence détaillé cas utilisation « Changement de mot de passe »	48
3.7.3	Diagramme de séquence détaillé cas utilisation « Consultation des notes »	49
3.7.4	Diagramme de séquence détaillé cas utilisation « Edition des PV modulaires »	50
3.7.5	Diagramme de séquence détaillé cas utilisation « Affichage des PV modulaires »	51
3.7.6	Diagramme de séquence détaillé cas utilisation « Création de comptes »	52
3.8	Diagramme d'activités	53
3.8.1	Diagramme d'activités du cas d'utilisation « S'authentifier »	53
3.8.2	Diagramme d'activités du cas d'utilisation « Changer le mot de passe »	54
3.8.3	Diagramme d'activités du cas d'utilisation « Consultation des notes »	55
3.8.4	Diagramme d'activités du cas d'utilisation « Affichage des PV modulaires »	56
3.8.5	Diagramme d'activités du cas d'utilisation « Création de comptes »	57
3.9	Conception de la base de données	57

3.9.1	Le diagramme de classes	57
3.9.2	Structure des tables de la base de données	59
	3.9.2.1 Table Etudiant	59
	3.9.2.2 Table Enseignant	59
3.9.3	Table Module	59
3.9.4	Table Unité d'enseignement	59
3.9.5	Table Semestre	59
3.9.6	Table Spécialité	60
3.9.7	Table Note	60
3.9.8	Table année d'étude	60
3.9.9	Table Obtenir	60
3.10	Conclusion	60
II Partie pratique du projet		61
4	Réalisation	62
4.1	Introduction	63
4.2	Description de l'environnement de travail	63
4.3	Environnement logiciel	63
	4.3.1 NotePad ++	63
	4.3.2 Brackets	64
	4.3.3 Paint.NET	65
	4.3.4 Uwamp	66
	4.3.5 Eclipse For PHP	67
	4.3.6 Python For windows	67
	4.3.7 EasyPHP	68
	4.3.7.1 Installer EasyPHP	68
	4.3.7.2 Démarrer EasyPHP	68
4.4	Environnement de programmation	69
	4.4.1 Le serveur Web Apache	69
	4.4.2 Le serveur de base de données MySQL (My Structured Query language)	69
	4.4.3 L'outil phpMyAdmin	70
4.5	Les langages de programmation utilisés	70
	4.5.1 HTML5 (HyperText Markup Language)	70
	4.5.2 CSS 3 (Cascade Style Sheet)	71
	4.5.3 PHP (HyperText PreProcessor)	71
	4.5.4 SQL (Structured Query Language ou Langage de requête structuré)	71
	4.5.5 JQuery	72
	4.5.6 Python	72
	4.5.7 JavaScript	72
4.6	Présentation de quelques interfaces	73
	4.6.1 La page d'accueil	73
	4.6.2 Page "Ajouter utilisateur"	73
	4.6.3 Page d'administrateur	74
	4.6.4 Page de l'agent de scolarité	74
	4.6.5 Page "Imprimer"	75
	4.6.6 Page "Ajouter note"	75
4.7	Conclusion	76
III Annexe		79
Bibliographie		88

Table des figures

1.1	LAN : Local Area Network	13
1.2	WAN : Wide Area Network	13
1.3	Le modèle OSI en 7 couches.	14
1.4	Modèle TCP/IP	16
1.5	Modèle Client/Serveur	17
1.6	Architecture Client/Serveur	17
1.7	Le Client/Serveur à 2 niveaux.	18
1.8	Le Client/Serveur à 3 niveaux.	19
1.9	Internet	20
1.10	File Transfer Protocol (FTP)	22
1.11	DNS (Domain Name System)	23
1.12	Internet selon le modèle Client/ Serveur	24
1.13	Page Web	25
1.14	Langage HTML	26
2.1	Architecture du LMD	31
2.2	Architecture du système LMD en Algerie	32
2.3	L’algorithme du premier système de progression	35
2.4	L’algorithme du deuxième système de progression	36
3.1	Composantes de la modélisation UML.	41
3.2	Diagramme des cas d’utilisation globale	45
3.3	Diagramme de séquence “s’authentifier Enseignant “	47
3.4	Diagramme de séquence “Changer le mot de passe”	48
3.5	Diagramme de séquence “consulter notes “	49
3.6	Diagramme de séquence “Edition PV modulaire “	50
3.7	Diagramme de séquence “affichage PV modulaire “	51
3.8	Diagramme de séquence “Création de comptes “	52
3.9	Diagramme d’activité du cas d’utilisation “S’authentifier “	53
3.10	Diagramme d’activité du cas d’utilisation “changer mot de passe “	54
3.11	Diagramme d’activité du cas d’utilisation “Consultation notes”	55
3.12	Diagramme activité du cas d’utilisation “Affichage PV modulaire “	56
3.13	Diagramme d’activité du cas d’utilisation “Création de compte “	57
3.14	Le diagramme de classe	58
4.1	Interface NotePad++	64
4.2	Interface de Brackets	65
4.3	Interface Paint.NET	66
4.4	Interface Uwamp	67
4.5	Interface de EasyPHP	69
4.6	Interface PhpMyAdmin	70
4.7	Page d’acceuil	73
4.8	Page “Ajouter utilisateur”	73
4.9	Page administrateur	74

4.10 Page de l'agent de scolarité	74
4.11 Page "Imprimer"	75
4.12 Page "Noter module"	75

Introduction générale

La fin du vingtième siècle a été caractérisée par l'évolution rapide des technologies de l'information en général et l'informatique en particulier. Les effets positifs de cette évolution ont touchés tous les domaines de notre vie quotidienne et ont provoqué un bouleversement dans notre façon de communiquer.

Ces évolutions remarquables et la démocratisation de l'outil informatique ont permis l'éclosion d'applications distribuées comme par exemple, le World Wide Web, la messagerie électronique, les forums de discussion, Ces deux dernières s'inscrivent dans le domaine de la communication sur Internet, qui permet aussi une communication en temps réel.

L'évolution a aussi touchée l'univers des études et en plus particulier le monde de l'université. En effet, la réforme de LMD « Licence-Master-Doctorat » désigne un ensemble de mesures modifiant le système d'enseignement supérieur pour l'adapter aux standards européens. Elle met en place principalement une architecture basée sur trois grades : licence, master et doctorat, une organisation des enseignements en semestres et unités. Le LMD est introduit de manière graduelle, ce système commence à se généraliser à l'université Mouloud MAMMERI de Tizi-Ouzou (UMMTO).qui a pour principes et objectifs de remédier aux dysfonctionnement du système classique, normalisation internationale des diplômes et des compétences, proposition de parcours de formation diversifiés et adaptés au milieu socioéconomique et promouvoir la mobilité des étudiants en garantissant la capitalisation et le transfert des acquis

Les principales caractéristiques de la réforme sont : Proposition de grades de formation, agréés et reconnus à l'échelle mondiale, organisation semestrielle des enseignements, les formations sont proposées sous forme d'offres de formation qui se déclinent en domaine, filière, spécialité.

L'objectif de notre travail est de développer une application web pour prendre en charge les enseignements au sein du département informatique, cette application permettra aux enseignants d'éditer les PV modulaires (saisie des notes,etc), consulter les PV de délibérations, changer le mot de passe, elle permettra à l'agent de scolarité d'éditer les listes des étudiants, d'éditer aussi les PV globaux de délibérations,....etc et aussi permettra à l'administrateur de créer des comptes, modifier ses derniers, et en fin permettra à l'étudiant de consulter ses notes.

Pour mener à bien notre travail, nous proposons une organisation de notre démarche en quatre points qui seront déroulés tout au long de ce rapport, ce dernier est présenté comme suit :

1. Le premier chapitre sera une présentation générale des réseaux et du Web : leurs avantages, leurs modèles d'architectures en s'axant beaucoup plus sur le modèle Internet, ses protocoles et ses applications ;
2. Le deuxième chapitre sera sur le système LMD ;
3. Le troisième chapitre comportera les différentes étapes de la conception de l'application ;
4. Le quatrième chapitre sera consacré à la réalisation de l'application .

Première partie

Partie Théorique du projet

Chapitre 1

Les technologies Web, Internet et réseaux

Introduction

les réseaux informatiques occupent aujourd'hui une place importante dans l'évolution technologique. à leur origine, ils permettent de relier des terminaux passifs à de gros ordinateurs centraux, actuellement, ils permettent l'interconnexion de tous types d'ordinateurs personnels ou de simples terminaux graphiques. ils sont nés d'un besoin d'échanger des informations de manière simple et rapide entre plusieurs machines.

actuellement, la technologie réseaux et les domaines qu'elle inclut prennent le monopole, en effet, les réseaux sont tellement répandus qu'ils touchent pratiquement à tous les aspects de notre vie quotidienne.

le présent chapitre est une introduction aux nouvelles technologies de l'information. dans un premier temps, nous allons commencer par introduire des généralités sur les réseaux et présenter l'architecture client/serveur, puis nous allons présenter l'internet, le service le plus populaire d'internet qui est le www (world wide web).

1.1 Les réseaux

Les réseaux sont nés d'un besoin d'échanger des informations de manière simple et rapide entre des machines. Lorsque l'on travaillait sur une même machine, toutes les informations nécessaires au travail étaient centralisées sur cette machine. Pour des raisons de coûts ou de performances, on est venu à multiplier le nombre de machines. Les informations devaient alors être dupliquées sur les différentes machines du même site. Cette duplication était plus ou moins facile et ne permettait pas toujours d'avoir des informations cohérentes. On est donc arrivé à relier d'abord ces machines entre elles. Ce fût l'apparition des réseaux locaux. Ces réseaux étaient souvent des réseaux propriétaires. Plus tard on a éprouvé le besoin d'échanger des informations entre des sites distants. Les réseaux moyenne et longue distance commencèrent à voir le jour. Aujourd'hui, les réseaux se retrouvent à l'échelle planétaire. Le besoin d'échanger de l'information est en pleine évolution.

1.1.1 Les objectifs des réseaux

Les Réseaux permettent :

- Le partage de fichiers ;
- Le transfert de fichier ;
- Le partage d'application : compilateur, système de gestion de base de données (SGBD) ;
- Le Partage de périphériques ;
- L'interaction avec les utilisateurs connectés : messagerie électronique, conférence électronique, Talk. . .
- Le transfert de données en général (réseaux informatiques) ;
- Le transfert de la parole (réseaux téléphoniques) ;
- Le transfert de la parole, de la vidéo et des données.

1.1.2 Avantages des réseaux [1]

- **Communication facile et rapide de l'information** : Particulièrement importante dans le domaine de la recherche qui a vu naître les grands réseaux, la communication rapide et à

- grande échelle de l'information est indispensable à toute organisation dont la taille dépasse le groupe d'individus. Une entreprise répartie sur plusieurs sites ne peut s'en passer d'un réseau.
- **Partage de ressources (matérielles, logicielles, données)** : La mise en commun des ressources matérielles (imprimantes, espace disque, périphériques coûteux ou calculateurs puissants) utilisées épisodiquement est une motivation à la mise en réseau.
 - La mise en commun de ressources logicielles procède de la même logique, une licence logicielle, comme une imprimante, peut être partagée. Ces deux techniques engendrent une économie de moyens.
 - La mise en commun des données est un point essentiel au bon fonctionnement d'une organisation, car la centralisation et le partage de l'information permettent d'éviter les incohérences et la duplication.
 - **Accès immédiat et transparent à l'outil informatique le plus adapté** : Il s'agit simplement d'optimiser l'investissement en moyens informatiques si l'accès transparent à l'information est assuré, la mise en réseau de machines de puissances très différentes permet d'utiliser chacune de façon optimale, sans inconvénient pour l'utilisateur.

1.1.3 Vocabulaire (notion) de base

- **Station de travail** : on appelle station de travail toute machine capable d'envoyer des données vers les réseaux (PC, MAC, SUN...). Chaque station de travail à sa propre carte interface (carte réseau).
- **Nœud** : c'est une station de travail, une imprimante, un serveur ou toute entité pouvant être adressée par un numéro unique. L'unicité de l'adresse est garantie par le constructeur d'une carte réseau qui donne un numéro unique ne pouvant être changé par une personne.
- **Serveur** : dépositaire centrale d'une fonction spécifique, serveur de base de données, de calcul, de fichier... .
- **Paquet** : C'est la plus petite unité d'information pouvant être envoyée sur le réseau. Un paquet contient en général l'adresse de l'émetteur, l'adresse du récepteur et les données à transmettre.
- **Trame** : c'est une unité structurée de bits.
- **Topologie** : organisation physique et logique d'un réseau. L'organisation physique concerne la façon dont les machines sont connectées (Bus, Anneau, Etoile, Maillé, Arborescence,...). La topologie logique montre comment les informations circulent sur le réseau (diffusion, point à point).

1.1.4 Réseaux d'ordinateurs

Un réseau d'ordinateurs est un ensemble d'ordinateurs interconnectés. Deux ordinateurs sont considérés interconnectés s'ils sont capables d'échanger des informations. Cette connexion peut être par câble de cuivre, fibres optiques, satellite...

1.1.5 Classification des réseaux [2]

La principale classification des réseaux est faite selon leur taille en : Réseau local (LAN), réseau métropolitain (MAN) et réseau étendu (WAN).

1.1.5.1 Réseau local (LAN : Local Area Network)

Il est formé d'un ensemble de stations situées dans une même zone géographique limitée (par exemple, un ensemble de bureaux, un immeuble).

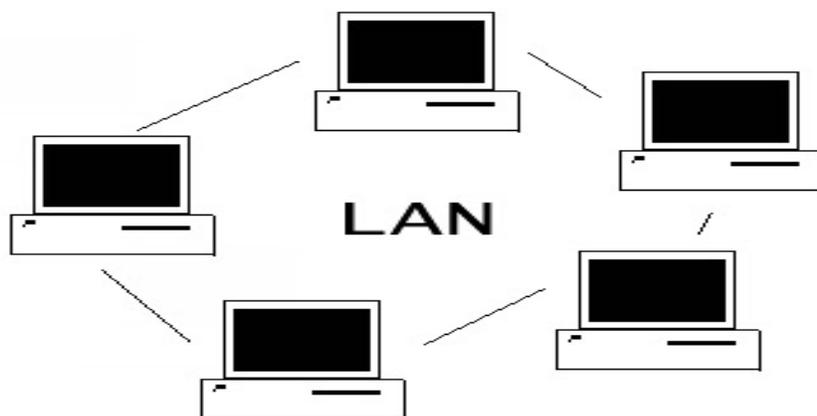


FIGURE 1.1: LAN : Local Area Network

1.1.5.2 Réseau métropolitain (MAN : Metropolitan Area Network)

C'est un réseau situé dans une zone géographique plus petite que les réseaux étendus. Les réseaux métropolitains permettent des communications entre différents grands comptes ou à partir d'ordinateurs personnels.

1.1.5.3 Réseau étendu (WAN : Wide Area Network)

Relie plusieurs réseaux locaux entre eux. Les réseaux étendus couvrent une zone géographique importante et les liaisons entre les réseaux locaux se font à travers des lignes de communication internationales et des routeurs. Ces réseaux étendus relient des machines hétérogènes qui utilisent des systèmes d'exploitation différents.

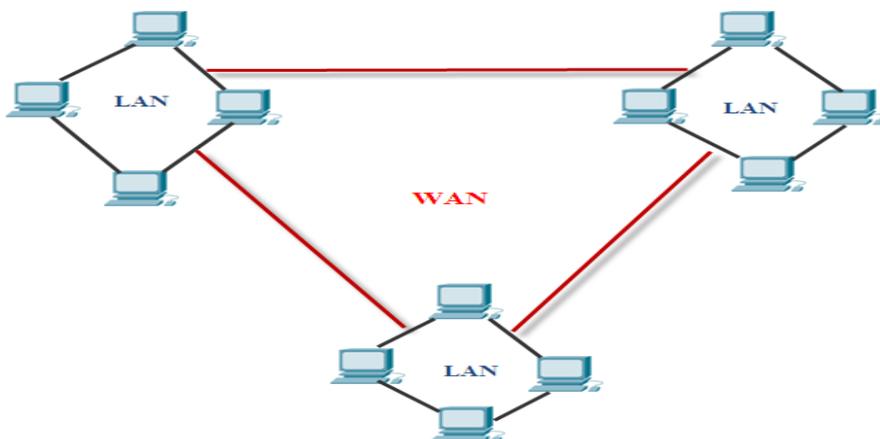


FIGURE 1.2: WAN : Wide Area Network

1.1.6 Architecture des réseaux [3]

Pour assurer la connexion d'une machine, il faut réunir les supports physiques. Mais pour s'assurer du bon transfert de l'information avec une qualité de service suffisante, il faut prévoir une architecture logicielle.

Une normalisation de l'architecture logicielle s'impose. Dans cette section nous allons décrire deux architectures réseau, la première provient de l'ISO et s'appelle OSI (open system interconnexion), la deuxième est l'architecture TCP/IP.

1.1.6.1 Modèle de référence OSI

Le modèle de référence OSI se fonde sur une proposition élaborée par l'organisation internationale de normalisation (ISO) ; il est appelé Modèle de Référence OSI (Open System Interconnexion) parce qu'il traite de la connexion entre les systèmes ouverts en communication avec d'autres systèmes. Les principes de base ayant conduit à l'élaboration des sept couches sont les suivants :

- Diviser les problèmes de communication sur les réseaux en problèmes plus simples et plus faciles à gérer.
- Chaque couche exerce des fonctions bien définies.
- Le nombre de couches doit être suffisamment grand pour éviter la cohabitation dans une couche de fonctions très différentes et suffisamment petit pour éviter que l'architecture devienne difficile à maîtriser.

D'un point de vue conceptuel, chaque couche interagit avec son homologue située sur un ordinateur distant. En pratique, chaque couche communique avec la couche au dessus et en dessous d'elle.

Fonctionnement : Chaque couche (n) offre un certain nombre de services à la couche (n+1) en déroulant un protocole uniquement défini à partir des services fournis par la couche (n-1).

L'architecture OSI est schématisée comme suit :

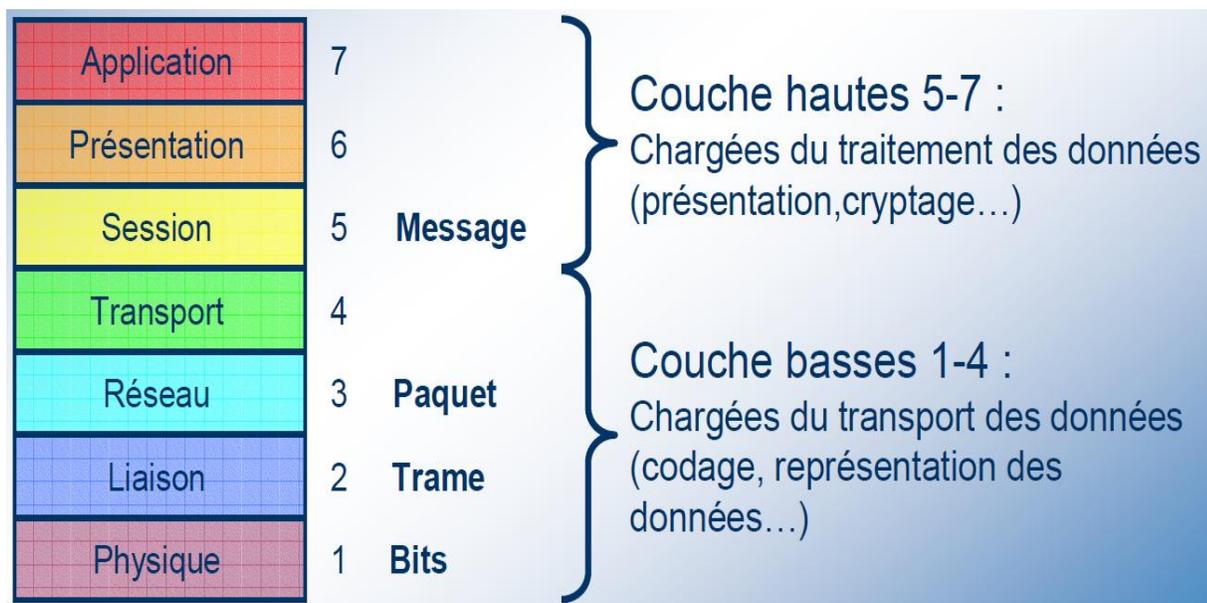


FIGURE 1.3: Le modèle OSI en 7 couches.

Couche physique :

Décrit les propriétés physiques, telles que les caractéristiques mécaniques des différents supports et les caractéristiques des signaux qui véhiculent l'information, rentrent dans cette définition les caractéristiques des câbles et des connecteurs, les niveaux de potentiel, les intensités de courants, etc. cette couche s'occupe de la transmission des bits de façon brute sur le canal de communication.

Couche liaison de données :

La tâche principale de la couche liaison de données est de prendre un moyen de transmission et de le transformer en une liaison de données, elle l'accomplit en fractionnant les données d'entrée de l'émetteur en trames de données, en transmettant les trames en séquences et en gérant les trames d'acquittement renvoyées par le récepteur, c.-à-d. elle s'occupe du transport des données à travers les supports physiques en manipulant les adresses physiques (elles sont intégrées dans les cartes réseau du fabricant), la couche physique accepte et transmet simplement un flot de bits sans connaître les frontières des trames, c'est à la couche liaison de données de créer et de reconnaître les frontières des trames. Et c'est à cette couche de récupérer les trames détruites au cours de route, empêcher un émetteur rapide de saturer de données un récepteur lent en utilisant un mécanisme de régulation de flux.

Couche réseau :

Cette couche est concernée par les adresses logiques (telles que les adresses IP). Cette couche fournit des connexions et un routage entre deux nœuds d'un même réseau.

Couche transport :

La couche transport fournit des services tout à fait similaires à ceux fournis par la couche réseau. Certaines couches réseau fournissent une qualité de services qui assure une bonne fiabilité. Cependant, toutes les couches réseau ne le font pas. Une partie des fonctionnalités fournies par la couche transport a pour objet de garantir une meilleure fiabilité. Cette fiabilité des transmissions est assurée par le lancement d'un protocole approprié à la couche transport, la fonction de base de cette couche est d'accepter des données de la couche session, de les couper, le cas échéant, en plus petites unités, de les passer à la couche réseau et de s'assurer que tous les morceaux arrivent correctement de l'autre côté.

Couche session :

Elle permet à des utilisateurs travaillant sur différentes machines d'établir des sessions entre eux. Les services de cette couche concernent la gestion de dialogue (ex : la synchronisation).

Couche présentation :

Remplit des fonctions suffisamment courantes et génériques pour ne pas les laisser à la charge de l'utilisateur. Plus précisément, à la différence des autres couches, qui sont concernées seulement par la transmission fiable des bits d'un point à un autre, la couche présentation s'intéresse à la syntaxe et la sémantique de l'information transmise.

Couche application :

Représente la couche la plus élevée du modèle OSI, elle utilise les services de la couche présentation (et indirectement des autres couches) pour exécuter une application spécifique. L'application peut être échange de courrier, transfert de fichiers ou toute application réseau.

1.1.6.2 Architecture TCP / IP [3]

La défense américaine devant le fonctionnement des machines utilisant des protocoles de communication différents et incompatibles à décider de définir sa propre architecture. Ces protocoles représentent aussi, comme l'architecture OSI, une architecture en couches. L'architecture TCP/IP se schématise comme suit :

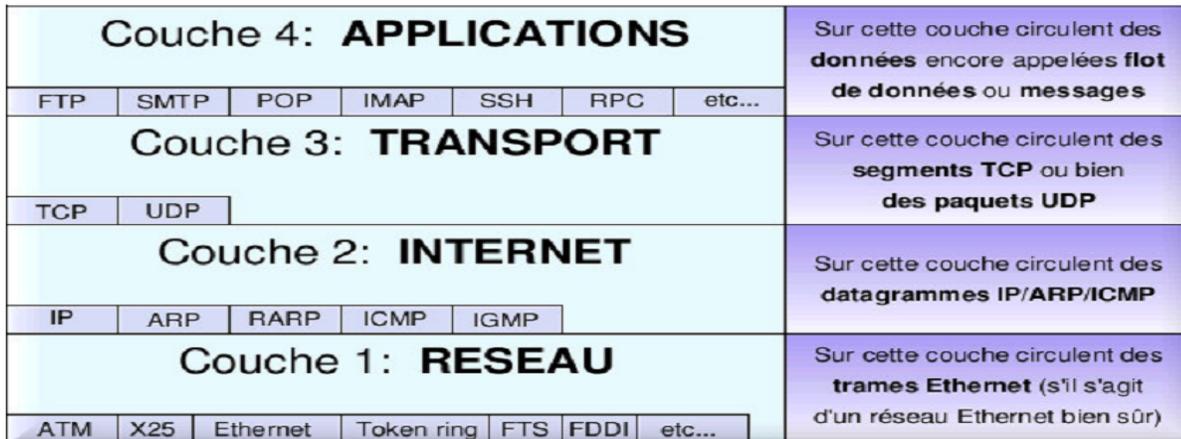


FIGURE 1.4: Modèle TCP/IP

1.2 Le client - Serveur

1.2.1 Définition

C'est un modèle informatique basé sur le traitement distribué selon le quel un utilisateur lance un logiciel client à partir d'un ordinateur relié à un réseau, déclenchant simultanément le lancement d'un logiciel serveur situé dans un autre ordinateur possédant les ressources souhaitées par l'utilisateur.

1.2.2 Présentation de l'architecture d'un système 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 terme de capacités d'entrée-sortie, qui leur fournit des services.

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client FTP, client de messagerie, lorsque l'on désigne un programme, tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès du serveur. (Dans le cas du client FTP il s'agit de fichiers, tandis que pour le client messagerie il s'agit de courrier électronique). Dans un environnement purement Client/serveur, les ordinateurs du réseau (les clients) ne peuvent voir que le serveur, c'est un des principaux atouts de ce modèle (chaque machine est soit client, soit serveur).

1.2.3 Notions de base

- **Client** : C'est le processus demandant l'exécution d'une opération à un autre processus par envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.

- **Serveur** : C'est un processus accomplissant une opération sur demande d'un client.
- **Requête** : C'est un message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte d'un client.
- **Réponse** : C'est un message transmis par un serveur à un client suite à l'exécution d'une opération contenant les paramètres de retour de l'opération.
- **Middleware** : C'est le logiciel qui, au milieu assure les dialogues entre les clients et les serveurs souvent hétérogènes.

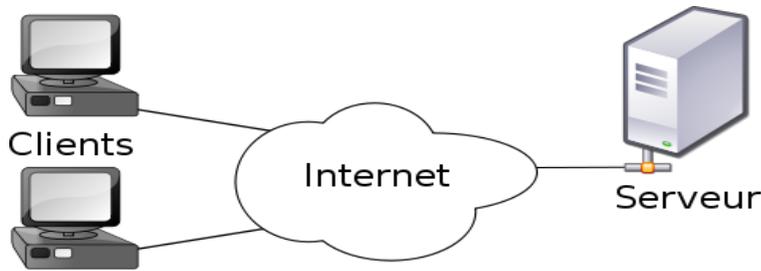


FIGURE 1.5: Modèle Client/serveur

1.2.4 Fonctionnement d'un système Client/serveur

Un système client/serveur fonctionne selon le schéma suivant :

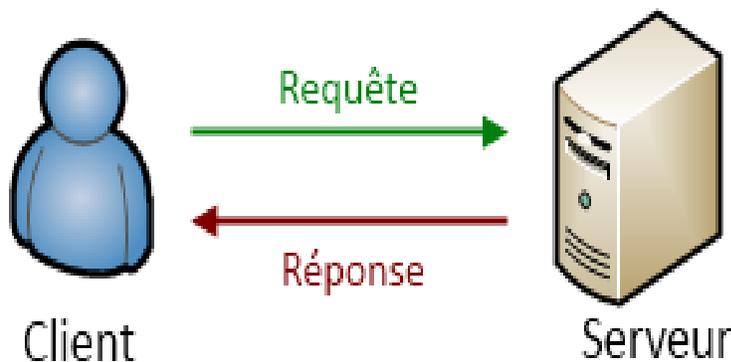


FIGURE 1.6: Architecture Client/serveur

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

Objectifs de cette architecture :

- Une plus grande flexibilité/souplesse.
- Une plus grande sécurité (la sécurité peut être définie pour chaque service).
- De meilleures performances (les tâches sont partagées).

1.2.5 L'évolution de l'architecture Client/Serveur

En espace de quelques années, les progrès réalisés dans les technologies de l'information nous ont fait entrer dans une nouvelle ère, celle de l'informatique ouverte au grand public, et du client/serveur.

Le client/serveur apparaît aujourd'hui comme le point de convergence des besoins qu'expriment les entreprises depuis des années, celui d'intégrer les nouvelles technologies sans remettre en cause les systèmes existants, de fournir aux utilisateurs rapidité, souplesse et confort dans l'utilisation quotidienne de l'outil informatique et plus généralement de disposer d'un système informatique adapté aux fortes exigences qu'impose le contexte économique actuel.

Le modèle client/serveur constitue une évolution majeure de l'informatique. Le principe de base est de décomposer un processus informatique en au moins deux tâches moins complexes (le client et le serveur) associés à un mécanisme de communication leur permettant de coopérer.

1.2.5.1 Le client/serveur de première génération

Né à la fin des années 80 et basé sur des outils clients autour des SGBD relationnels. Le développement s'effectue sur le serveur pour la base de données et sur le client pour l'application.

L'inconvénient de cette architecture est que tout le code applicatif est exécuté sur le client.

1.2.5.2 Le client/serveur de deuxième génération [4]

Né au milieu des années 90, cette génération est caractérisée par :

- La possibilité de développer des traitements applicatifs au sein du serveur de données sous forme de procédures déclenchées par l'application ou lors d'événements sur la base de données.
- L'utilisation de l'approche orientée objet. La facilité de déploiement des applicatifs avec partitionnement automatique du code applicatif entre le client et le serveur.

Présentation de l'architecture à deux niveau :

L'architecture à deux niveaux (appelée aussi architecture 2-tier, tier signifiant étage en anglais) caractérise les systèmes client/serveur dans lesquels le client demande la ressource et le serveur la lui fournit directement (sans intermédiaire). Cela signifie que le serveur ne fait appel à une autre application afin de fournir le service.

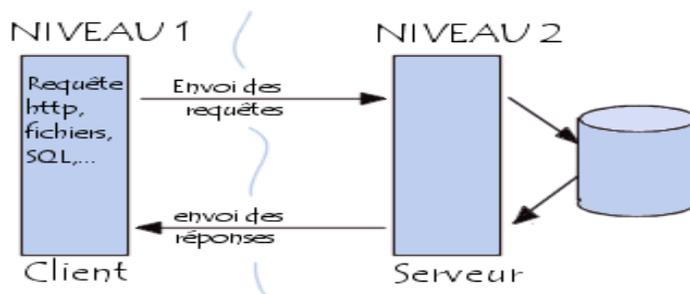


FIGURE 1.7: Le Client/Serveur à 2 niveaux.

1.2.5.3 Le client/serveur de troisième génération [4]

Avec l'apparition d'Internet et le Web, les architectures client/serveur ont évoluées vers des architectures à trois niveaux. Le client est responsable de la présentation. Il utilise pour cela des navigateurs Web, comme Internet explorer. Le serveur d'application exécute le code applicatif. Le serveur de données supporte le SGBD qui gère éventuellement des types de données très riches, par exemple multimédias ou spécifiques à l'entreprise.

Le serveur d'application et les données peuvent être regroupés sur la même machine. La couche application comporte le serveur Web (ou serveur HTTP) capable de comprendre le protocole HTTP. La couche cliente se compose de navigateurs Web chargés uniquement de la présentation.

L'architecture à trois-tier, appelée encore client/serveur Web, est aujourd'hui bien adaptée aux systèmes répartis autour d'un réseau local et/ou d'Internet.

Elle permet à de multiples postes ou stations de travail distribué sur la planète de partager les mêmes données. Dans cette architecture la gestion des données et des applications est centralisée au niveau du serveur, en offrant la distribution d'accès.

Présentation de l'architecture à trois niveaux :

Dans l'architecture à trois niveaux (appelé aussi architecture 3-tier) il existe un intermédiaire, cette architecture est généralement partagée entre :

- **Le client** : qui demande la ressource.
- **Le serveur d'application (ou middleware)** : C'est le serveur chargé de fournir la ressource mais faisant appel à un autre serveur.
- **Le serveur secondaire** : celui qui fournit le service au premier serveur (souvent un serveur de base de données).

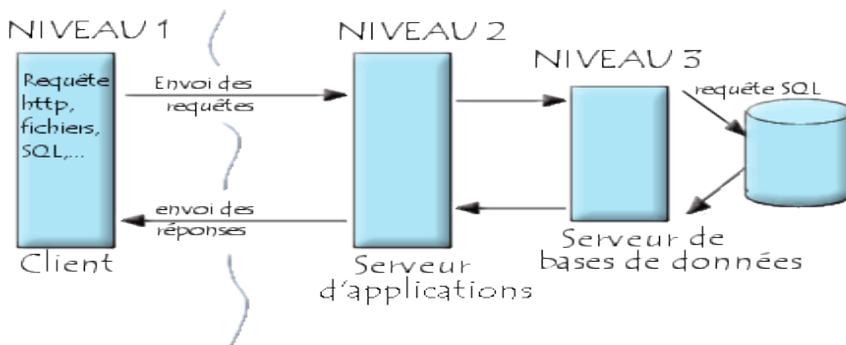


FIGURE 1.8: Le Client/Serveur à 3 niveaux.

1.2.6 Les caractéristiques du Client-Serveur

- **Protocole** : c'est toujours les clients qui déclenchent la demande de service. Le serveur attend passivement les requêtes des clients.
- **Partage des ressources** : un serveur traite plusieurs clients en même temps et contrôle leurs accès aux ressources.
- **Localisation** : le logiciel client-serveur masque aux clients la localisation du serveur.
- **Hétérogénéité** : le logiciel client-serveur est indépendant des plates-formes matérielles et logicielles.

- **Redimensionnement** : il est possible d'ajouter et de retirer des stations clientes. Il est possible de faire évoluer les serveurs.
- **Intégrité** : les données du serveur sont gérées sur le serveur de façon centralisée. Les clients restent individuels et indépendants.
- **Souplesse et adaptabilité** : on peut modifier le module serveur sans toucher au module client. La réciproque est vraie. Si une station est remplacée par un modèle plus récent, on modifie le module client sans modifier le module serveur.

1.2.7 Avantages de l'architecture client/serveur

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

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

1.2.8 Inconvénients du modèle client/serveur

L'architecture client/serveur a tout de même quelques lacunes parmi lesquelles :

- Un coût élevé dû à la technicité du serveur.
- Un maillon faible : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui.

1.3 Internet

1.3.1 Définition

Il est presque impossible de cerner une telle question, car c'est une technologie en constante évolution. Pour cela, on dira que l'Internet est un gigantesque réseau mondial d'ordinateurs qui englobe plusieurs autres réseaux de moindre taille. Ces derniers permettent eux-mêmes d'interconnecter des ordinateurs individuels.



FIGURE 1.9: Internet

1.3.2 Historique

Internet est issu des technologies développées au cours des années 70 par la DARPA (Defense Advanced Research Project Agency) qui s'appelait ARPANET (une organisation liée à l'armée américaine). A l'origine, l'objectif de cette organisation était d'interconnecter les organismes militaires et les centres de recherche dans le but de permettre le transfert fiable de données et d'autre part d'accéder à des centres dotés de super calculateurs coûteux. Très vite le réseau draina un grand nombre d'universités et d'entreprises privées impliquées dans le projet de recherche du gouvernement américain.

En 1983, l'infrastructure de base actuelle (de l'Internet) était achevée, avec l'arrivée des différentes entreprises commerciales sur les réseaux. Ceci poussa l'armée américaine à abandonner le projet Internet pour laisser le champ à ces entreprises et les utilisateurs individuels. Alors depuis ce temps, l'Internet permet d'échanger des messages électroniques (e-mails) et les fichiers FTP entre les ordinateurs sauf que le mode d'emploi de l'Internet était compliqué. C'était des lignes de commandes presque incompréhensibles. Ce n'est que vers les années 90 que l'Internet a connu un tel succès grâce au développement de l'interfaçage graphique et l'invention du WWW (World Wide Web (toile d'araignée mondiale)).

1.3.3 Les principaux services de l'Internet

Les services d'Internet sont généralement liés à la communication :

1.3.3.1 E-MAIL (la messagerie électronique)

C'est utiliser Internet comme on utilise la poste. Il est possible de déposer un message dans la boîte aux lettres de son correspondant, qu'il soit ou non devant sa machine. Ce dernier sera capable à sa prochaine connexion de consulter sa boîte aux lettres pour lire et envoyer des messages à ses correspondants.

1.3.3.2 IRC (Internet Relay Chat)

C'est une forme de communication interactive entre un individu et un autre par envoi de textes écrits.

1.3.3.3 NEWS

A l'inverse du mail ou la discussion est réalisée de l'émetteur vers le récepteur, les forums de discussion sont des moyens de discussion entre plusieurs personnes.

1.3.3.4 FTP

C'est le service d'échange de fichiers, il permet de déposer des fichiers sur une machine distante, mais aussi de télécharger des fichiers sur sa machine.

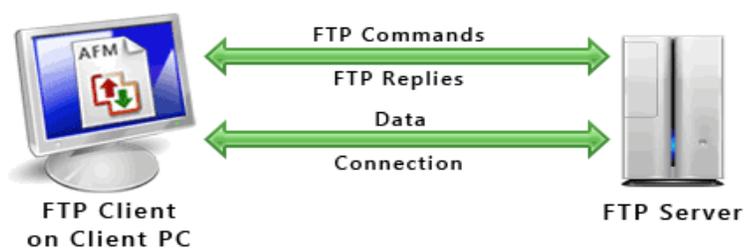


FIGURE 1.10: File Transfer Protocol (FTP)

1.3.3.5 TELNET

Elle permet à un internaute de se connecter et donc d'utiliser à distance une machine comme s'il se trouvait face à elle.

1.3.3.6 WAIS et GOPHER

les services Wais (wide area information servers) et Gopher permettent de retrouver des documents de divers types localisés sur des serveurs distants. Les informations sont très variées on y trouve par exemple, des images, des sons, des banques de données.

1.3.3.7 WWW ou Web

c'est le service le plus connu de l'Internet que nous allons détailler après.

1.3.3.8 DNS (Domain Name System)

DNS est un service qui convertit l'URL d'une page WEB en son adresse IP (adresse binaire sur 32 bits définissant sans ambiguïté chaque ordinateur ou routeur du réseau Internet). Les adresses IP sont obtenues par interrogation d'un serveur de noms. Celui-ci gère un tableau contenant les noms de domaine et les adresses IP correspondantes.

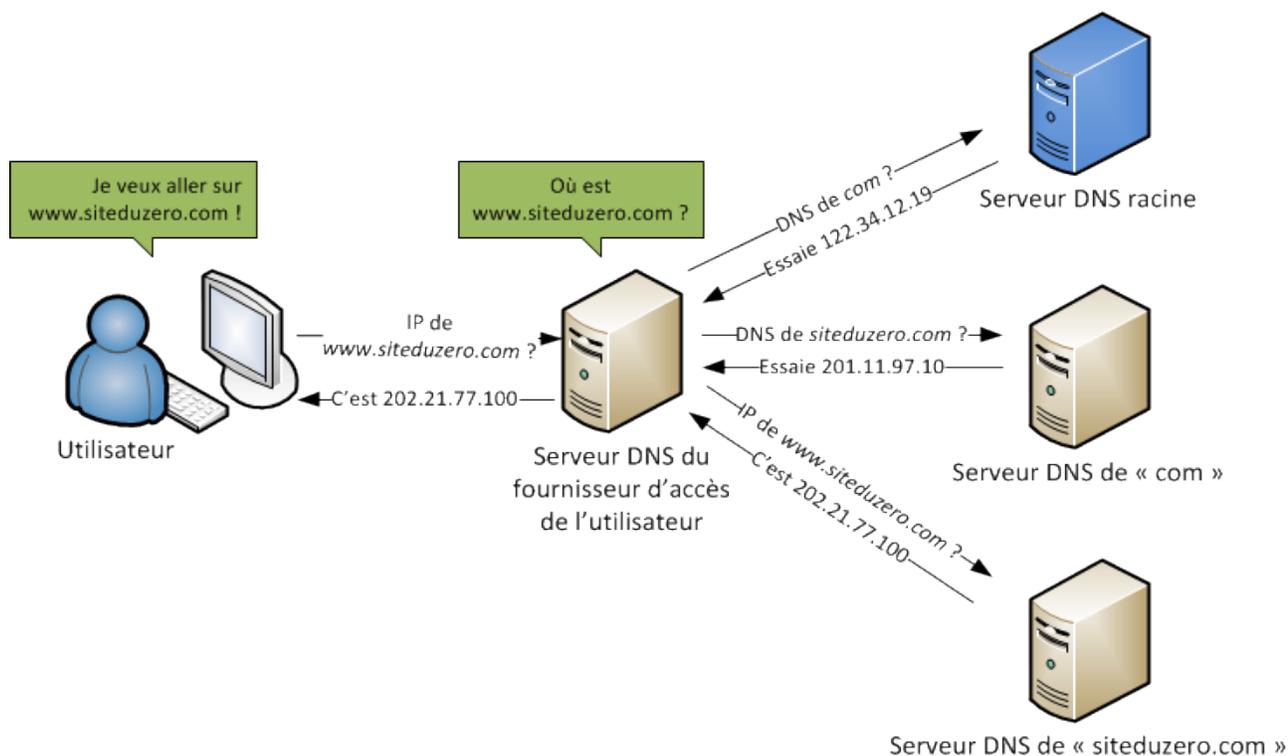


FIGURE 1.11: DNS (Domain Name System)

1.3.3.9 Multimédia

Avec le Web, le multimédia est entré dans l'air du grand public, boutiques de modes, agences de voyages, publicités, bibliothèques virtuelles, musique en ligne, commerce électronique, toutes ces applications mixent du multimédia. Dans les bases de données le terme « multimédia » signifie en conséquence le mélange de différents types de données incluant texte libre, géométrie, image, son, vidéo, etc.

1.3.4 Internet et le Client/Serveur

Le paradigme du client-serveur s'applique totalement à Internet. On parle de services, de serveurs Web, de serveurs FTP et de clients Internet pour le navigateur. On distingue dans le réseau Internet deux types de machines :

1. Celles qui ne servent qu'à consulter des informations (clients), sur lesquelles n'existe qu'un navigateur. Elles ne font qu'utiliser les services de l'Internet.
2. Celles qui diffusent des services (serveurs), des informations, etc.

Les machines sur Internet utilisent le protocole TCP/IP pour dialoguer, le protocole HTTP pour les hyperdocuments, NNTP (News Network Protocol) pour les news, selon l'approche client-serveur et sachant que tout accès Internet est soit une demande de service(s), soit une offre de service(s), Internet peut être vu comme un réseau de réseaux de serveurs à la disposition de millions de clients.

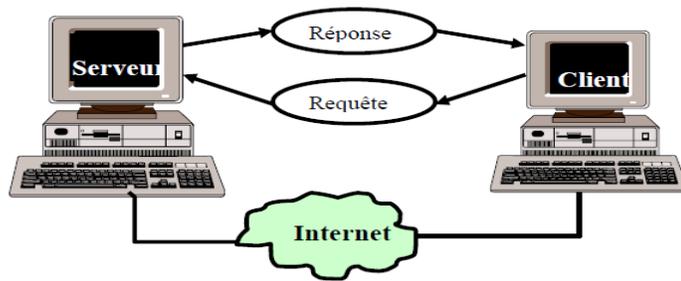


FIGURE 1.12: Internet selon le modèle Client/ Serveur

1.4 Le Web

World Wide Web (toile d'araignée mondiale) est un système d'information hypermédia sur Internet. Il est fondé sur l'extension de la technique de l'hypertexte aux réseaux, et a créé le premier réseau hypermédia réparti de couverture mondiale, donnant ainsi la possibilité d'accès à de très grandes quantités et variétés de documents de façon très simple. Le WEB repose sur une architecture client-serveur distribuée et une interface de navigation offrant la puissance d'hypertexte en environnement graphique.

1.4.1 Historique

Le WWW est inventé en 1989 et développé en 1992, au CERN (Centre Européen de Recherche Nucléaire) à Genève, par Tim Berners Lee, un chercheur atomiste en quête d'un outil de partage de connaissances scientifiques avec les autres chercheurs de toute l'Europe. Il était le premier à proposer de relier les documents sous forme d'une toile d'araignée c.-à-d. des documents reliés avec des liens hypertextes. En 1993, une première interface graphique, Mosaic, est livrée au NCSA (National Center For Supercomputing Applications).

1.4.2 Facteurs de succès du Web

Ces facteurs sont principalement :

Universalité : Ceci grâce à :

- La navigation hypertexte étendue : les liens peuvent s'appliquer à des documents hébergés sur d'autres sites au sens géographique du terme.
- Support multimédia : images fixes, animées, sons, vidéos, etc.
- Multi-plateforme : les codages utilisés permettent d'utiliser toutes les machines existantes.

Simplicité : est assurée par :

- L'unité de visualisation du contenu (navigateur)
- Les liens hypertextes
- Les systèmes d'adresses URL.
- Intégration de tous les services qui existent (messagerie, FTP, news, etc.)

Gratuité des protocoles : Sont offerts au domaine public comme « http ».

1.4.3 Principaux termes du Web

1. **Page Web** : C'est une série de mots disposés linéairement et d'objets graphiques, sonores ou vidéo. Certains mots ou graphiques (souvent en surbrillance ou soulignés) pointent vers d'autres pages, c'est ce qu'on appelle hyperlien qui peuvent être obtenues en cliquant sur ces mots ou graphiques. Cette technique d'hypertexte permet la lecture non linéaire d'un document (navigation, surf).



FIGURE 1.13: Page Web

2. **Site Web** : C'est un ensemble de pages Web reliées les unes aux autres par des hyperliens (liens hypertextes).
3. **Page d'accueil** : C'est la page d'introduction d'un site Web qu'on peut aussi appeler « page de couverture ». Elle est très importante car elle détermine la première impression de l'internaute.
4. **Site Internet** : C'est un ensemble de données (texte, image, son, vidéo, etc.) Hébergées sur un serveur ayant une adresse IP et accessible aux internautes. Il existe différents types de sites selon leurs contenus et la nature des services qu'ils offrent. Citons les sites FTP, WEB, etc.
5. **Serveur Internet** : C'est l'ensemble des ressources matérielles (espace disque, connexion Internet, contenu du site, etc.) et logicielles (les programmes) qui servent les clients d'où émanent les demandes de services.
6. **URL (Uniform Resource Locator)** : C'est une adresse universelle d'une page WEB. Elle a le format suivant : Protocole utilisé : // nom du serveur/ chemin d'accès au fichier Une URL s'apparente à un nom de serveur et une information sur le type de protocole de réseau à utiliser pour atteindre la ressource.
7. **HTML (HyperText Markup Language)** : Les pages Web (ayant l'extension html) sont écrites en HTML. C'est un langage de balisage hypertextuel permettant d'inclure outre du texte, des graphiques, sons, vidéos et des pointeurs vers d'autres pages du même site ou d'un site distant.

```

<link rel="stylesheet" type="text/css" href="Content/light-theme.css" />
<link rel="stylesheet" type="text/css" href="Content/msls-light.css" />

<link rel="stylesheet" type="text/css" href="Content/jquery.mobile.structure-1.2.0.min.css" />
<link rel="stylesheet" type="text/css" href="Content/msls-1.0.0.min.css" />

<!-- Default font, header, and icon styles can be overridden in user-customization.css -->
<link rel="stylesheet" type="text/css" href="Content/user-customization.css"/>
</head>
<body>
<div id="msls-id-app-loading" class="ui-body-a msls-layout-ignore">
  <div class="msls-app-loading-img"></div>
  <span class="ui-icon ui-icon-loading"></span>
  <div class="ui-bottom-load">
    <div>EndToEndExample</div>
  </div>
</div>

<script type="text/javascript" src="//ajax.aspnetcdn.com/ajax/globalize/0.1.1/globalize.min.js"></script>
<script type="text/javascript" src="Scripts/winjs-1.0.min.js"></script>
<script type="text/javascript" src="Scripts/jquery-1.8.2.min.js"></script>
<script type="text/javascript" src="Scripts/jquery.mobile-1.2.0.min.js"></script>
<script type="text/javascript" src="Scripts/datajs-1.1.0.min.js"></script>
<script type="text/javascript" src="Scripts/Generated/resources.js"></script>
<script type="text/javascript" src="Scripts/msls-1.0.0.min.js"></script>
<script type="text/javascript" src="Scripts/Generated/generatedAssets.js"></script>
<script type="text/javascript">

```

FIGURE 1.14: Langage HTML

8. **Hypertexte** : C'est une forme non séquentielle non narrative d'organisation et d'accès à l'information représenté par un réseau de liens associatifs qui forment une structure d'hypertexte
9. **Hypermédia** : Le terme hypermédia désigne un hypertexte dont on insiste sur la nature multimédia de son contenu informationnel.
10. **Hyperdocument** : il désigne un ensemble de documents multimédia ayant une organisation et un fonctionnement hypertextuel.
11. **Navigateur** : C'est un logiciel qui offre une interface d'accès aux différentes informations disponibles sur le Web et l'exploration de ce dernier, Ce dernier interprète le texte et les commandes de formatage qu'il contient et affiche à l'écran la page correctement formatée.
12. **Moteur de recherche** : Un moteur de recherche est un programme qui indexe le contenu de différentes ressources Internet, et plus particulièrement de sites Web, qui permet à l'internaute de rechercher de l'information à partir d'un navigateur Web, selon différents paramètres, en ce servant de mots clés, et qui permet d'avoir accès à l'information ainsi trouvée.
13. **Lien hypertexte** : C'est un pointeur qui pointe sur un noeud de destination, souvent il est sous forme d'une icône, symbole, petite image, un mot dans un texte, etc. Il permet d'explorer le Web avec un simple clic de souris. On distingue :
 - Les liens internes qui renvoient à d'autres pages du même site
 - Les liens externes qui renvoient à d'autres pages d'un autre site
 - Les liens vers un signet qui renvoient à un signet de la même page ou d'une autre page
 - Les liens vers un e-mail qui permettent à l'internaute d'envoyer un e-mail directement depuis la même page.
 - Liens de téléchargement qui permettent de télécharger des documents depuis la même page.

1.4.4 Types de sites Web [5]

Les sites Web sont difficiles à classer, les limites entre les différents types varient en fonction de l'objectif et du public visé. En général on retrouve la classification ci-après :

1. **Sites personnels (homepage)** : Se caractérisent par leur manque d'uniformité, leur présentation est liée aux capacités et aux possibilités de leurs créateurs, qui peuvent être très

variés. Evidemment, il est difficile d'obtenir un tout harmonieux à partir de motifs provenant de styles variés. En ce qui concerne leur contenu, tous les thèmes sont traités.

2. **Sites d'informations** : Ces sites sont axés sur le texte offrant souvent de nombreux liens et peuvent se composer de plusieurs milliers de pages qui traitent des thèmes différents.
3. **Sites pour présentation d'entreprises** : Ils se composent généralement de quelques pages qui servent à présenter l'entreprise. La présentation est souvent rigide et le sérieux est de rigueur.
4. **Sites "boutiques"** : Ce sont des boutiques qui commercialisent leurs produits en ligne en offrant des informations sur chaque produit.
5. **Sites de bases de données** : Ces bases de données présentent leur contenu seulement sur demande, en remplissant un formulaire de requête.
6. **Sites portail** : Un portail est une collection de liens qui est conçue autour d'une fonction de recherche et se compose souvent de contenus prêtés

1.4.5 Les sites Web statiques [6]

Ce sont des pages Web écrites en HTML invariantes, proposées à l'avance, autrement dit, à chaque fois qu'un client affiche une page cette dernière se présente de la même manière tout au moins tant que le code HTML correspondant n'a pas été mis à jour par l'auteur.

Les fichiers de code HTML ne sont pas en mesure d'interagir avec le serveur. Si par exemple, on veut récupérer des informations liées à l'utilisateur pour les stocker dans une base de données, il va falloir mettre en oeuvre d'autres outils capables de supporter ce type d'interaction.

La mise à jour des pages HTML est très difficile et doit s'effectuer via un éditeur, de nombreux fichiers sont liés et la cohérence des liens est difficile à gérer.

1.4.6 Sites Web dynamiques

Les lourdeurs de la consultation et la mise à jour des pages Web statiques ainsi que le faible volume d'informations qu'elles contiennent ont rapidement conduit les concepteurs à proposer des «sites Web dynamiques». Dans ce cas, les pages ne sont pas fabriquées à l'avance. Cette composante dynamique peut recouvrir des aspects tels que la diffusion d'informations évoluant dans le temps ou la récupération des renseignements saisis par l'utilisateur pour décider de déclencher telle ou telle autre action.

Les auteurs chargés de la conception de telles pages Web doivent faire appel à des ressources de programmation dépassant le cadre du HTML, qui connaît quelques limitations dont la principale est de proposer des pages statiques. Ces sources de programmation peuvent être des langages de scripts, les outils d'interfaçage Web /base de données ou autre (que nous allons aborder plus tard). Voici une liste forcément incomplète des fonctionnalités dynamiques de base :

- Pages d'accueil personnalisables.
- Accès à des bases de données.
- Transactions commerciales.
- Formulaires de renseignements.
- Services d'inscription.

Conclusion :

L'évolution des technologies subit actuellement une accélération exponentielle, et Internet est un atout majeur qui contribue activement à cette évolution c'est pour ça qu'il est vivement recommandé d'intégrer cette dernière à toutes les infrastructures possible.

Chapitre 2

Le système LMD

Introduction :

La situation, passée, actuelle et future de l'enseignement supérieur en Algérie suscite depuis quelques années l'intérêt et la préoccupation de tous. Tous les acteurs directs (enseignants, étudiants et administration) ainsi que la société dans son ensemble s'accordent unanimement sur l'état de "dysfonctionnement" et de "crise" de l'université algérienne et soulignent, par conséquent, la nécessité et l'urgence de la doter de moyens pédagogiques, scientifiques, humains, matériels et structurels qui lui permettent de répondre aux attentes de la société et de s'intégrer au système international de l'enseignement supérieur.

Et pourtant, si l'on considère la période récente, pas moins de trois réformes ont été engagées au niveau de l'université. La réforme de la gestion de l'université (Loi d'orientation de l'enseignement supérieur du 17 - 08 - 1998) mise en application en 2003, la réforme des enseignements du supérieur adoptée en 2002 (Introduction du nouveau système LMD et enfin le statut particulier des enseignants, en discussion actuellement).

Bien que deux années d'application semblent insuffisantes pour faire le bilan du nouveau système, il nous semble, cependant, intéressant de faire un état des lieux par l'exposé des expériences de certaines universités. On insistera sur les difficultés rencontrées par ces universités pour s'adapter aux exigences du nouveau système et on essayera de donner quelques éléments de réponses quant aux espoirs suscités par l'adoption de ce système par certaines universités et facultés et les raisons qui expliquent les réticences des autres ?

2.1 Présentation du système LMD

2.1.1 Les objectifs de la réforme

Théoriquement, le nouveau système est censé répondre à quelques préoccupations de l'université algérienne en poursuivant les objectifs suivants :

- améliorer la qualité de la formation universitaire ;
- encourager le travail personnel de l'étudiant ;
- faciliter la mobilité et l'orientation des étudiants en garantissant la capitalisation et le transfert des acquis ;
- proposer des parcours de formation diversifiés et adaptés ;
- faciliter l'insertion professionnelle des étudiants en ouvrant l'université sur le monde extérieur ;
- consacrer le principe de l'autonomie des établissements universitaires ;
- unifier le système (architecture, diplômes, durée...) dans toutes les disciplines aux niveaux national, et international ;
- encourager et diversifier la coopération internationale.

2.1.2 Organisation des enseignements en LMD

Le dispositif LMD est organisé en trois paliers :

- Le niveau **LICENCE** correspondant à un cycle de formation de trois années après le BAC.
- Le niveau **MASTER** correspondant à deux années supplémentaires après le niveau Licence.
- Le niveau **DOCTORAT** correspondant à trois années supplémentaires après le niveau Master.

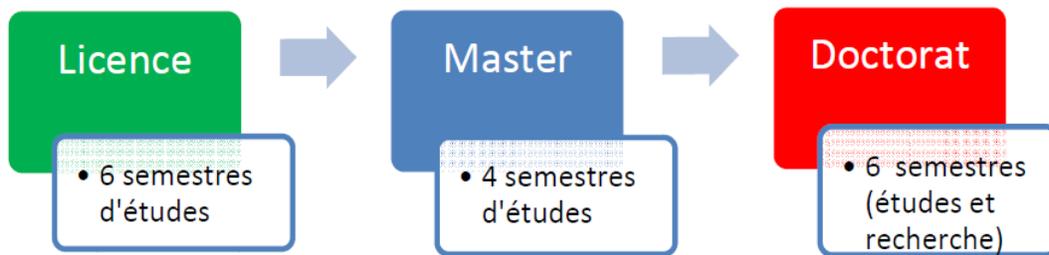


FIGURE 2.1: Architecture du LMD

dont la gestion pédagogique met en évidence des éléments nouveaux tels que :

- **Le principe du semestre** : Les études sont organisées en semestres afin d'assouplir l'organisation des parcours de formation. La licence est ainsi assurée en 6 semestres, dont 4 semestres sont dits de tronc commun. Il faut cependant préciser que l'évaluation et la progression des étudiants est annuelle. La durée moyenne d'un semestre est de 16 à 18 semaines. Le volume horaire hebdomadaire de formation présentielle varie de 20 heures pour les sciences humaines et sociales à 25 heures pour les sciences et technologies.
- **Les Unités d'Enseignement (UE)** : Dans chaque semestre les enseignements sont regroupés en trois Unités d'Enseignement :
 1. *Unité Fondamentale (UEF)* : regroupe les matières fondamentales pour une discipline donnée .
 2. *Unité de découverte (UED)* : concerne l'enseignement de matières correspondant à d'autres spécialités, voire d'autres champs disciplinaires afin d'élargir la culture universitaire et faciliter les passerelles de réorientation.
 3. *Unité Méthodologique (UEM)* : regroupe les matières d'enseignement d'outils méthodologiques destinés à aider l'apprenant à réaliser son parcours de formation (mathématiques, Langues, Informatique, Recherche Documentaire...)

Les 3 unités d'enseignements sont organisées de manière à permettre à la fois "une orientation progressive , une spécialisation adaptée aux vœux et capacités et des passerelles assurant une réorientation éventuelle de l'étudiant".

- **Les crédits** : A chaque Unité d'Enseignement correspond un nombre de crédits capitalisables et transférables pour chaque UE acquise par l'étudiant.

Un crédit comprend globalement entre 20 et 25 heures de travail de l'étudiant dans une matière donnée. Le nombre de crédits cumulés pour toutes les UE d'un semestre est de 30. On peut donc schématiser l'architecture de formation comme suit :

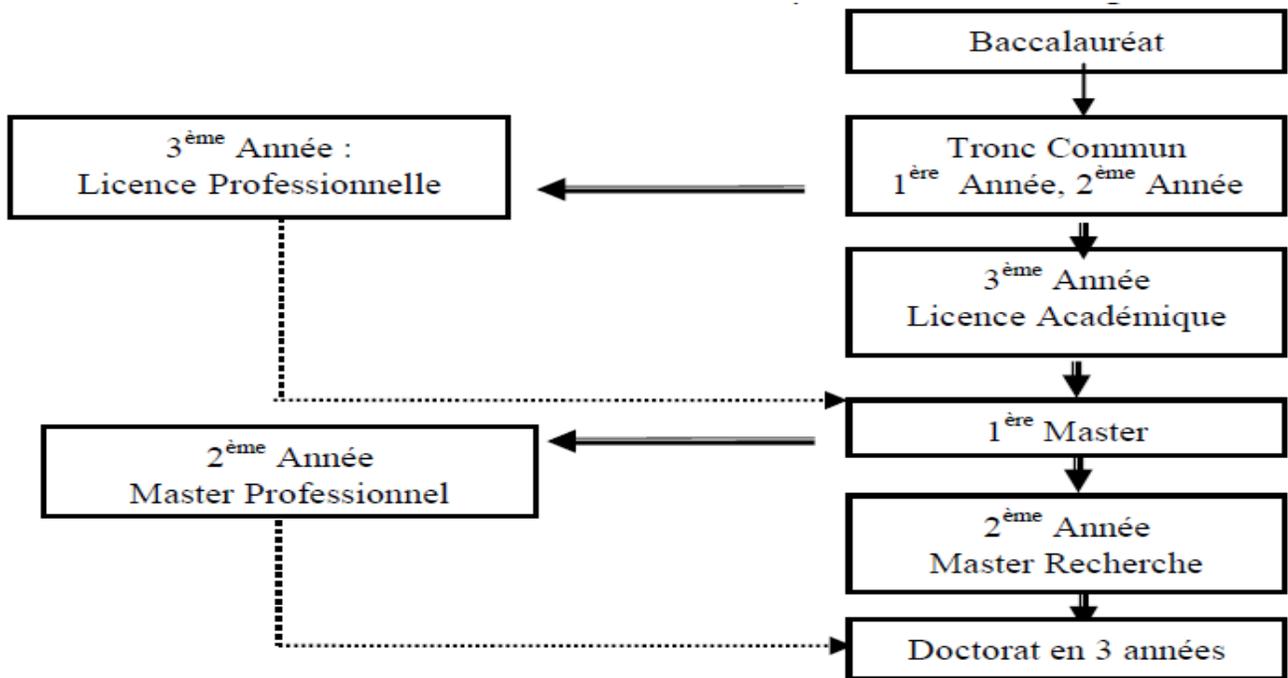


FIGURE 2.2: Architecture du système LMD en Algérie

- **Les domaines d'études** : Une des nouveautés du système LMD a été la réduction des spécialités ou domaines d'études. Le nouveau système retient seulement 13 domaines. Cependant, cette diminution du nombre de domaines ne veut en aucun cas dire qu'il y ait une restriction des spécialités. A l'inverse de l'ancien système, le système LMD donne la possibilité à l'établissement universitaire de créer un nombre indéfini de licences dans le cadre d'un domaine.
- **Les mentions ou filières** : Le domaine est décliné en filières ou mentions. A l'instar du domaine, les filières sont arrêtées par l'administration centrale et ne peuvent donc être modifiées. A cet effet, le ministère de l'enseignement supérieur a mobilisé dans la hâte les doyens des facultés, et quelques professeurs, qui en commissions, ont arrêté dans le domaine de spécialité un certain nombre de filières rendues par la suite légales.
- **Le tronc commun** : En Licence, il est prévu un tronc commun qui s'étale sur la quasi-totalité des deux premières années. C'est un programme unique au domaine et arrêté par l'administration centrale à l'issue des commissions de domaines. Il s'agit donc d'un programme national fixant les unités d'enseignement, leur contenu (matières, cours, TP, TD...), les crédits qui leur sont affectés ainsi que les matières qui les composent. L'autonomie de l'établissement est reléguée à un second plan. Cependant, la faculté n'est pas astreinte à un contenu précis des matières, pourvu qu'elle garde l'intitulé officiel du module.

2.2 Historique de la réforme LMD [8]

La réforme de LMD « Licence-Master-Doctorat » désigne un ensemble de mesures modifiant le système d'enseignement supérieur pour l'adapter aux standards européens. Elle met en place principalement une architecture basée sur trois grades : licence, master et doctorat, une organisation des enseignements en semestres et unités d'enseignement, la mise en oeuvre des crédits européens et par la délivrance d'une annexe descriptive au diplôme. Les textes fondateurs de cette réforme

sont parus en 2002, mais celle-ci s'est étalée sur plusieurs années, et en 2014 certaines formations, notamment celles de la santé n'ont pas été modifiées (Médecine, chirurgie dentaire, Pharmacie, ...).

Le **25 mai 1998**, les ministres en charge de l'enseignement supérieur de la France, de l'Allemagne, du Royaume-Uni et de l'Italie se sont réunis à la Sorbonne et ont fait une déclaration commune en vue d'**harmoniser l'architecture du système européen d'enseignement supérieur**.

Le **19 juin 1999** à **Bologne**, les ministres de l'éducation de 29 pays européens poursuivent la réflexion sur la base de la déclaration de la Sorbonne et se fixent une série d'objectifs dont la réforme actuelle est l'aboutissement.

Le **30 mars 2001** à **Salamanque**, plus de 300 institutions européennes se réunissent afin de rappeler les principes d'harmonisation du système européen d'enseignement supérieur et de préparer la conférence de Prague.

Enfin, le **19 mai 2001** à **Prague**, la déclaration commune des Ministres européens de l'Éducation réaffirme la volonté de continuer les efforts sur les six points principaux de la déclaration de Bologne :

1. Adoption d'un système de reconnaissance rendant les diplômes universitaires plus transparents et lisibles.
2. Mise en place de cursus universitaires fondés notamment sur un premier cycle de trois ans.
3. Introduction d'un système de crédits
4. Promotion de la mobilité des étudiants, des chercheurs ainsi que du personnel administratif.
5. Développement d'instruments communs permettant d'évaluer la qualité des enseignements.
6. Accroissement de la dimension européenne du contenu des cursus universitaires.

La réforme LMD (Licence/Mastère/Doctorat) est entrée en vigueur à partir de la rentrée universitaire 2004-2005 en Algérie, et elle a touchée dans un premier temps 10 établissements de l'enseignement supérieur sur les 58 existants. Introduit de manière graduelle, le système LMD, licence, master, doctorat, commence à se généraliser à l'**université Mouloud Mammeri de Tizi-Ouzou (UMMTO)** à partir de **2005/2006**.

2.3 Evaluation et progression [9]

Les aptitudes et l'acquisition des connaissances, concernant chaque unité d'enseignement, sont appréciées semestriellement soit par un contrôle continu (TD, TP, Stages, Travail personnel, etc...) soit par un examen final soit par les deux modes de contrôle.

- L'unité d'enseignement est définitivement acquise pour tout étudiant ayant obtenus une note supérieure ou égale à 10/20 dans toute les matières qui la constituent, ou par compensation si la moyenne de l'ensemble des notes obtenues pondérées de leurs coefficient respectifs est égale ou supérieur a 10/20.
- L'exclusion d'un étudiant dans une matière composant une unité d'enseignement ne permet pas l'acquisition de cette dernière par le calcul de la moyenne des notes obtenues dans les autres matières qui la constituent.
- En cas d'échec à la première session, l'étudiant se présente à la session de rattrapage aux épreuves relatives aux unités d'enseignement non acquises. Dans ce cas, l'étudiant garde le bénéfice des matières acquises

- Dans le cas d’une unité d’enseignement acquise par la compensation, l’étudiant peut être autorisé à se présenter, en session de rattrapage, aux matières non acquises de ladite unité.
- Lors de la session de rattrapage, la note, pour chacune des matières concernées, est alors déterminée sur la base de la note obtenue à l’épreuve de rattrapage selon les modalités de contrôle des connaissances.
- La note finale retenue pour la matière sera la meilleure des moyennes entre la première session et la session de rattrapage.
- Dans le cas où une unité d’enseignement n’est pas acquise, les crédits affectés aux matières acquises qui la composent sont capitalisables.
- Le principe de compensation s’applique :
 1. A l’unité d’enseignement : il permet l’acquisition de l’unité d’enseignement par le calcul de la moyenne des notes des matières constitutives affectées de leur coefficient respectif.
 2. Au semestre : il permet l’acquisition du semestre par le calcul de la moyenne des notes des unités d’enseignement qui le composent, affectées de leur coefficient respectif.
 3. Au niveau (L1, L2, L3) : il permet l’acquisition du niveau (L1, L2 ou L3) par le calcul de la moyenne des notes des unités d’enseignement qui le composent, affectées de leur coefficient respectif.

2.3.1 La progression dans les études [9]

Le passage du premier au second semestre d’une même année universitaire dans un même parcours de formation est de droit pour tout étudiant régulièrement inscrit.

2.3.2 La progression dans les études de Licence [9]

- L’acquisition d’une année pédagogique dans le cycle licence est conditionnée par l’acquisition des semestres qui la compose.
- Cependant, le passage peut aussi se faire comme suit :
 1. De la première à la deuxième année de Licence peut être autorisé pour tout étudiant ayant acquis au minimum 30 crédits dont 1/3 au moins dans un semestre.
 2. De la deuxième à la troisième année peut être autorisé à tout étudiant ayant validé au minimum 90 crédits et acquis les unités d’enseignements fondamentales requises à la poursuite des études en spécialité.
- L’étudiant, non admis à progresser en deuxième ou en troisième année d’un parcours de formation, est, selon le cas, autorisé à se réinscrire dans le même parcours ou orienté, par l’équipe de formation, vers un autre parcours de formation.
- La procédure d’orientation fait autant que possible l’objet d’une application prioritaire pour les étudiants en situation d’échec dans leur parcours de formation initial. Elle doit conduire, par le biais de passerelles, à la construction d’un parcours individualisé plus conforme aux aptitudes de l’étudiant et devrait lui permettre une meilleure progression dans son cursus d’étude.
- En aucun cas, l’étudiant inscrit en licence ne peut y séjourner plus de cinq années au maximum, même dans le cas d’une réorientation.

2.3.3 La progression dans les études de Master [9]

le passage de la première à la deuxième année est de droit si l'étudiant a acquis les deux premiers semestres du cursus de formation.

Cependant, le passage de la première à la deuxième année peut être autorisé pour tout étudiant ayant validé au minimum 45 crédits et acquis les unités d'enseignements requises à la poursuite des études en spécialité.

Au aucun cas, l'étudiant inscrit en Master ne peut y séjourner plus de 03 années maximum, même dans le cas d'une réorientation.

2.4 L'algorithme général pour les deux système de progression

On distingue deux (2) systèmes de progression :

2.4.1 Premier système de progression

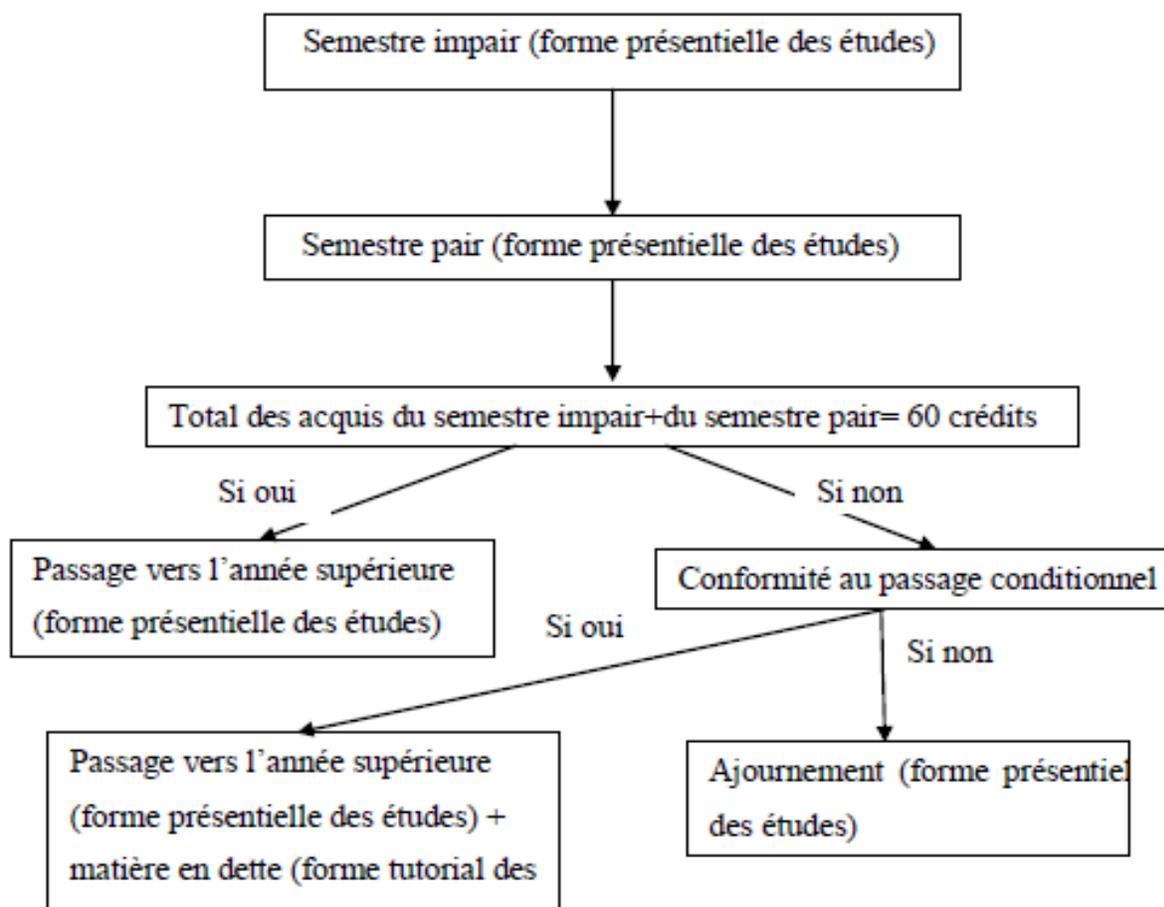


FIGURE 2.3: L'algorithme du premier système de progression

2.4.2 Deuxième système de progression

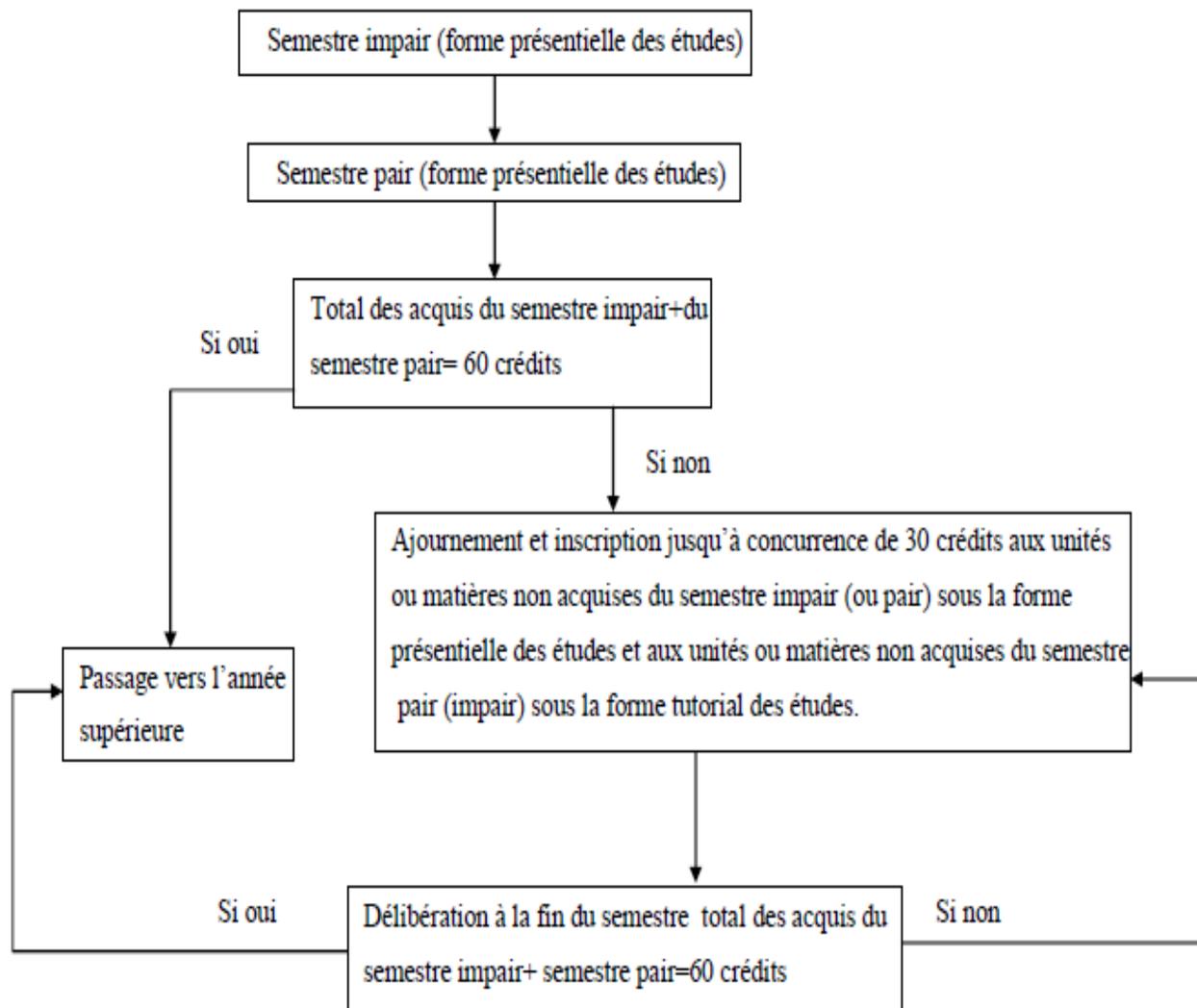


FIGURE 2.4: L'algorithme du deuxième système de progression

2.5 Forme présentielle des études et Forme Tutorial des études

2.5.1 Forme présentielle des études

Forme classique des études (cours, TD avec présence obligatoire).

2.5.2 Forme tutorial des études

- Prise en charge par un enseignant appelé tuteur principal ;
- Peut être secondé par des tuteurs secondaires dans le cas de sureffectif étudiant ;
- Le tuteur principal a pour tâches :

1. Afficher le programme détaillée .
2. Annoncer le(s) documents de référence .

3. Organiser une(s) séances de consultation hebdomadaires .
4. Répondre aux sollicitations des étudiants par courrier électronique .
5. Rédiger, organiser et corriger le(s) examens .
6. Remettre le PV des résultats a l'administration.

2.6 Présentation de département informatique de l'Université Mouloud Mammeri (UMMTO)

2.6.1 Historique [10]

En **1977**, Le président Boumediène avait inauguré le centre universitaire de Tizi-Ouzou (C.U.T.O) d'après le décret exécutif N°17-77 du **20 Juin 1977**.

En **1978**, le C.U.T.O. avait démarré avec cinq (5) départements :

- Département de Biologie ;
- Département de Sciences Juridiques et Administratives ;
- Département de Langues et Littérature Arabes ;
- Département de Sciences Exactes ;
- Département des Sciences Economiques crée en **1978-1979**.

En **1984**, la nouvelle carte universitaire avait éclatée le C.U.T.O en neuf (09) instituts (I.N.E.S) dont celui d'Informatique :

- I.N.E.S des Sciences Juridiques et Administratives ;
- I.N.E.S. d'Agronomie ;
- I.N.E.S. de Biologie ;
- I.N.E.S. de Langue et Littérature Arabes ;
- I.N.E.S. de Génie Civil ;
- I.N.E.S. des Sciences Economiques ;
- I.N.E.S. des Sciences Médicales ;
- I.N.E.S. d'Electrotechnique ;
- **I.N.E.S. d'Informatique**

En 1989, le C.U.T.O devient une université à part entière avec huit (8) instituts.

Pendant les années 2000, le département Informatique est devenue l'un des quatre (4) départements de la faculté de Génie Electrique et Informatique de l'université Mouloud MAMMERI de Tizi-Ouzou (UMMTO) :

- Département d'Automatique ;
- Département d'Electronique ;
- Département d'Electrotechnique ;
- Département d'Informatique.

2.6.2 Les formations [10]

Le département reçoit des étudiants, ayant passé avec succès les quatre semestres du domaine Sciences et Techniques et après étude des fiches de vœux et sur la base des places pédagogiques disponibles, dans les licences et Masters suivants :

2.6.2.1 Licence

1. Licence Informatique
2. Licence Informatique et Multimédia

2.6.2.2 Master

1. Master Conduite de Projets Informatiques ;
2. Master Réseaux Mobilités et Systèmes Embarqués ;
3. Master Systèmes Informatiques ;
4. Master Ingénierie des Systèmes d'Information (Master Professionnel) .

Remarque : Le tronc commun Mathématiques et Informatiques (L1) est géré par la faculté des sciences .

Conclusion :

Toutefois les règles de progression sont compliquées et justifient l'utilisation de l'outil informatique permettant d'automatiser le calcul de moyenne et le passage. Nous allons présenter dans le chapitre suivant les détails de conception de cet outil.

Chapitre 3

Analyse et Conception

Introduction :

Dans le but d'une meilleure organisation et avant toute réalisation d'une application informatique, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou démarche bien définie, en mettant en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité. Dans ce chapitre nous allons entamer le développement de notre application par une analyse qui mettra en évidence les différents acteurs intervenants dans le système ainsi que leurs besoins.

La phase conception, s'appuyant sur les résultats de cette analyse, donnera la modélisation des objectifs à atteindre.

Des progrès énormes ont été consentis dans le but d'avoir une meilleure analyse et de rendre la conception plus complète. L'approche objet, s'est avérée un modèle d'analyse et de conception très puissant et se trouve de plus en plus utilisée. Pour cela nous avons adopté la conception avec l'UML (Unified Modelling language) qui permet de bien représenter la dynamique d'une application par la série des diagrammes qu'il offre.

L'objectif de l'application est de gérer les différentes tâches de la scolarité LMD du département Informatique UMMTO.

3.1 Rappel sur UML

La description de la programmation par objets a fait ressortir l'étendue du travail conceptuel nécessaire : définition des classes, de leurs relations, des attributs et méthodes, etc.

Pour réaliser une application, il ne convient pas de se lancer tête baissée dans l'écriture du code : il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et étapes de la réalisation. C'est cette démarche antérieure à l'écriture que l'on appelle modélisation ; son produit est un modèle.

UML (Unified Modeling Language) est la synthèse des différentes notations que l'on retrouve dans OMT, OOSE, Booch, qui recouvrent à elles trois plus de la moitié du marché des méthodes objets. UML que ses concepteurs ont voulu simple, est plus intuitive, plus expressive, plus cohérent, plus homogène que les autres méthodes. Elle est en particulier conçue pour être lisible sur des supports courants et variés, comme le papier, les écrans d'ordinateur, les tableaux blancs, etc.

UML propose neuf (9) diagrammes de modélisation, répartis sur trois (3) axes du niveau conceptuel :

1. Fonctionnel.
2. Statique.
3. Dynamique.

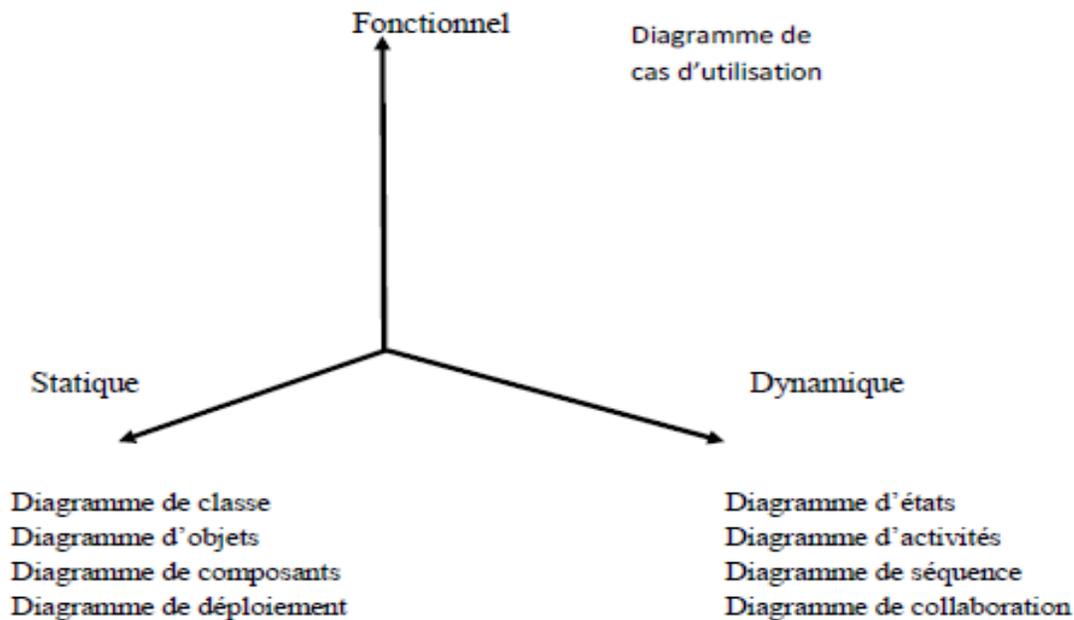


FIGURE 3.1: Composantes de la modélisation UML.

3.2 Identification des acteurs

Notre système communique avec des acteurs afin de réaliser un ensemble d'opérations. Un acteur représente un rôle que peut jouer l'utilisateur dans le système. L'acteur est associé à un cas d'utilisation (voir la section suivante), c'est-à-dire qu'il peut interagir avec lui et participer à son scénario, il est représenté par un personnage stylisé.

Les acteurs de notre système sont :

1. L'Enseignant : C'est un rôle qui peut être joué par tous les enseignants du département ;
2. L'Etudiant : C'est un rôle qui peut être joué par tous les étudiants ;
3. L'Agent de la scolarité : c'est un rôle qui peut être joué par la personne chargée des services de la scolarité ;
4. Administrateur : c'est un rôle qui peut être joué par la personne qui est chargée de l'administration de la base de données .

3.3 Identification des cas d'utilisation

Un cas d'utilisation (USE CASE) modélise une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système.

Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur.

Pour notre cas, nous avons pu dégager les cas d'utilisation suivants :

Acteurs	Cas d'utilisation (Tâche)
Enseignant	T0 : Se connecter au site. T1 : Accéder a son espace personnel. T2 : S'authentifier. T3 : Edition des PV modulaires. T4 : Affichage des PV modulaires. T5 : Consultation des PV de délibérations. T6 : Changer son mot de passe.
Etudiant	T7 : Se connecter au site T8 : Consulter les notes
Agent de la Scolarité	T9 : Se connecter au site. T10 : Accéder a son espace privé. T11 : S'authentifier. T12 : Edition des listes des étudiants. T13 : Edition des relevés de notes.. T14 : Edition des PV globaux de délibérations. T15 : Changer son mot de passe.
Administrateur	T16 : Se connecter au site. T17 :Accéder à son espace personnel. T18 : S'authentifier. T19 : Création de comptes. T20 : Modification de comptes. T21 : Suppression de comptes T 22 : Changer son mot de passe.

3.4 Spécification des scénarios :

3.4.1 Définition d'un scénario :

Un scénario est une succession particulière d'enchaînement qui s'exécute du début à la n du cas d'utilisation. Un cas d'utilisation peut avoir une ou plusieurs instances rerésentées par des scénarios de chacun des acteurs qu'on a dénis, eectue des tâches et chaque tâche est décrite par des scénarios qu'on résume comme suit :

Tâche	Scénario
T0 : Se connecter au site.	S0 : Saisir l'URL dans le navigateur
T1 : Accéder à son espace personnel.	S1 : Cliquer sur le lien Connexion
T2 : S'authentifier.	S2 : Accéder au formulaire. S3 : Saisir le mot de passe et login. S4 : Valider le login et le mot de passe
T3 : Edition des PV modulaires.	S5 : Sélectionner le lien « Editer PV ». S6 : Remplir le formulaire. S7 : Valider la saisie.
T4 : Affichage des PV modulaires	S8 : Sélectionner le lien « PV Modulaire ». S9 : Choisir module . S10 : Valider.
T5 : Consultation des PV de délibérations.	S11 : Sélectionner le lien « PV Délibération ». S12 : Valider.
T6 : Changer son mot de passe.	S13 : Cliquer sur le lien "Changer mot de passe" S14 : Remplir le formulaire . S15 : Valider le formulaire.
T7 : Se connecter au site .	S16 : Saisir l'URL du site sur le navigateur.
T8 : Consulter ces notes.	S17 : Sélectionner le lien « consultation des PV délibérations ».
T9 : Se connecter au site.	S18 : Saisir l'URL du site dans le navigateur.
T10 : Accéder a son espace privé.	S19 : Cliquer sur le lien "connexion"
T11 : S'authentifier.	S20 : Accéder au formulaire. S21 : Saisir le mot de passe et login. S22 : Valider le login et le mot de passe
T12 : Edition des listes des étudiants.	S23 : Cliquer sur le lien "Liste des etudiants " S24 : Valider .
T13 : Edition des relevés de notes.	S25 : Sélectionner "Relevé de notes " S26 : Sélectionner l'étudiant . S27 : Valider la sélection.
T14 : Edition des PV de délibérations	S28 : Cliquer sur le lien " PV de délibération"
T15 : Changer son mot de passe.	S29 : Cliquer sur le lien "Changer le mot de passe" S30 : Remplir le formulaire. S31 : Valider le formulaire.
T16 : Se connecter au site.	S32 : Saisir l'URL du site dans le navigateur.
T17 : Accéder à son espace personnel.	S33 : Cliquer sur le lien "Connexion"
T18 : S'authentifier .	S34 : Accéder au formulaire. S35 : Saisir le login et le mot de passe. S36 : Valider le login et le mot de passe.
T19 : Création de comptes	S37 : Cliquer sur le lien "Création de comptes " S38 : Remplir le formulaire. S39 : Valider le formulaire.
T20 : Modification de comptes.	S40 : Cliquer sur le lien 'Modification de comptes' S41 : Remplir le formulaire. S42 : Valider le formulaire.
T21 : Suppression de comptes	S43 : Cliquer sur le lien 'Suppression de comptes' S44 : Remplir le formulaire. S45 : Valider le formulaire.
T22 : Changer le mot de passe.	S46 : Cliquer sur le lien 'changer le mot de passe' S47 : Remplir le formulaire. S48 : Valider le formulaire.

3.5 Spécification des cas d'utilisation (Use case)

Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il modélise un service rendu par le système, sans imposer le mode de réalisation de celui-ci , Voici quelques exemples :

3.5.1 Cas d'utilisation "S'authentifier"

Use Case : S'authentifier.
 Scénarios : S2,S3,S4 / S20,S21,S22 / S34,S35,S36
 Acteurs : Enseignant, Agent de la scolarité, Administrateur
 Description :
 1- L'utilisateur saisit l'URL du site dans son navigateur.
 2- Le système achemine la page d'accueil.
 3- L'utilisateur clique sur le lien Connexion.
 4- Le système affiche le formulaire d'authentification .
 5- L'utilisateur saisit le login et le mot de passe .
 6- L'utilisateur clique sur le bouton Connexion .
 7- Le système affiche son espace, sinon il affiche un message d'erreur et renvoie le formulaire

3.5.2 Cas d'utilisation "Edition des PV Modulaire"

Use case : Edition des PV modulaire.
 Scénarios : S5,S6,S7
 Acteurs : Enseignant .
 Description :
 1- L'enseignant atteint son espace personnel .
 2- L'enseignant clique sur le lien Edition PV modulaire.
 3- Le système affiche la liste des étudiants.
 4- L'enseignant saisit les notes de chaque étudiant puis valide sa demande en cliquant sur le bouton Valider.
 5- Le système affiche un message de confirmation .

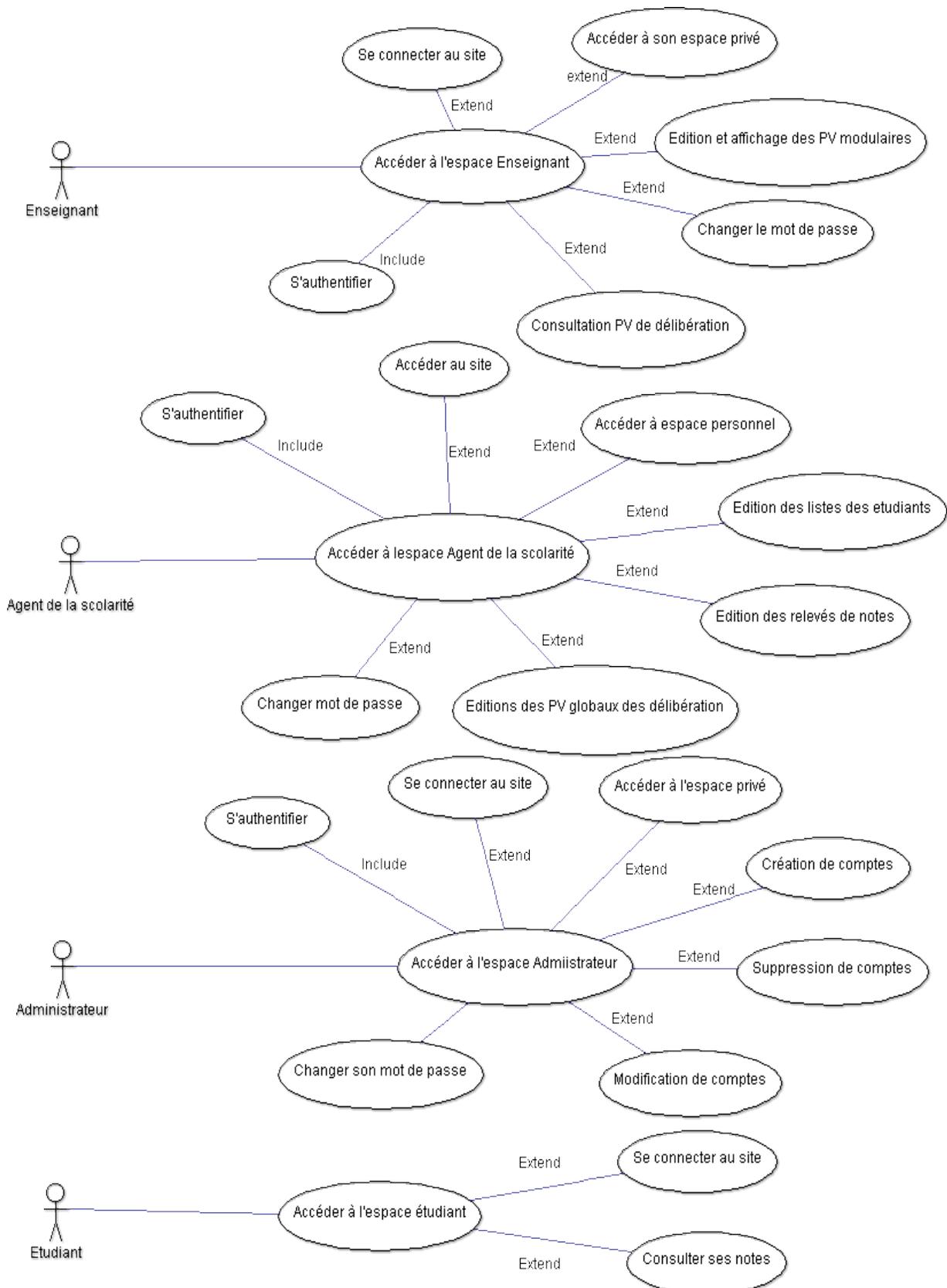
3.5.3 Cas d'utilisation "Edition des relevés de notes"

Use case : Edition des relevés de notes.
 Scénarios : S25,S26,S27
 Acteur : Agent de la scolarité
 Description :
 1- L'agent de la scolarité atteint son espace personnel .
 2- Le système affiche la liste des étudiants.
 3- L'agent choisit un étudiant parmi la liste et valide la sélection.
 4- Le système affiche Le relevé de notes l'étudiant

3.5.4 Cas d'utilisation "Changer le mot de passe"

Use Case : Changer le mot de passe
 Scénarios : S13,S14,S15 / S29,S30,S31 / S46,S47,S48
 Acteurs : Enseignant, Agent de Scolarité, Administrateur
 Description :
 1- L'utilisateur atteint son espace personnel et clique sur le lien Changer le mot de passe
 2- Le système affiche le formulaire de changement de mot de passe.
 3- L'utilisateur remplit le formulaire et soumet la requête .
 4- Le système envoie un message de confirmation si le mot de passe a été changé, sinon il affiche un message d'erreur.

3.6 Diagramme des cas d'utilisation globale de l'application gestion de la scolarité LMD



Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets mis en oeuvre par le système.

La relation Include : Une relation inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B, signifie qu'une instance de A contient le comportement décrit dans B, le cas d'utilisation A ne peut pas être utilisé seul.

La relation Extend : Une relation d'extension d'un cas d'utilisation A par rapport à un cas d'utilisation B, signifie qu'une instance de A peut être étendue par le comportement décrit dans B.

3.7 Diagramme de séquence détaillée cas utilisation

Selon le formalisme UML (Unified Modeling Language), les diagrammes de séquences détaillés représente graphiquement les interactions (acteurs-systèmes) dans l'évolution temporelle. les acteurs interagissent donc avec les pages clients et les pages serveurs qui sont utiles pour les interactions avec les ressources du serveur.

Nous allons décrire ci-dessous quelques diagrammes de séquences cas utilisation important.

3.7.1 Diagramme de séquence détaillé cas utilisation « S'authentifier »

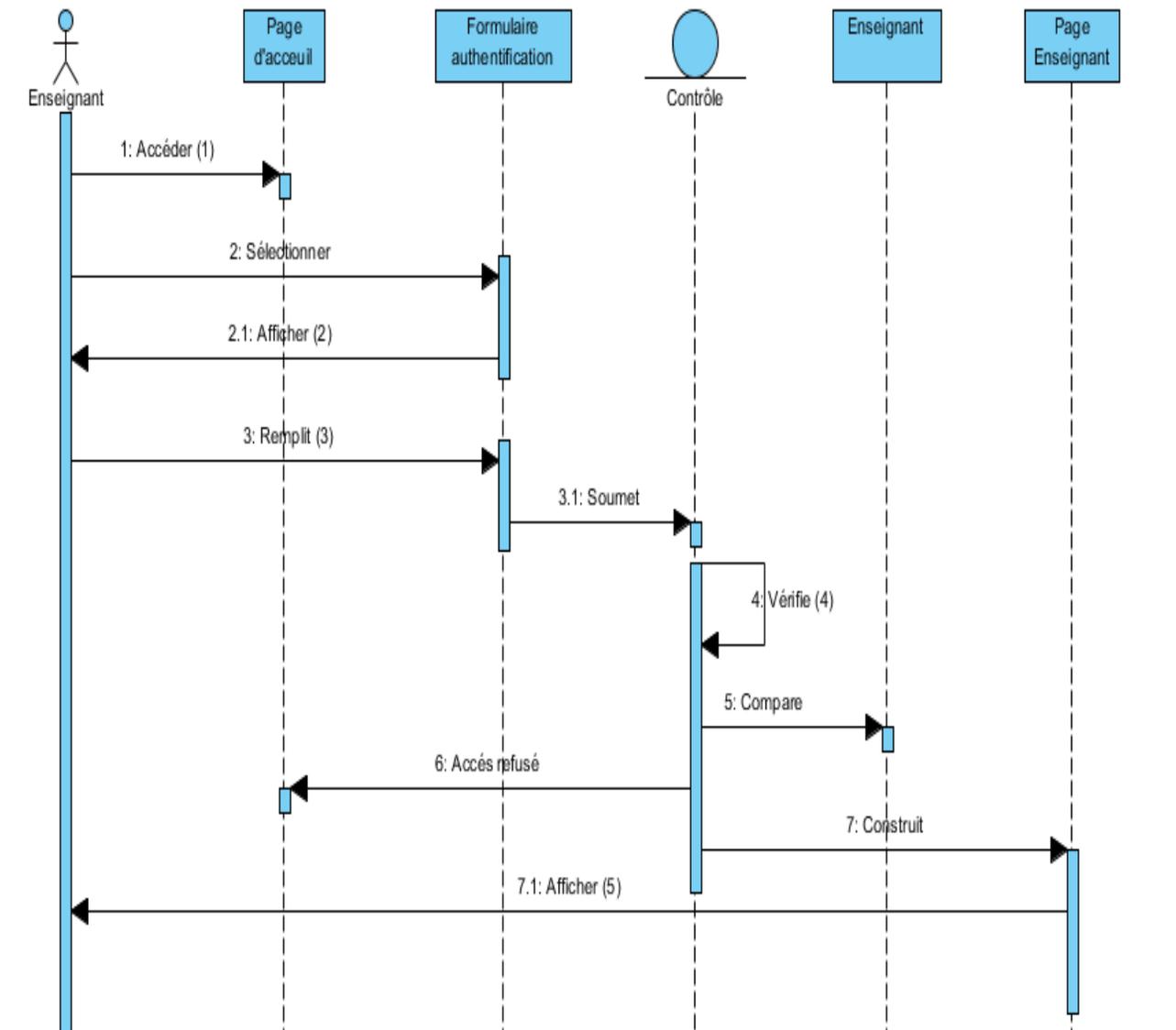


FIGURE 3.3: Diagramme de séquence “s’authentifier Enseignant “

- (1) L’acteur Enseignant saisit l’URL de l’application puis valide la saisie le système alors construit la page d’accueil de l’application puis l’affiche.
- (2) L’acteur Enseignant clique sur le lien Authentification alors le système construit la page authentification puis l’affiche.
- (3) L’acteur Enseignant saisit son login et son mot de passe dans le formulaire authentification puis le valide en appuyant sur le bouton valider, le système vérifie alors les données saisies dans la base de données ensuite il construit la page Espace enseignant puis l’affiche. Dans le cas où les données login et/ou mot de passe ne sont pas juste le système construit un message d’erreur puis l’affiche.

3.7.2 Diagramme de séquence détaillé cas utilisation « Changement de mot de passe »

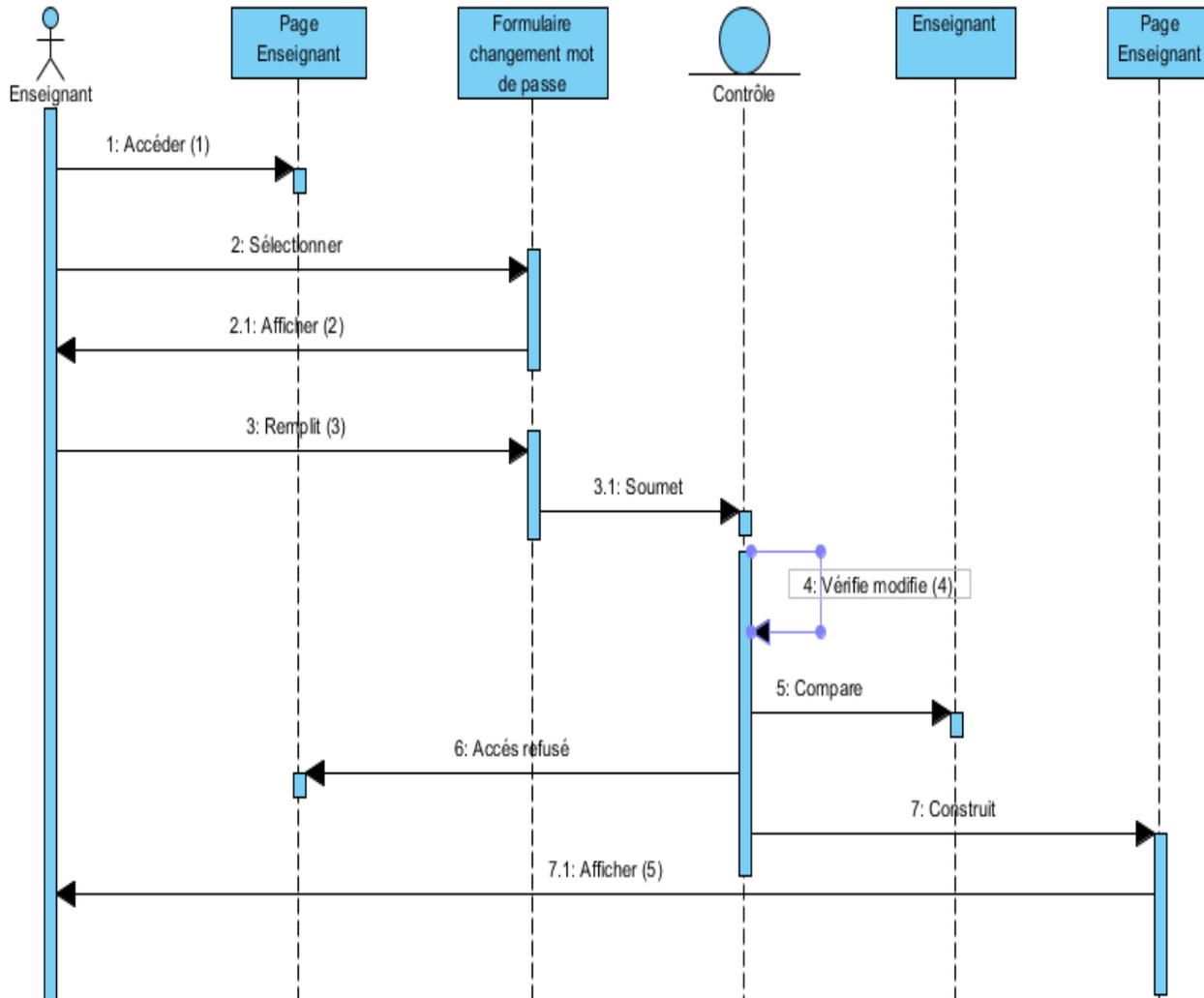


FIGURE 3.4: Diagramme de séquence “Changer le mot de passe”

(1) (2) L’acteur Enseignant clique sur le lien Changement de mot passe qui se trouve dans son espace, le système affiche alors le formulaire du changement de mot passe.

(3) (4) L’acteur Enseignant saisit son login et son mot de passe actuelle ainsi que son nouveau mot de passe et la confirmation du nouveau mot de passe dans le formulaire de changement du mot de passe puis valide la saisie en appuyant sur le bouton valider, le système vérifie alors les données saisies dans la base de données puis procède a la modification du mot de passe si les données saisie sont exactes, sinon il construit un message d’erreur puis l’affiche.

(5) Après avoir modifier le mot de passe dans la base de données le système construit la page de confirmation puis l’affiche.

3.7.3 Diagramme de séquence détaillé cas utilisation « Consultation des notes »

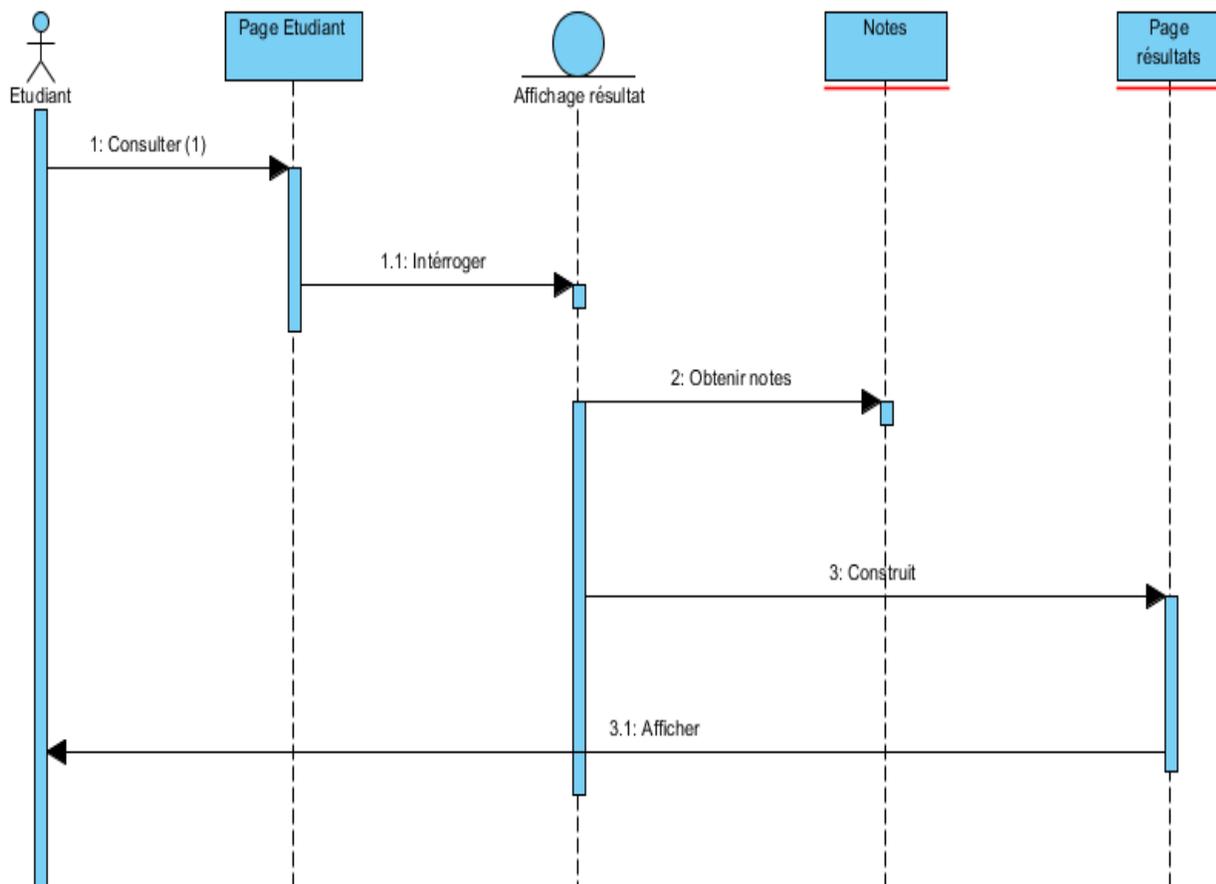


FIGURE 3.5: Diagramme de séquence “consulter notes “

(1) L'acteur **Etudiant** clique sur le lien **Consultation des notes** qui se trouve dans son espace, le système obtient alors les données depuis la base de données puis il construit la page **Résultats**, ensuite il l'affiche.

3.7.4 Diagramme de séquence détaillé cas utilisation « Edition des PV modulaires »

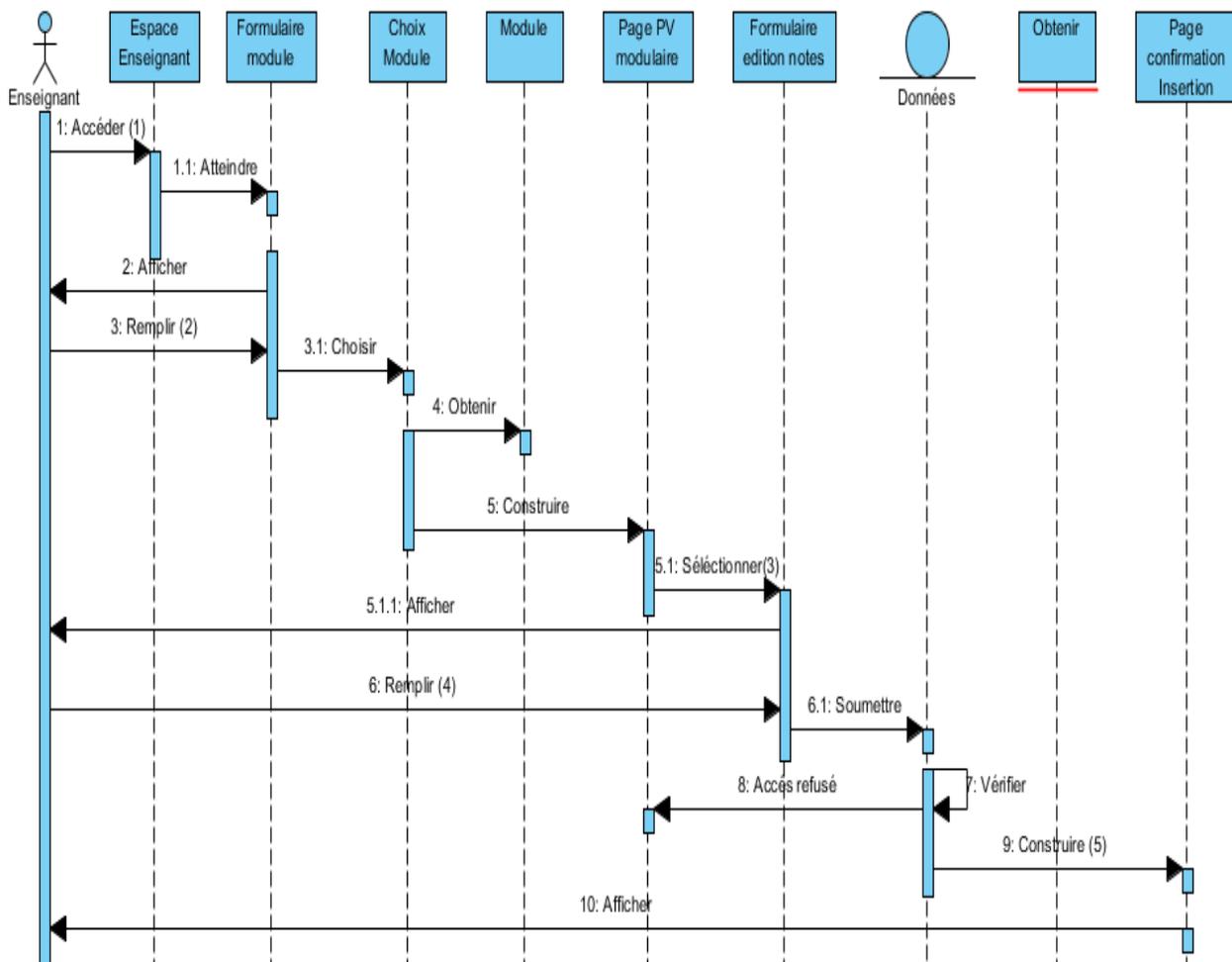


FIGURE 3.6: Diagramme de séquence “Edition PV modulaire “

- (1) L'acteur Enseignant clique sur le lien PV Modulaire qui se trouve dans son espace, le système construit alors le formulaire choix du module et l'affiche.
- (2) L'acteur Enseignant choisit alors son module puis le valide en cliquant sur le bouton Valider , le système choisit alors le module dans la base de données puis construit la page PV modulaire en rapport avec le module et l'affiche.
- (3) L'acteur Enseignant clique alors sur le lien Editer PV , le système construit alors le formulaire édition note puis l'affiche.
- (4) L'acteur Enseignant saisie le numéro et la note de l'étudiant, le système vérifie alors les données saisies dans la base de données puis construit un message message d'erreur si les données sont erronées puis l'affiche.
- (5) Si les données sont vérifiées le système les inserts alors dans la base de données puis construit une page de confirmation et l'affiche.

3.7.5 Diagramme de séquence détaillé cas utilisation « Affichage des PV modulaires »

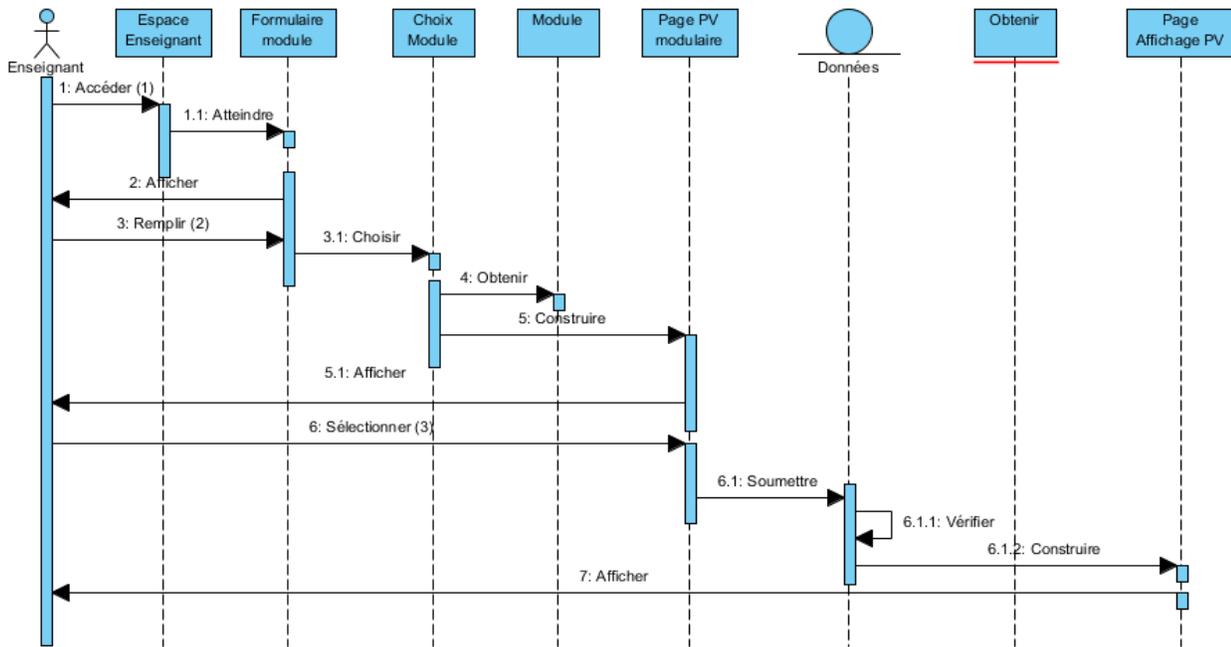


FIGURE 3.7: Diagramme de séquence “affichage PV modulaire “

- (1) L'acteur Enseignant clique sur le lien PV Modulaire qui se trouve dans son espace, le système construit alors le formulaire choix du module et l'affiche.
- (2) L'acteur Enseignant choisit alors son module puis le valide en cliquant sur le bouton Valider , le système choisit alors le module dans la base de données puis construit la page PV modulaire en rapport avec le module et l'affiche.
- (3) L'acteur Enseignant clique sur le lien afficher PV, le système obtient alors les données le la base de données puis construit une page affichage du PV modulaire et l'affiche.

3.7.6 Diagramme de séquence détaillé cas utilisation « Création de comptes »

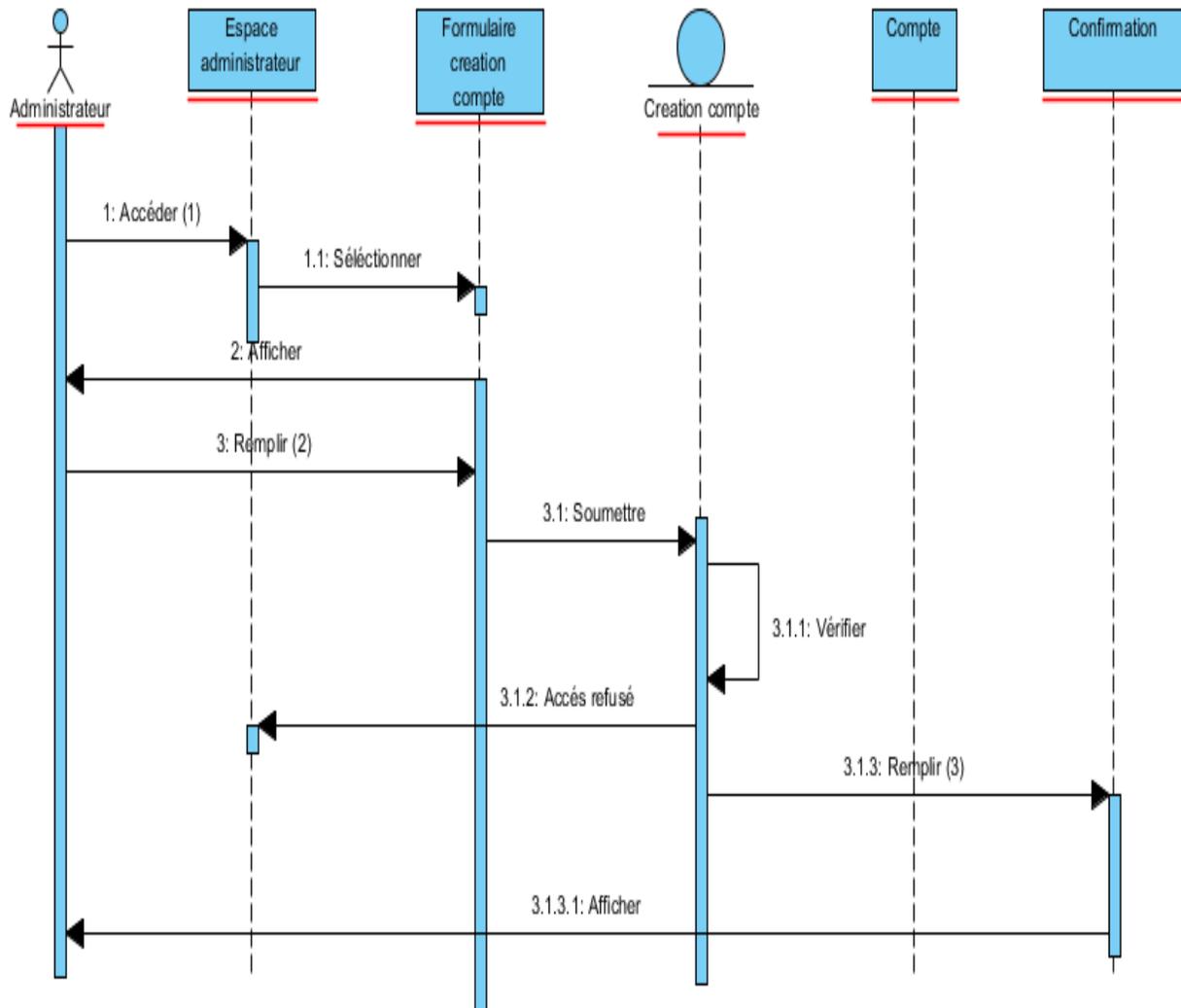


FIGURE 3.8: Diagramme de séquence “Création de comptes “

- (1) L'acteur Administrateur clique sur le lien Création de compte qui se trouve dans son espace, le système construit alors le formulaire création compte puis l'affiche.
- (2) L'acteur Administrateur remplit alors le formulaire puis le valide la saisie avec le bouton Valider, le système vérifie alors les données dans la base de données puis construit un message d'erreur et l'affiche si les données sont pas en norme ou si le compte existe déjà.
- (3) Si les données sont dans les normes et le compte n'existe pas alors le système crée le compte et insert les données dans la base de données puis construit une page de confirmation et l'affiche.

3.8 Diagramme d'activités

Un diagramme d'activités est semblable à un organigramme. Il se focalise sur le flot des activités impliquées dans un processus unique, il permet la représentation graphique du comportement d'une méthode ou le déroulement d'un cas d'utilisation.

3.8.1 Diagramme d'activités du cas d'utilisation « S'authentifier »

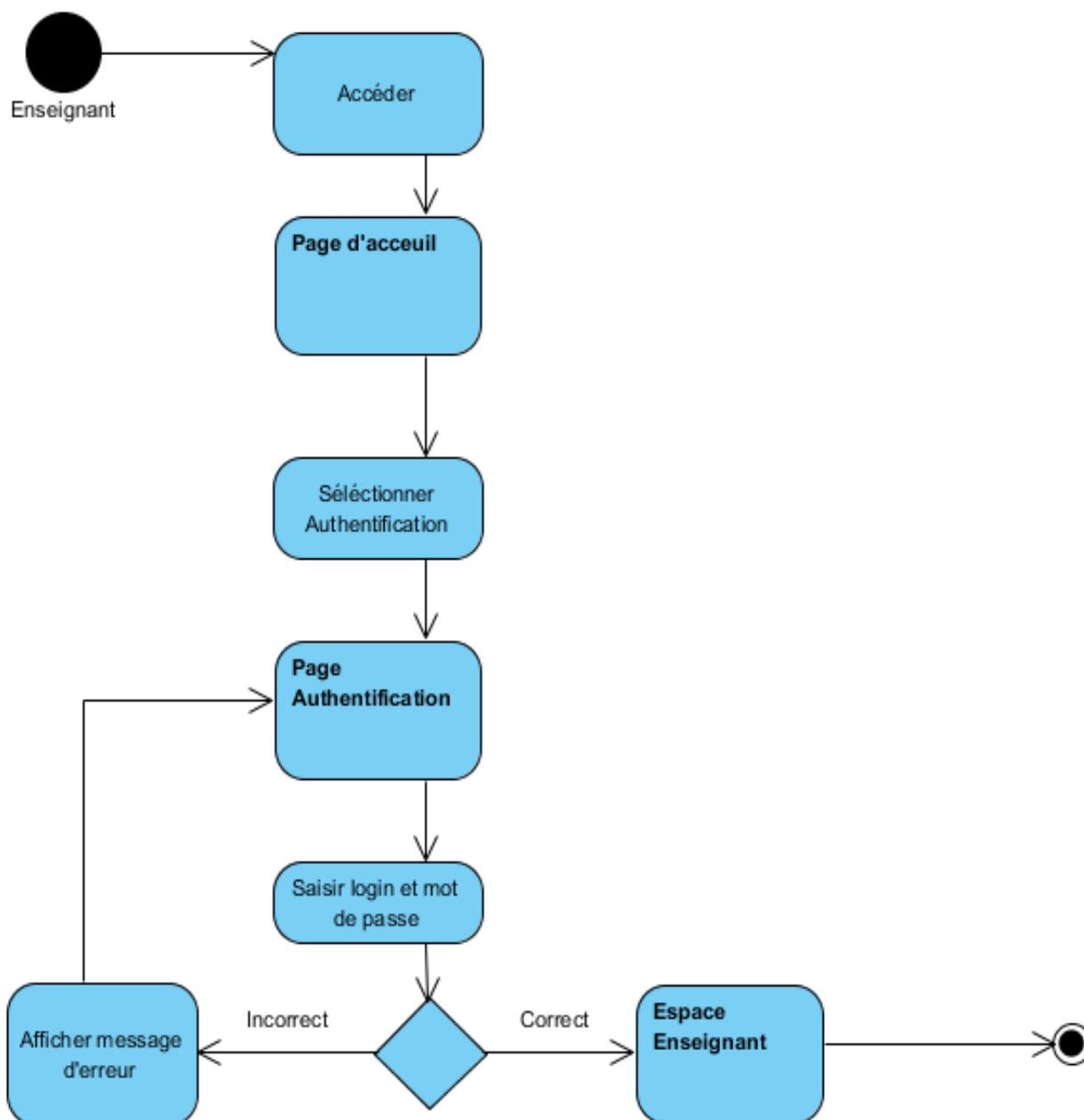


FIGURE 3.9: Diagramme d'activité du cas d'utilisation "S'authentifier "

3.8.2 Diagramme d'activités du cas d'utilisation « Changer le mot de passe »

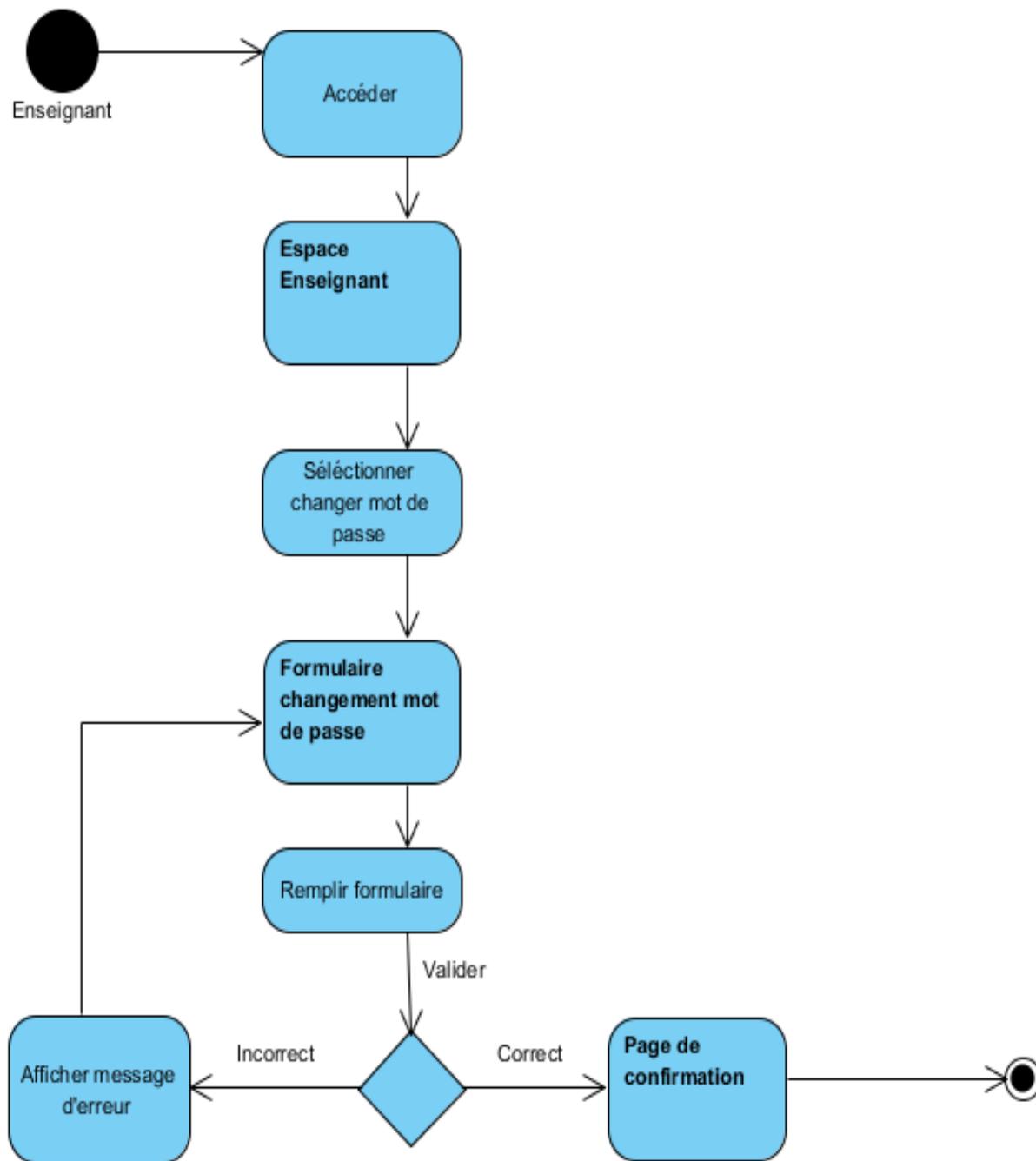


FIGURE 3.10: Diagramme d'activité du cas d'utilisation "changer mot de passe"

3.8.3 Diagramme d'activités du cas d'utilisation « Consultation des notes »

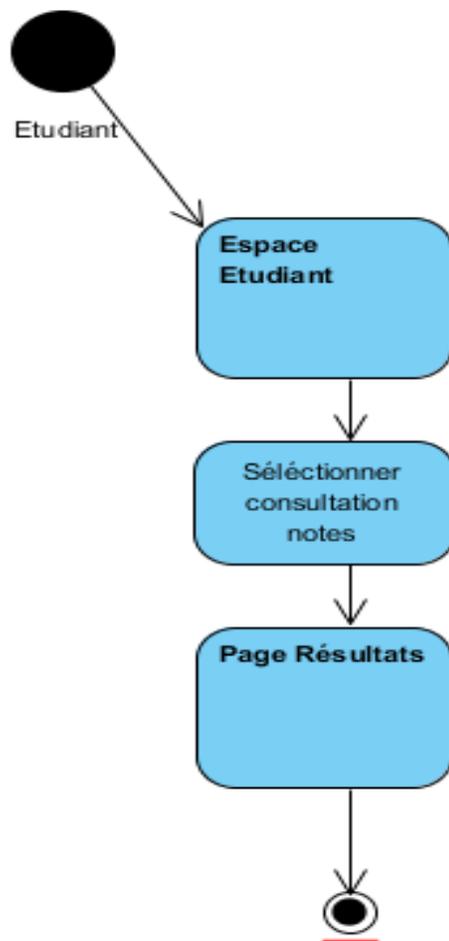


FIGURE 3.11: Diagramme d'activité du cas d'utilisation "Consultation notes"

3.8.4 Diagramme d'activités du cas d'utilisation « Affichage des PV modulaires »

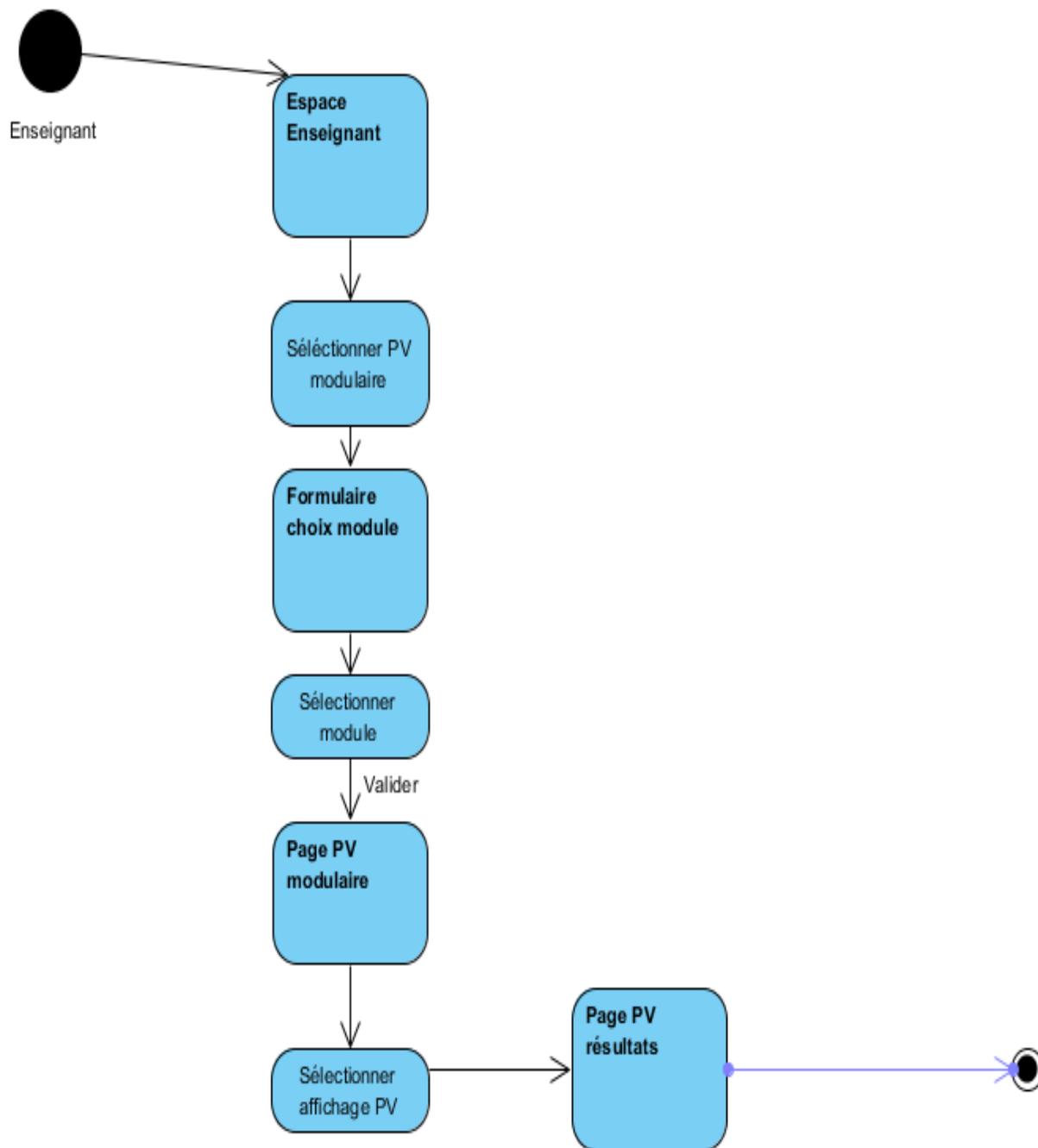


FIGURE 3.12: Diagramme activité du cas d'utilisation «Affichage PV modulaire »

3.8.5 Diagramme d'activités du cas d'utilisation « Création de comptes »

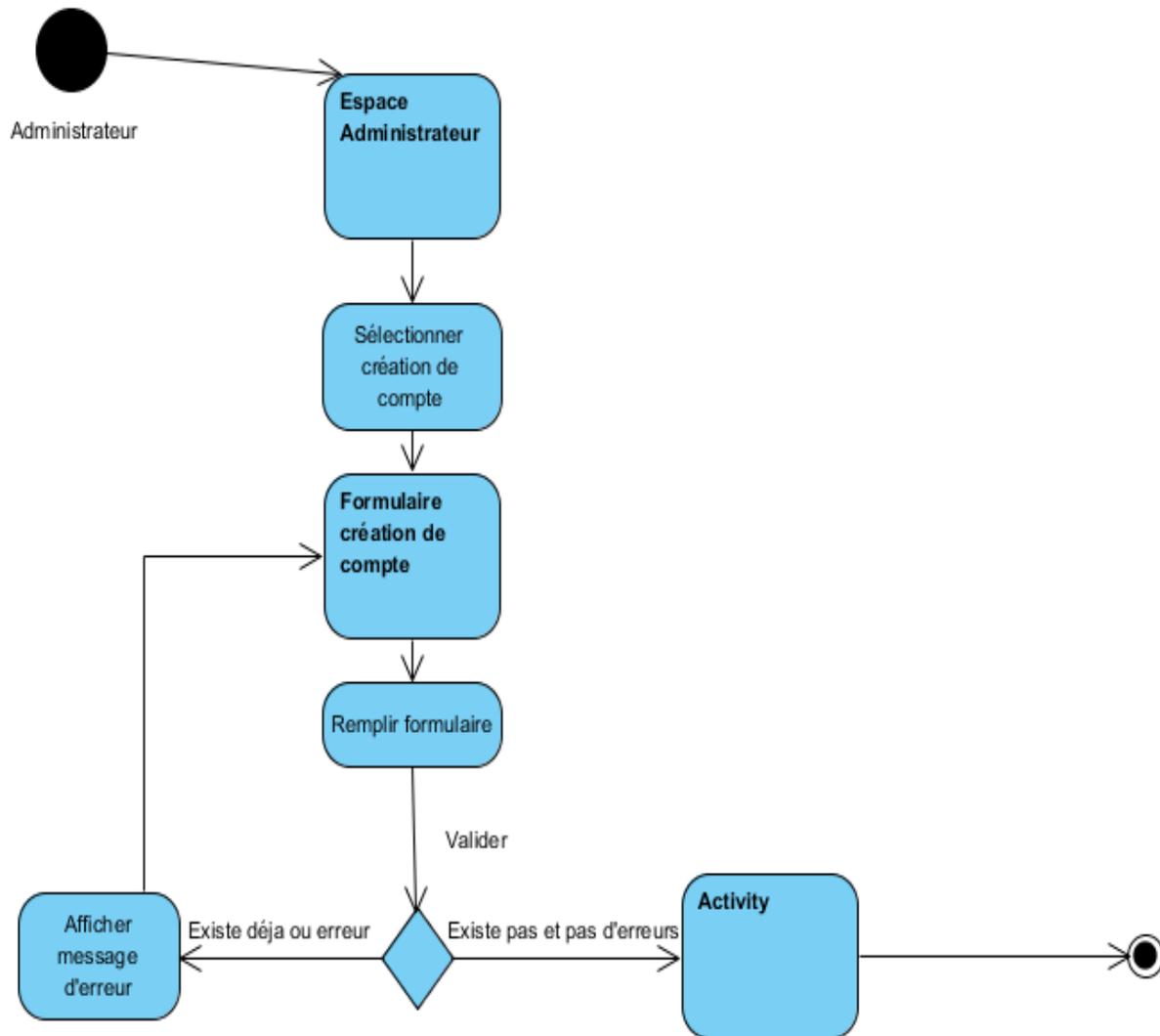


FIGURE 3.13: Diagramme d'activité du cas d'utilisation "Création de compte"

3.9 Conception de la base de données

Après avoir modéliser notre application web avec les différents diagrammes écrits par le langage de modélisation UML, nous avons suivi le déroulement de chaque cas d'utilisation pour en extraire les différentes données nécessaires. Pour l'implémentation de la base de données, on aura besoin d'élaborer un modèle logique de données. Nous présenterons dans ce qui suit, le diagramme des classes et la structure des tables de la base de données.

3.9.1 Le diagramme de classes

la structure de notre base de données est décrite à travers le diagramme de classe suivant :

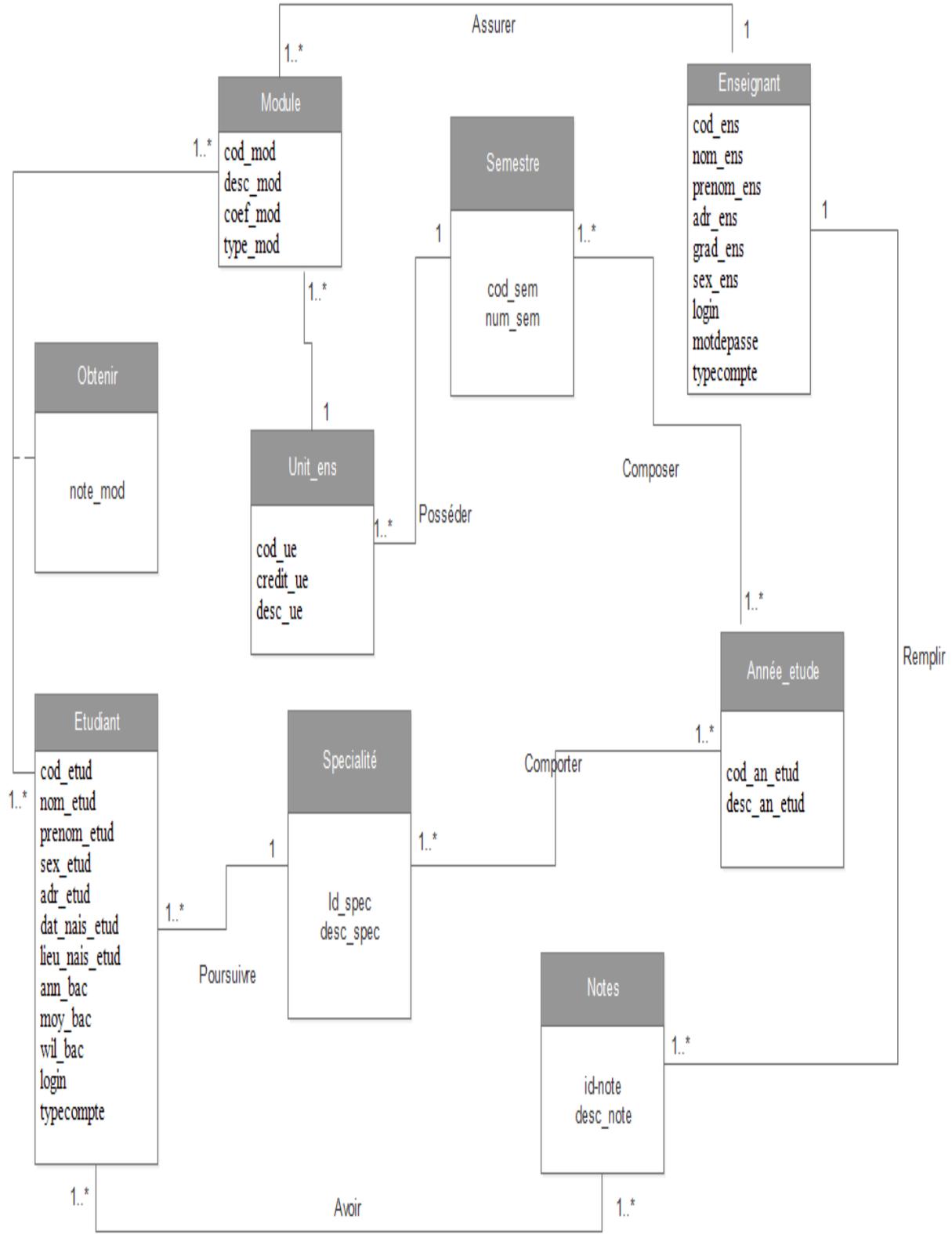


FIGURE 3.14: Le diagramme de classe

3.9.2 Structure des tables de la base de données

3.9.2.1 Table Etudiant

Champ	Signification	Type	Clé
Cod_etud	Code de l'étudiant	INT (11)	PRIMAIRE
Nom_etud	Nom de l'étudiant	VARCHAR (20)	
Prenom_etud	Prénom de l'étudiant	VARCHAR (2)	
Sex_etud	Sexe de l'étudiant	VARCHAR (10)	
Adr_etud	Adresse de l'étudiant	VARCHAR (70)	
Dat_nais_etud	Date de naissance de l'étudiant	VARCHAR (10)	
Lieu_nais_etud	Lieu de naissance de l'étudiant	VARCHAR (25)	
Ann_bac	Année du bac	INT (4)	
Moy_bac	Moyenne du bac	DOUBLE	
Wilaya_bac	Wilaya d'obtention du bac	VARCHAR (10)	
Login	Login de l'étudiant	VARCHAR (10)	
Typecompt	Type de compte	VARCHAR (25)	
Id_spec	Id de la spécialité	VARCHAR (25)	ETRANGERE

3.9.2.2 Table Enseignant

Champ	Signification	Type	Clé
Cod_ens	Code de l'enseignant	INT (11)	PRIMAIRE
Nom_ens	Nom de l'enseignant	VARCHAR (20)	
Prenom_ens	Prénom de l'enseignant	VARCHAR (25)	
Sex_etud	Sexe de l'enseignant	VARCHAR (10)	
Adr_etud	Adresse de l'enseignant	VARCHAR (30)	
Grad_ens	Grade de l'enseignant	VARCHAR (10)	
Login	Login de l'enseignant	VARCHAR (20)	
Motdepass	Mot de passe de l'enseignant	VARCHAR (20)	
Type_compt	Type du compte de l'enseignant	VARCHAR (25)	

3.9.3 Table Module

Champ	Signification	Type	Clé
Cod_mod	Code du module	INT (11)	PRIMAIRE
Desc_mod		VARCHAR (25)	
Coef_mod		INT (2)	
Typ_mod		VARCHAR (25)	
Cod_ue		INT (11)	ETRANGERE

3.9.4 Table Unité d'enseignement

Champ	Signification	Type	Clé
Cod_ue	Code de l'unité d'enseignement	INT (11)	PRIMAIRE
Cred_ue	Crédit total de l'unité	INT (2)	
Desc_ue	Description de l'unité d'enseignement	VARCHAR (25)	

3.9.5 Table Semestre

Champ	Signification	Type	Clé
Cod_sem	Code du semestre	INT (11)	PRIMAIRE
Num_sem	Numéro du semestre	INT (1)	

3.9.6 Table Spécialité

Champ	Signification	Type	Clé
Cod_spec	Code de la spécialité	INT (11)	PRIMAIRE
Desc_spec	Description de la spécialité	VARCHAR (11)	

3.9.7 Table Note

Champ	Signification	Type	Clé
Id_note	Identifiant de note	INT (11)	PRIMAIRE
Desc_note	Description de la note	VARCHAR (25)	
Cod_etud	Code de l'étudiant	INT (11)	ETRANGERE

3.9.8 Table année d'étude

Champ	Signification	Type	Clé
Cod_an_etud	Code de l'année d'étude	INT (11)	PRIMAIRE
Desc_an_etud	Description de l'année d'étude	VARCHAR (25)	

3.9.9 Table Obtenir

Champ	Signification	Type	Clé
Note_mod	Note du module	INT (2)	PRIMAIRE
Cod_etud	Code de l'étudiant	INT (11)	ETRANGERE
Cod_mod	Code du module	INT (11)	ETRANGERE

3.10 Conclusion

Dans ce chapitre, nous avons proposé une démarche de modélisation pour développer notre application, cette démarche est basée sur la méthode UML extension Web. En commençant par la spécification des cas d'utilisation et le cas d'utilisation générale en phase d'analyse, et en phase de conception nous avons dénit les diagrammes de séquences, les diagrammes de classe, le modèle conceptuel de données et enfin les tables de base données.

Deuxième partie

Partie pratique du projet

Chapitre 4

Réalisation

4.1 Introduction

Après avoir finalisé l'étape d'analyse et conception, nous passons à la réalisation de notre application. Nous commençons tout d'abord par la description de notre environnement de travail, ensuite nous présenterons la façon dont le site fonctionne avec ces différentes interfaces.

4.2 Description de l'environnement de travail

Configuration logicielle :

- Système d'exploitation Microsoft Windows 7 64 bits .
- Notepad ++.
- Brackets.
- Paint.NET.
- Uwamp git.
- Eclipse For PHP.
- Python For Windows.
- Cygwin.
- EasyPHP 5.4 (environnement de développement Apache, PHP, MySQL, phpMyAdmin).

Langages de programmation :

- CSS3 : utilisé pour le style et l'apparence du site web.
- HTML 5 : créer le contenu statique du site web.
- PHP : pour le traitement de données.
- SQL : pour l'interrogation de la base de données.
- JavaScript : souvent utilisé pour des vérifications.
- JQuery : souvent utilisé pour les effets et les animations du site.
- Python : Utilisé dans plusieurs domaines : script, applications graphiques et multimédia, réseaux,....

4.3 Environnement logiciel

4.3.1 Notepad ++

Notepad++ est un éditeur de texte générique codé en C++, qui intègre la coloration syntaxique de code source pour les langages et chiers C, C++, Java, C#, XML, HTML, PHP, JavaScript, makele, art ASCII, doxygen, .bat, MS chier ini, ASP, Visual Basic/VBScript, SQL, Objective-C, CSS, Pascal, Perl, Python, R, MATLAB, Lua, TCL, Assembleur, Ruby, Lisp, Scheme, Properties, Di, Smalltalk, PostScript et VHDL ainsi que pour tout autre langage informatique, car ce logiciel propose la possibilité de créer ses propres colorations syntaxiques pour un langage quelconque.

Ce logiciel, basé sur la composante Scintilla, 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 ecace. Il est également une alternative au bloc-notes de Windows (d'où le nom). Le projet est sous licence GPL.

Il ne bloque pas le fichier en cours d'édition et détecte toute modification apportée à celui-ci par un autre programme (il propose de le recharger).

```

24
25 <?PHP
26 //le id du commentaire a qui en repond
27 $id=$_GET['id_re'];
28
29 mysql_connect("localhost","root","");
30 mysql_select_db("projet");?>
31
32 <?php
33
34
35 $reponse = mysql_query("SELECT * FROM reserver where id='".$id.' " );
36 while ($rep = mysql_fetch_array ($reponse))
37 {
38     ?>
39
40
41 <table class="table_consult_ab"width="80%">
42 <tr>
43     <td class="td_consulter_ab"width=35%>Id </td> <td><?php echo $rep['id'];?></td>
44 </tr>
45
46 <tr>
47     <td class="td_consulter_ab" width=35%>Nom </td> <td><?php echo $rep['nom'];?></td>
48 </tr>
49 <tr>
50     <td class="td_consulter_ab" width=35%>Prenom </td> <td><?php echo $rep['prenom'];?></td>
51 </tr>
52 <tr>
53     <td class="td_consulter_ab" width=35%>Véhicule voulu</td> <td><?php echo $rep['typev'];?></td>
54 </tr>
55 <tr>
56     <td class="td_consulter_ab" width=35%>Point de RDV </td> <td><?php echo $rep['ndrv'];?></td>

```

FIGURE 4.1: Interface NotePad++

4.3.2 Brackets

Brackets est un éditeur open source maintenu par Adobe qui a la particularité d'être centré sur l'intégration d'un bout à l'autre.

Partant du principe qu'il est important de comprendre comment est fait un outil pour pouvoir l'adapter à ses besoins, Bracket est entièrement fait en HTML, CSS et JS. Il se lance de plus dans un navigateur, pour nous permettre de travailler et tester dans le même environnement développement/production.

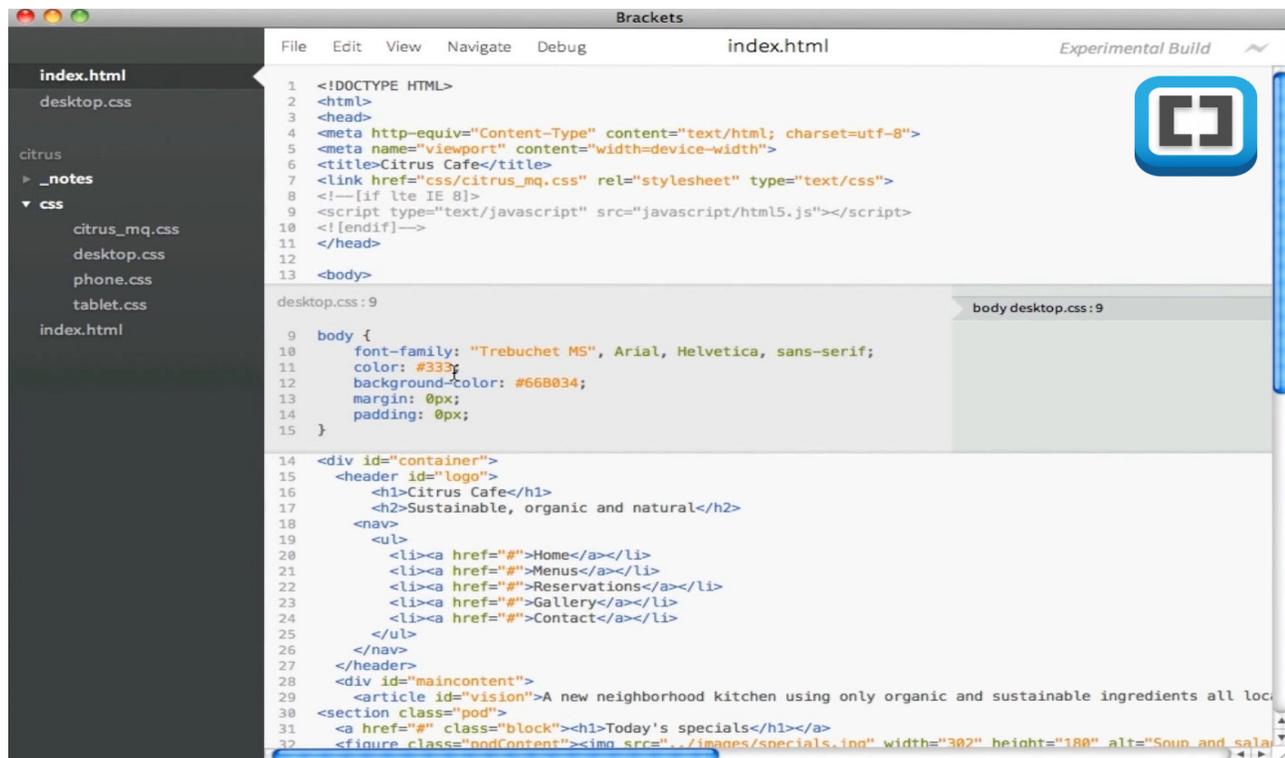


FIGURE 4.2: Interface de Brackets

4.3.3 Paint.NET

Paint.NET est un logiciel de retouche photo freeware.

C'est un programme d'édition graphique gratuit, destiné aux systèmes d'exploitation Windows Vista, 7, 8 et Server 2003.

Le logiciel est écrit en C#. Une bibliothèque de greffons permet d'enrichir les fonctionnalités du logiciel, en proposant notamment des pinceaux personnalisés (sur un canevas séparé), un remplacement de couleurs, la capture d'écran, la gestion du format Bitmap en 32 bits, etc. La version 3.5 nécessite l'environnement .NET 3.5 SP1.

Le logiciel Paint.NET nécessite l'environnement .NET de Microsoft, et prend en charge nativement le format .PDN, qui préserve les différentes couches de dessin ainsi que d'autres informations.

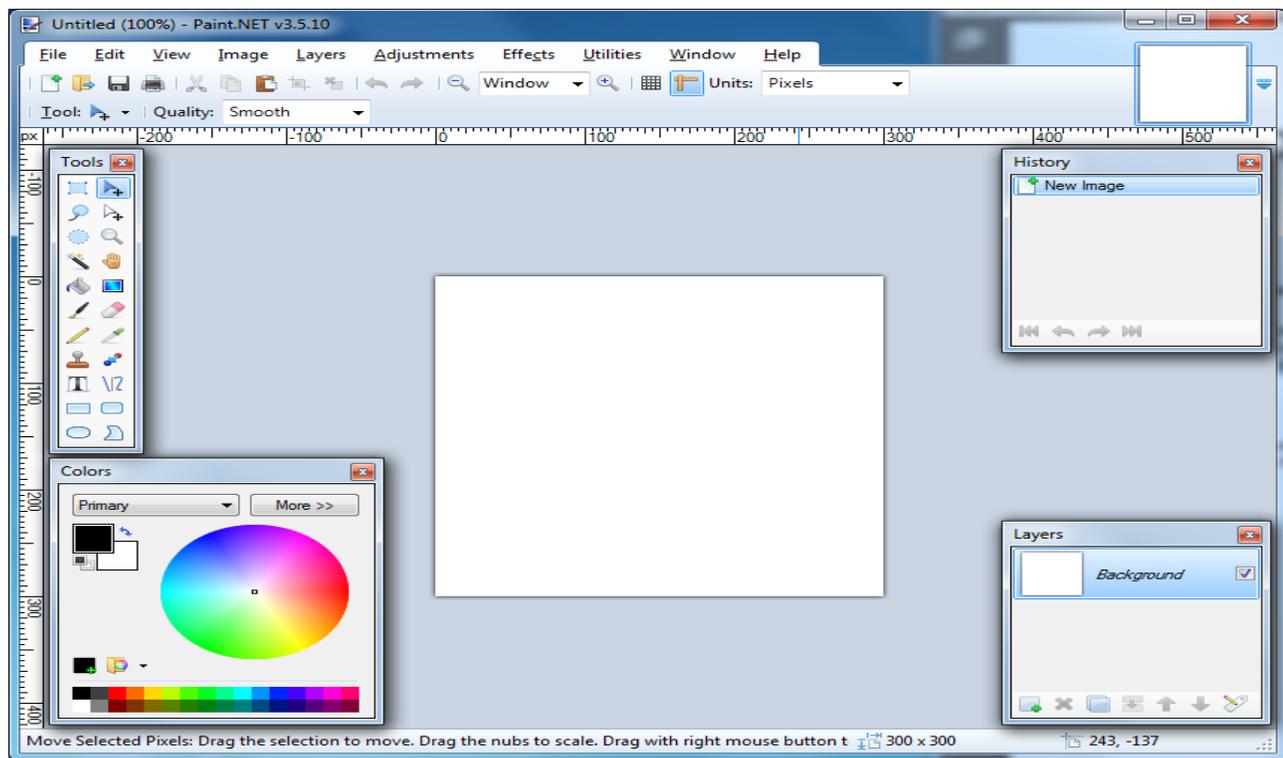


FIGURE 4.3: Interface Paint.NET

4.3.4 Uwamp

UwAmp est un package de type WAMP regroupant Apache, PHP, MySQL, SQLite et php-MyAdmin qui simplifie la création d'un serveur Web sur Windows afin de développer des sites en PHP. Ce qui caractérise ce package WAMP c'est son interface de gestion permettant de configurer Apache et PHP sans avoir à modifier les fichiers de configuration à la main.

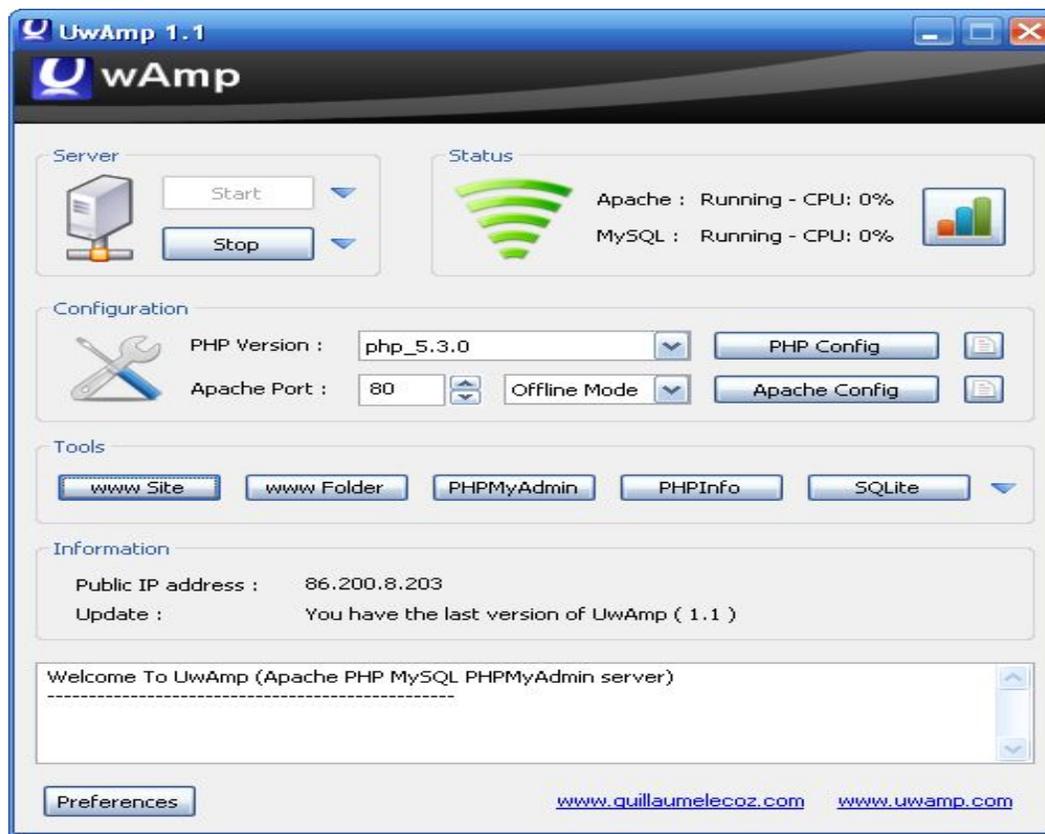


FIGURE 4.4: Interface Uwamp

4.3.5 Eclipse For PHP

Eclipse est un EDI (Environnement de Développement Intégré), ou en anglais un IDE (Integrated Development Environment).

Développé par IBM à partir de ses ancêtres Visual Age et Visual Age For Java, il a depuis été rendu open-source et son évolution est maintenant gérée par la Fondation Eclipse.

La license d'Eclipse (Eclipse Public License et sa FAQ) permet de fournir des plug-ins open sources comme des plug-ins closed-source, des plug-ins gratuits ou encore payants.

C'est pourquoi il est maintenant bien plus qu'un IDE Java, et gère un grand nombre de langages de programmation tels que PHP, C, C++, etc.

Eclipse est également devenu une plate-forme, servant de socle à d'autres applications, que celles-ci soient destinées au développement logiciel ou à tout autre domaine.

Eclipse For PHP ce n'est que Eclipse installé avec le plugin PHP Eclipse.

4.3.6 Python For windows

Python est simplement un langage de programmation souple et puissant , en plus d'être facile à apprendre. Voici quelques caractéristiques :

1. **Légèreté et clarté du code :** En moyenne 5 fois moins de lignes qu'en C++ et une syntaxe limpide. D'ailleurs voici le "Hello World" : `print "Hello World"`

2. **Puissance et flexibilité** : Faisant le lien avec de nombreuses bibliothèques C, python trouve son utilité dans des domaines variés : script, applications graphiques et multimédia, réseaux, bases de données, parseurs divers... Et tout cela en POO.
3. **Portabilité et liberté** : Python est sous licence libre, et de plus gratuitement utilisable pour toute utilisation, professionnelle ou non. En tant que langage interprété, il peut se révéler parfois lent mais compense ses faiblesses par de nombreuses astuces (générer du bytecode Java par exemple) qui diminuent fortement ce handicap. Et l'avantage est bien sur que tout système disposant d'un interpréteur peut accueillir Python, il est donc disponible un peu partout : Windows, Mac, Linux, BSD, Solaris, certains PDA et même Amiga.

Installation de Python sous Windows :

1. Cliquez sur le lien Download dans le menu principal de la page.
2. Sélectionnez la version de Python que vous souhaitez utiliser (je vous conseille la dernière en date).
3. Sélectionnez celle qui conviendra à votre processeur. Si vous avez un doute, téléchargez une version x86 .
4. Enregistrez puis exécutez le fichier d'installation et suivez les étapes.
5. Une fois l'installation terminée, vous pouvez vous rendre dans le menu Démarrer > Tous les programmes et l'exécuter .

4.3.7 EasyPHP

EasyPHP fut le premier package WAMP à voir le jour en 1999. Il s'agit d'une plateforme de développement Web simple et gratuite, qui a pour ambition de simplifier l'approche du monde de développement de site Web dynamique permettant de faire fonctionner localement sans se connecter à un serveur externe des scripts PHP. EasyPHP regroupe deux serveur (un serveur Web Apache et un serveur de base de données MySQL), un interpréteur de script PHP, ainsi qu'une administration SQL PhpMyAdmin. Il dispose d'une interface d'administration permettant de gérer les alias (pour qu'une page PHP soit interprétée il faut placer les fichiers dans le répertoire www, ou dans un alias).

4.3.7.1 Installer EasyPHP

L'installation de EasyPHP est très simple, il sut juste de :

1. Télécharger une version de EasyPHP sur le site <http://www.easyphp.org>.
2. Double clique sur l'exécutable télécharger.
3. Sélectionner le répertoire d'installation et suivre la procédure.

4.3.7.2 Démarrer EasyPHP

Pour démarrer Apache, MySQL et PHP, il vous sut de lancer EasyPHP à partir du groupe créé dans le menu démarrer. Une fois que EasyPHP est lancé une icône apparait à l'extrémité de la barre des tâches, et avec un clique droit, différents menus apparaissent :

1. Fichier.log : renvoi aux erreurs générées par Apache et MySQL.
2. Configuration : accéder aux différentes congurations de EasyPHP.

3. Administration : accéder à la page d'administration des alias ou bien à l'outil PhpMyAdmin.
4. Web local : ouvrir le Web local.
5. Démarrer/Arrêter : démarrer ou arrêter Apache et MySQL.
6. Redémarrer : redémarrer Apache et MySQL.
7. Quitter : fermer EasyPHP.

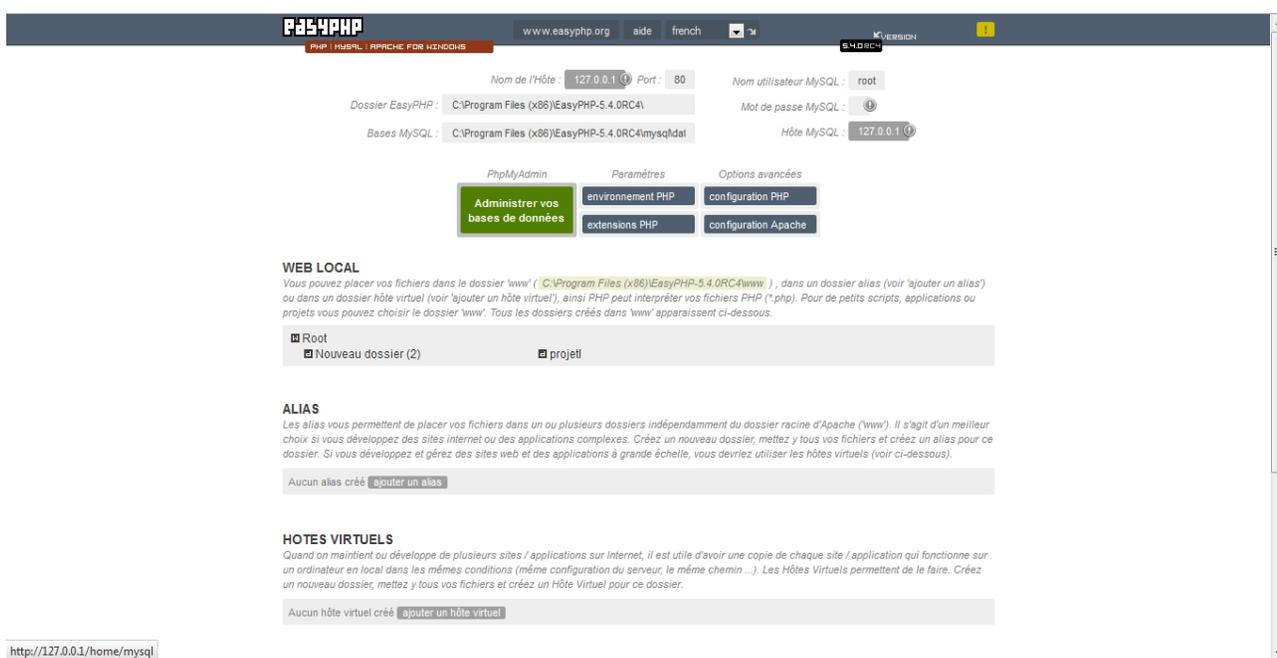


FIGURE 4.5: Interface de EasyPHP

4.4 Environnement de programmation

4.4.1 Le serveur Web Apache

C'est le serveur le plus utilisé sur le Web. A l'origine, fonctionnant sur Unix il a été porté sur de nombreux systèmes dont Microsoft Windows. Apache présente de nombreux avantages :

- Gratuit ;
- Robuste ;
- Un niveau de performances élevé pour des exigences matérielles modestes.

4.4.2 Le serveur de base de données MySQL (My Structured Query language)

C'est un système de gestion de base de données Open Source très répandu sur le Web. Il fait partie d'un grand nombre de produit intégré de type LAMP (Linux Apache MySQL PHP) ou WAMP (Windows Apache MySQL PHP) et en particulier EasyPHP. Ses points forts sont :

- Gratuit.
- Rapide et robuste.
- Multiutilisateurs.
- Simple d'utilisation, manuel d'utilisation en ligne.
- Fonctionne sur de nombreuses plateformes : Windows, Linux et bien d'autres.

4.4.3 L'outil phpMyAdmin

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, qu'ils soient gratuits ou payants, le proposent ce qui permet à l'utilisateur de ne pas avoir à l'installer. Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances dans le domaine des bases de données, de nombreuses requêtes comme les créations de table de données, les insertions, les mises à jour, les suppressions, les modifications de structure de la base de données. Ce système est très pratique pour sauvegarder une base de données et ainsi transférer facilement ses données. De plus celui-ci accepte la formulation de requêtes SQL directement en langage SQL, cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.

The screenshot shows the phpMyAdmin interface for the 'wordpress' database, specifically the 'wp_comments' table. The table structure is displayed with columns and their types, including 'comment_ID' (bigint(20)), 'comment_post_ID' (int(11)), 'comment_author' (tinytext), 'comment_author_email' (varchar(100)), 'comment_author_url' (varchar(200)), 'comment_author_IP' (varchar(100)), 'comment_date' (datetime), 'comment_date_gmt' (datetime), 'comment_content' (text), 'comment_karma' (int(11)), 'comment_approved' (varchar(20)), 'comment_agent' (varchar(255)), 'comment_type' (varchar(20)), 'comment_parent' (bigint(20)), and 'user_id' (bigint(20)).

Below the table structure, there is a section for 'Index' and 'Statistiques'. The 'Index' section shows the primary key 'comment_ID' and other indexes like 'comment_approved', 'comment_post_ID', and 'comment_date_gmt'. The 'Statistiques' section provides information about the table's size, format, and creation/modification dates.

FIGURE 4.6: Interface PhpMyAdmin

4.5 Les langages de programmation utilisés

4.5.1 HTML5 (HyperText Markup Language)

C'est un langage dit de marquage (ou de balisage) dont le rôle est de formaliser l'écriture d'un document avec des balises qui permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents.

Le langage HTML permet notamment la lecture de documents sur Internet à partir de machines, grâce au protocole HTTP, permettant d'accéder via le réseau à des documents repérés par une adresse unique, appelée URL.

Syntaxe :

```
<html>
```

```
<head>
<title> Nom du document </title>
</head>
<body>
</body>
</html>
```

4.5.2 CSS 3 (Cascade Style Sheet)

CSS (Cascading Style Sheets : feuilles de style en cascade) est un langage informatique qui 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 pris en charge par les navigateurs web dans les années 2000.

Le développement du troisième niveau des feuilles de styles en cascade commence dès 1999, parallèlement à celui de CSS 2.1.

CSS3 devient modulaire, afin de faciliter ses mises à jour, mais aussi son implémentation par des agents utilisateurs aux capacités et aux besoins de plus en plus variés (navigateurs graphiques, navigateurs pour mobiles, navigateurs vocaux). Les navigateurs peuvent ainsi implémenter des sous-ensembles de CSS.

4.5.3 PHP (HyperText PreProcessor)

PHP est un langage de script libre apparu en 1994 grâce à Rasmus Lerdorf. Sa syntaxe provient du langage C, Perl et Java.

Associé notamment au serveur web Apache et à la base de données MySQL, c'est le langage le plus populaire pour les pages Web. Il s'exécute côté serveur permettant de générer des pages HTML dynamiques, c'est-à-dire modifiables en fonction des interactions de l'utilisateur et des éventuelles données historiques associées à l'utilisateur et à la session courante.

PHP dispose des principaux atouts suivant :

- La gratuité et la disponibilité du code source.
- La simplicité d'écriture des scripts.
- La possibilité d'inclure le script PHP au sein d'une page HTML
- L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, etc.).

4.5.4 SQL (Structured Query Language ou Langage de requête structuré)

Langage d'interrogation et de manipulation de bases de données relationnelles développé par IBM dans les années 70, et normalisé en 1986 par l'ANSI (American National Standard Institute).

Il est proposé par une multitude de systèmes de gestion de bases de données comme MySQL, Sybase ou Access.

Ce langage concerne plusieurs types d'opérations sur les bases de données relationnelles, répartis en trois (3) familles :

- **LDD** : Langage de **D**éfinition de **D**onnées : Création, modification et suppression de données.
- **LMD** : Langage de **M**anipulation de **D**onnées : Sélectionner, insérer, modifier, supprimer des données dans une table.
- **LCD** : Langage de **C**ontrôle de **D**onnées : Définir les permissions des utilisateurs.

4.5.5 JQuery

jQuery est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006.

La bibliothèque contient notamment les fonctionnalités suivantes :

- Parcours et modification du DOM (y compris le support des sélecteurs CSS 1 à 3 et un support basique de XPath) ;
- Événements ;
- Effets et animations ;
- Manipulations des feuilles de style en cascade (ajout/suppression des classes, d'attributs. . .)
- ; Ajax ;
- Plugins ;
- Utilitaires (version du navigateur. . .).

4.5.6 Python

Le langage Python est un langage de programmation complet grâce aux nombreuses bibliothèques spécialisées, appelées modules. Utilisé professionnellement dans la recherche biomédicale et pour la gestion de Google, il est néanmoins idéal pour un apprentissage de la programmation, étant spatialement structuré. Orienté objet, on peut l'utiliser comme un bon vieux BASIC, moyennant quelques restrictions par rapport à ce dernier :

- les instructions graphiques sont l'objet de bibliothèques particulières, comme **tkinter**
- certaines commandes basiques se trouvent dans des bibliothèques, appelées **modules** : «quitter» dans **sys**, «attendre» dans **time**, la trigonométrie dans **math**, le pseudo-aléatoire dans **random...**
- on ne dispose d'aucun **GOTO** .

4.5.7 JavaScript

Créé en 1995 par Brendan Eich, JavaScript est un langage de script incorporé dans des pages HTML permettant d'apporter des améliorations en exécutant des commandes du côté client, c'est-à-dire au niveau du navigateur et non du côté serveur web.

Il est souvent utilisé dans les processus de vérifications des données de formulaire.

4.6 Présentation de quelques interfaces

4.6.1 La page d'accueil

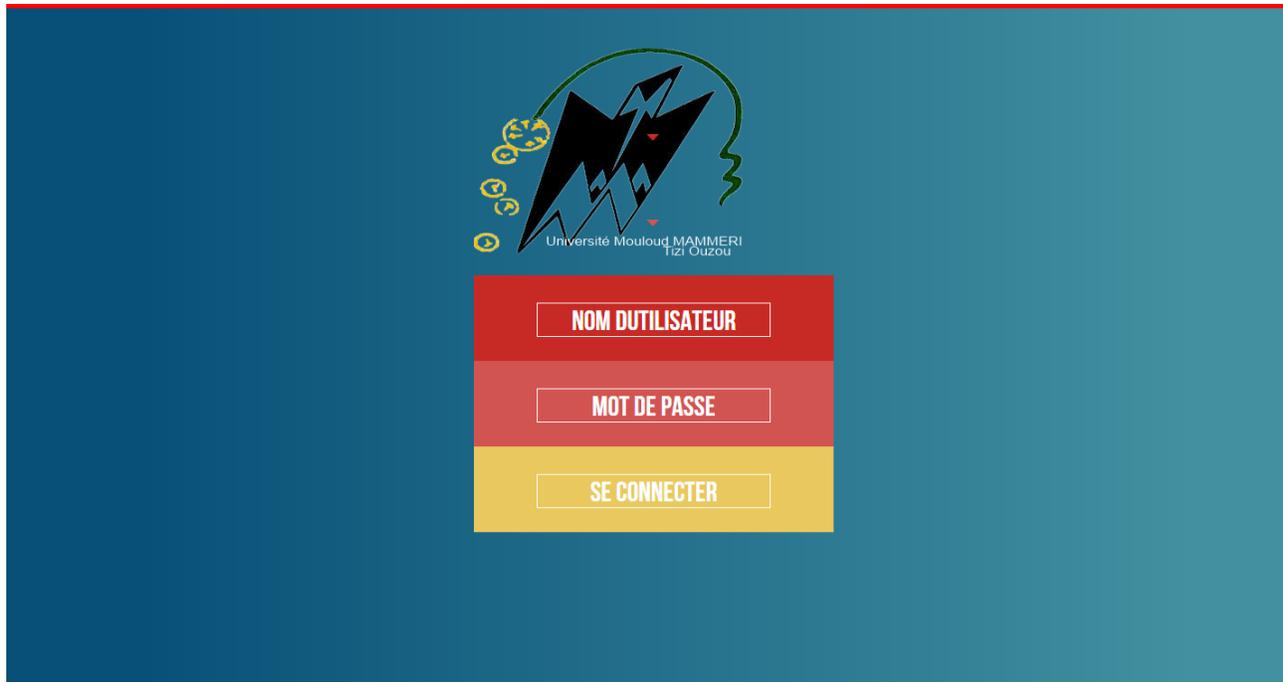


FIGURE 4.7: Page d'accueil

4.6.2 Page “Ajouter utilisateur”

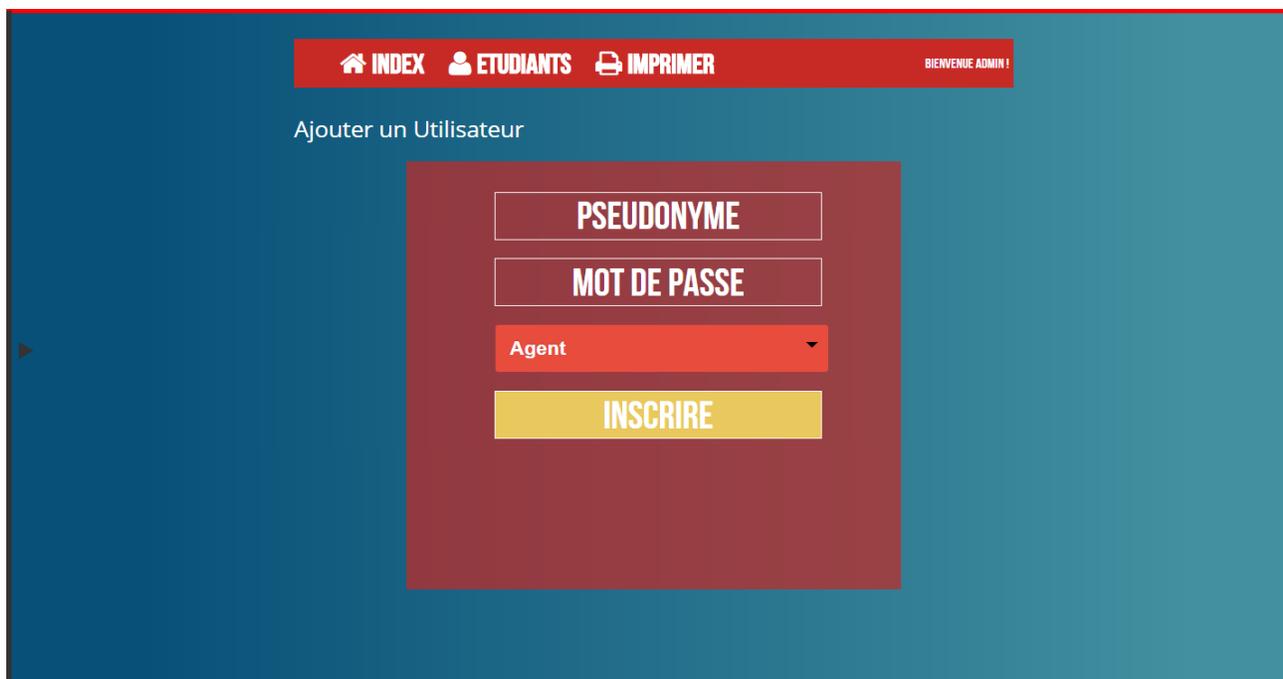


FIGURE 4.8: Page “Ajouter utilisateur”

4.6.3 Page d'administrateur

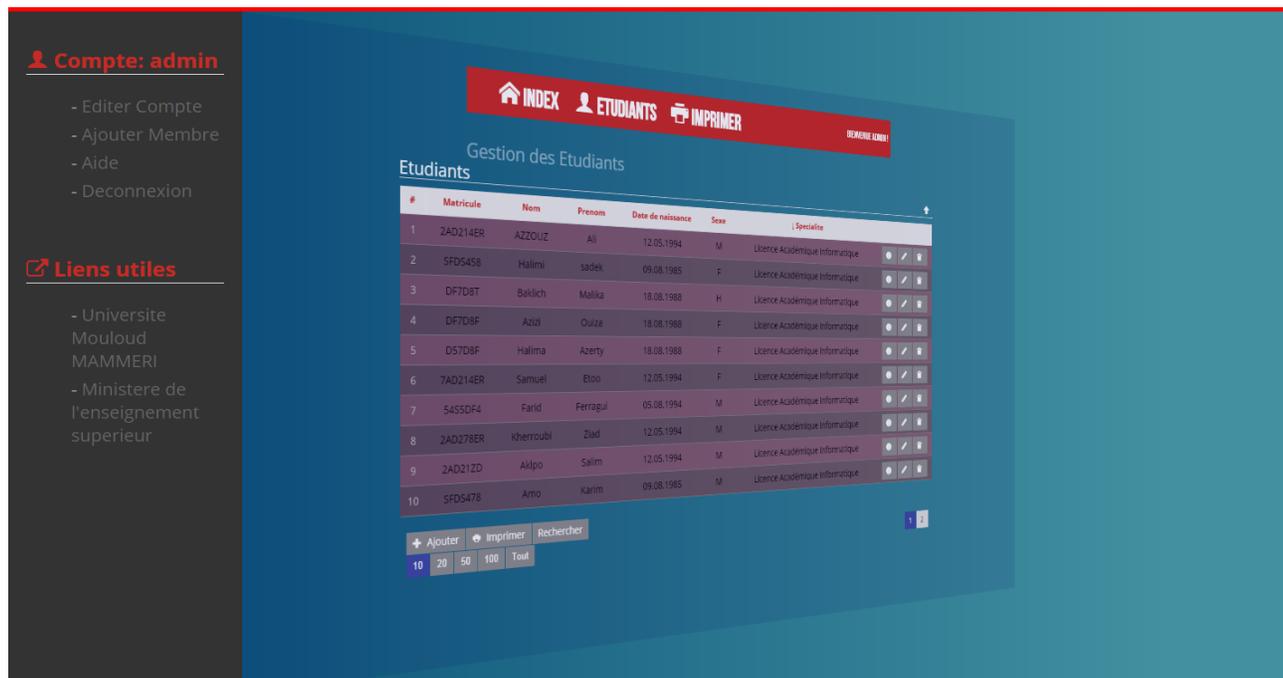


FIGURE 4.9: Page administrateur

4.6.4 Page de l'agent de scolarité

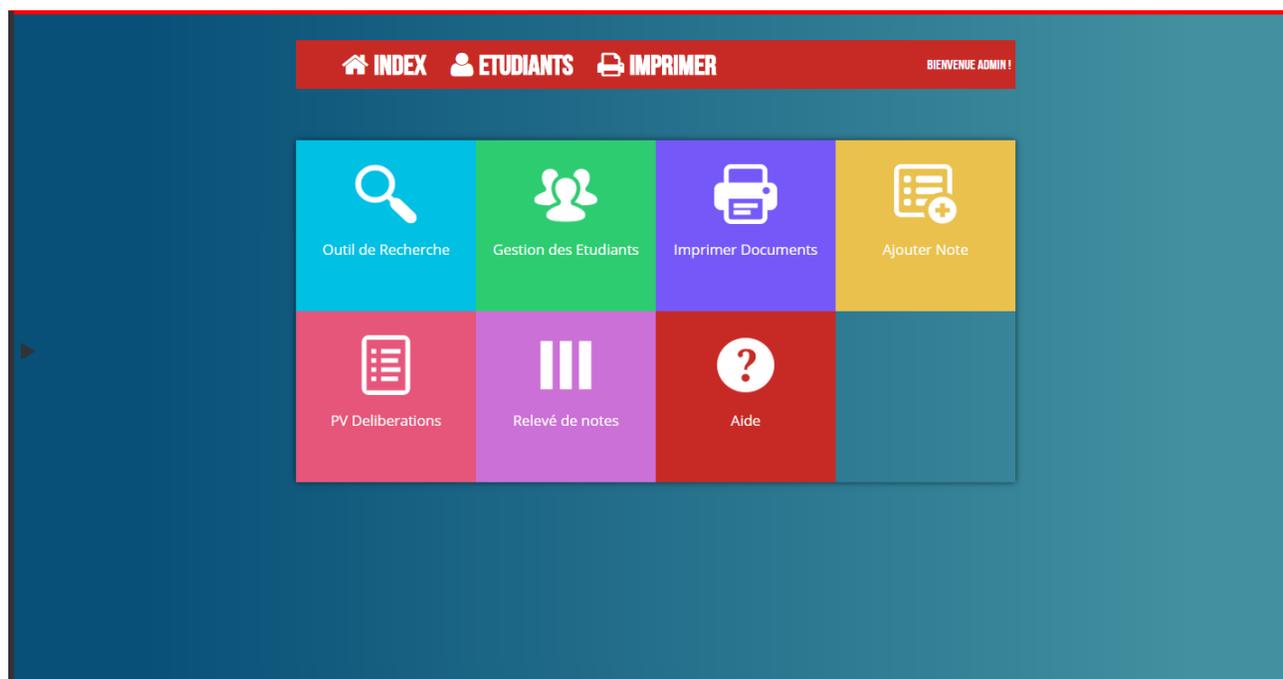


FIGURE 4.10: Page de l'agent de scolarité

4.6.5 Page “Imprimer”

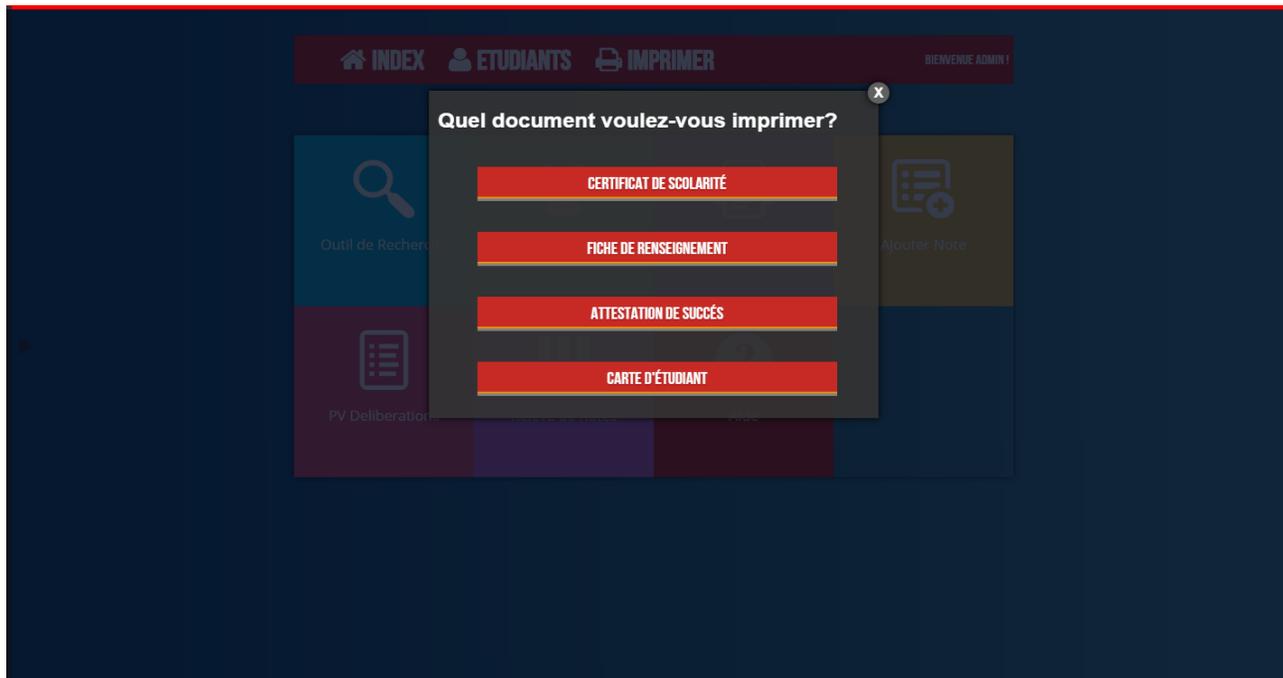


FIGURE 4.11: Page “Imprimer”

4.6.6 Page “Ajouter note”

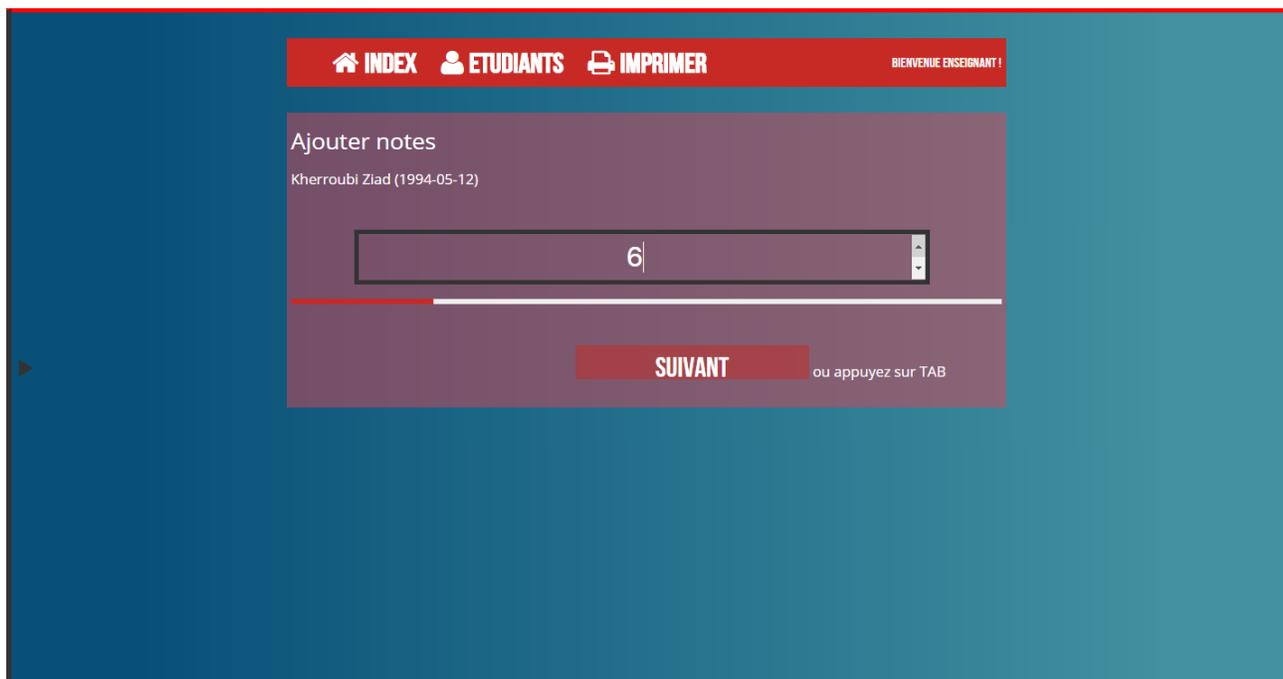


FIGURE 4.12: Page “Noter module”

4.7 Conclusion

Dans ce dernier chapitre, nous avons présenté d'une part les différents outils et langages utilisés pour réaliser notre site Web et d'autre part nous avons présenté quelques interfaces afin d'illustrer des fonctionnalités du site.

Conclusion générale

L'objectif de ce travail est de réaliser une application web pour la gestion de la scolarité LMD. Pour la réalisation de cette dernière nous avons jugé utile de décomposer notre travail en deux parties :

- La première partie consistait à présenter une étude bibliographique composée de deux chapitres :
 1. Le premier présente les notions de bases relatives à la technologie des réseaux, l'Internet, le Web, le client/serveur.
 2. Le deuxième est consacré à la présentation du système LMD.
- La deuxième partie est la base de notre travail, elle est composée de deux chapitres :
 1. Le premier est destiné à la conception de l'application Web.
 2. Le deuxième présente la réalisation de l'application.

Les objectifs assignés dans le cadre de ce projet ont été globalement atteints. La réalisation de ce travail nous a permis d'acquérir de nouvelles connaissances sur les langages de programmation comme Python, HTML5 et CSS3 pour la création des interfaces, Le langage JavaScript et JQuery pour les contrôles, Le langage de modélisation unifié UML, et les bases de données en utilisant (Apache, MySQL et PHP).

La réalisation de cette application nous a permis aussi d'approfondir nos connaissances dans les technologies web.

Troisième partie

Annexe

A. Modèle conceptuel d'UML

Introduction :

La notation UML est une fusion des notations d'OOD, OMT et OOSE. Les concepteurs de cette notation ont recherché avant tout la simplicité. Les symboles embrouillés, redondants ou superflus ont été éliminés en faveur d'un meilleur rendu visuel.

UML est un langage de modélisation fondé sur les concepts objet standard conçu pour l'écriture de plans d'élaboration de logiciels. L'objectif d'UML est de fournir une notation standard utilisable dans le développement de systèmes informatiques basés sur l'objet.

UML est adapté à la modélisation de systèmes, depuis les systèmes informatiques d'entreprise jusqu'aux applications distribuées basées sur le Web, c'est un langage très expressif qui couvre toutes les perspectives nécessaires au développement et au déploiement de tels systèmes. Pour apprendre à s'en servir efficacement, il faut d'abord s'appuyer sur une représentation conceptuelle de ce langage, ce qui nécessite l'assimilation de trois éléments fondamentaux qui sont :

Les briques de base d'UML, (Des éléments. Des relations. Des diagrammes) et les règles qui déterminent la manière de les assembler et quelques mécanismes généraux qui s'appliquent à ce langage.

UML s'articule autour de neuf diagrammes différents, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Par ailleurs, UML modélise le système suivant deux modes de représentation : l'un concerne la structure du système pris "au repos" modèle statiques (diagramme de cas d'utilisation, de classe...etc.) l'autre concerne sa dynamique de fonctionnement (diagramme de séquence, d'états transitions, d'activités...etc.).

Les deux représentations sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composantes fonctionnent entre elles.

Le modèle conceptuel d'UML comprend les notions de bases génériques du langage. Il définit trois sortes de briques de base :

- Les éléments, qui sont les abstractions essentielles à un modèle.
- Les relations, qui constituent des liens entre ces éléments.
- Les diagrammes, qui regroupent des éléments et des liens au sein de divers ensembles.

1. Quelques définitions

1.1 Les interactions :

Une interaction est un comportement qui comprend un ensemble de messages échangés au sein d'un groupe d'éléments, dans un contexte particulier, pour atteindre un but bien défini. Le comportement d'un ensemble d'objets ou celui d'une opération individuelle peut être précisé par une interaction. Comme le montre la figure suivante un message est représenté par une ligne fléchée, qui indique le nom de son opération.

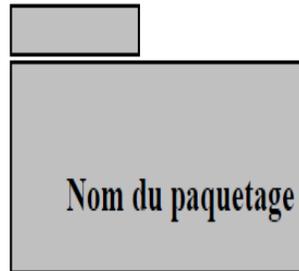


1.2 Les éléments de regroupement :

Les éléments de regroupements représentent les parties organisationnelles des modèles UML. Ce sont des boîtes dans lesquelles un modèle peut être décomposé.

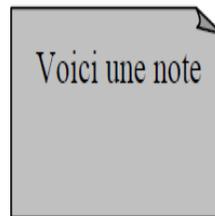
Il existe un seul type fondamental d'éléments de regroupement : le « **paquetage** ».

Un **paquetage** est un ensemble d'éléments de modélisation : des classes, des associations, des objets, des composants. ...



1.3 Les éléments d'annotation :

Les éléments d'annotation représentent les parties explicatives des modèles UML. Ce sont les commentaires qui peuvent accompagner tout élément dans un modèle, à des fins de description, d'exploitation et de remarque. Il existe un type fondamental d'éléments d'annotation appelé « note » qui est simplement un symbole utilisé pour représenter les contraintes et les commentaires rattachés à un élément ou un ensemble d'éléments. Comme le montre la figure suivante une note est représentée par un rectangle écorné qui contient un commentaire textuel ou graphique.



Dans l'étude conceptuelle de notre plate-forme, nous n'avons utilisé que quatre diagrammes, à savoir le diagramme de cas d'utilisation, le diagramme de classe, le diagramme de séquence et le diagramme d'activités.

Nous allons décrire ces diagrammes ainsi que les éléments et les relations les constituant dans ce qui suit.

2. Les Diagrammes d'UML

Un diagramme est la représentation graphique d'un ensemble d'éléments qui constituent un système. Il se présente sous la forme d'un graphe connexe où les sommets correspondent aux éléments et les arcs aux relations. Les diagrammes servent à visualiser un système sous différents perspectives, ce sont des " vues " qui permettent d'isoler certaines parties des modèles pour les rendre plus aisément compréhensibles.

2.1 Description des cas d'utilisation [11] :

Les cas d'utilisations sont des outils formels qui permettent de consigner et d'exprimer des interactions entre les utilisateurs et le système.

On peut noter que les cas d'utilisations sont utilisés durant tout le processus car ils servent à la création de l'IHM, à la spécification des tests

A. L'acteur : Un acteur est un type stéréotypé représentant une abstraction qui réside juste en dehors du système à modéliser. Un acteur représente un rôle joué par une

personne ou une chose qui interagit avec le système. Un acteur n'est pas nécessairement une personne physique : il peut être un service, une société, un système informatique. Il existe 4 catégories d'acteurs :

- les acteurs principaux : les personnes qui utilisent les fonctions principales du système
- les acteurs secondaires : les personnes qui effectuent des tâches administratives ou de maintenance.
- le matériel externe : les dispositifs matériels incontournables qui font partie du domaine de l'application et qui doivent être utilisés.
- les autres systèmes : les systèmes avec lesquels le système doit interagir.

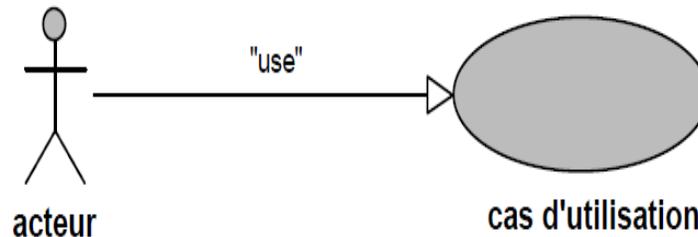
B. Le cas d'utilisation : Le cas d'utilisation (ou use case) correspond à un objectif du système, motivé par un besoin d'un ou plusieurs acteurs. L'ensemble des use cases décrit les objectifs (le but) du système.

C. La relation : Elle exprime l'interaction existant entre un acteur et un cas d'utilisation.

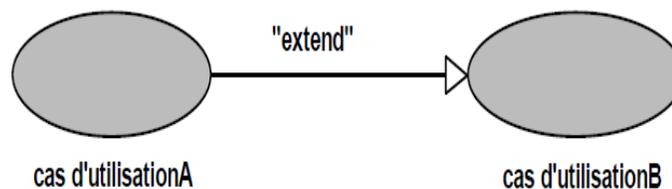
Il existe 3 types de relations entre cas d'utilisation :

- - la relation d'utilisation
- - la relation d'extension
- - la relation d'inclusion

C.1. La relation généralisation : Dans une relation de généralisation entre 2 cas d'utilisation, le cas d'utilisation enfant est une spécialisation du cas d'utilisation parent.

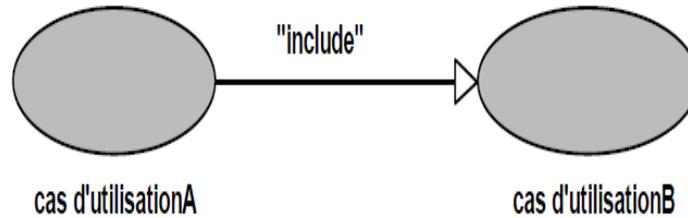


C.2. La relation d'extension : Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à condition. Le comportement ajouté est inséré au niveau d'un point d'extension défini dans le cas d'utilisation destination. Cette relation permet de modéliser les variantes de comportement d'un cas d'utilisation (selon les interactions des acteurs et l'environnement du système).



C.3. La relation d'inclusion : Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. L'inclusion a un caractère obligatoire, la

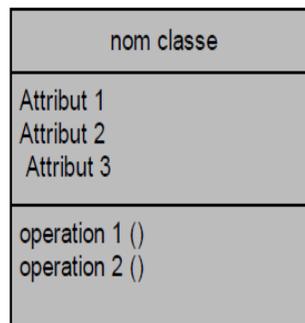
source spécifiant à quel endroit le cas d'utilisation cible doit être inclus. Cette relation permet ainsi de décomposer des comportements et de définir des comportements partageables entre plusieurs cas d'utilisation.



2.2 Le diagramme de classes :

Le diagramme de classes est un diagramme structurel qui exprime d'une manière générale la structure statique d'un système en termes de classes, d'interfaces et de collaborations, ainsi que leurs relations.

2.2.1 Classe : C'est une description d'un ensemble d'objets qui partage les mêmes attributs, opérations, méthodes, relations et contraintes. Une instance d'une classe est appelée objet. Elle est représentée par un rectangle, qui, en règle générale, contient son nom, ses attributs et ses opérations.



2.2.2 les relations : Il existe plusieurs types de relations entre classes :

- l'association
- la généralisation/spécialisation
- la dépendance

a. L'association : Une association est une relation structurelle bidirectionnelle qui décrit un ensemble de liens entre différents éléments (la nature des liens est ajoutée comme décoration).

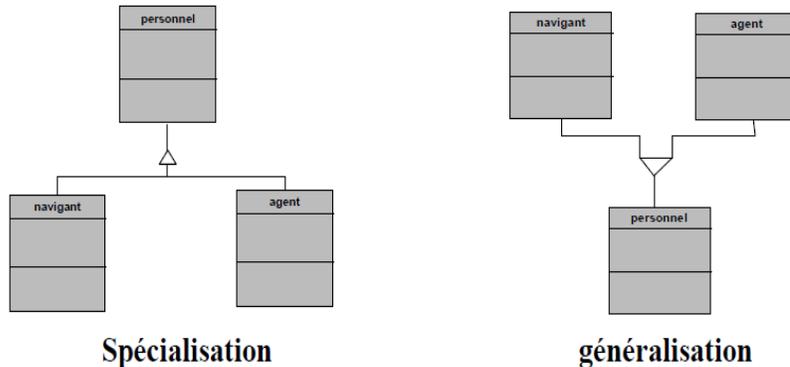
Elle est représentée par une ligne qui peut être fléchée ; elle comprend parfois une étiquette et souvent d'autres décorations, comme la multiplicité et les noms de rôles.

b. La généralisation / spécialisation : Le principe de généralisation / spécialisation permet d'identifier parmi les objets d'une classe (générique) des sous-ensembles d'objets (des classes spécialisées) ayant des définitions spécifiques. La classe plus spécifique est cohérente avec la classe

plus générale c'est-à-dire qu'elle contient par héritage tous les attributs, les membres, les relations de la classe générale, et peut contenir d'autres.

Les relations de généralisation peuvent être découvertes de 2 manières :

- **La généralisation** : il s'agit de prendre des classes existantes (déjà mises en évidence) et de créer de nouvelles classes qui regroupent leurs parties communes ; il faut aller du plus spécifique au plus général.
- **La spécialisation** : il s'agit de sélectionner des classes existantes (déjà identifiées) et d'en dériver des nouvelles classes plus spécialisées, en spécifiant simplement les différences.

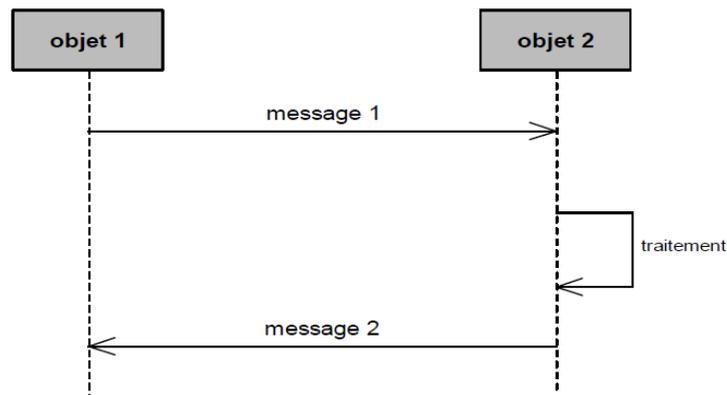


c. Relation de dépendance : Relation entre éléments du modèle ne nécessitant pas forcément un lien entre objets. Lorsque cette relation est réalisée par des liens entre objets, ces derniers sont limités dans le temps, contrairement à d'autres relations plus structurales.

Un élément A dépend d'un élément B, lorsque A utilise des services de B. De ce fait, tout changement dans B peut avoir des répercussions sur A.

2.3. Diagramme de séquence [10]

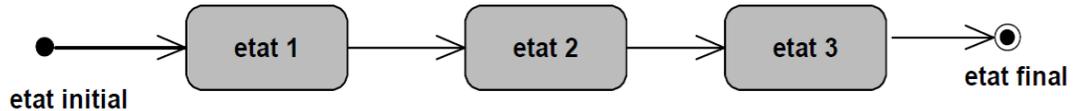
Le diagramme de séquences montre les interactions entre objets selon un point de vue temporel. La représentation du contexte des objets se concentre sur l'expression des interactions. Un objet est matérialisé par une barre verticale appelée ligne de vie des objets. Les objets, communiquent en échangeant des messages représentés au moyen de flèches orientées, de l'émetteur du message vers le destinataire. L'ordre des messages est donné par leur position sur l'axe vertical.



2.4. Diagrammes d'activités :

Le diagramme d'activité est attaché à une catégorie de classe il décrit le déroulement des activités de cette catégorie et il décrit le comportement d'une opération en terme d'actions.

Le déroulement s'appelle "**flot de contrôle**". Il indique la part prise par chaque objet dans l'exécution d'un travail. Il sera enrichi par les conditions de séquençement.



2.5 Les autres diagrammes UML [12] :

- **Le diagramme de collaboration** qui est une représentation spatiale des objets, des liens et des interactions.
- **Le diagramme de composants** qui décrit les composants physiques d'une application.
- **Le diagramme de déploiement** qui décrit les composants sur les dispositifs matériels.
- **Le diagramme d'états transitions** qui décrit le comportement d'une classe en terme d'états.
- **Diagrammes d'objets** permet de mettre en évidence des liens entre les objets. Les objets, instances de classes, sont reliés par des liens, instances d'associations.

B. Modèle Conceptuel de Données (MCD) :

Le modèle conceptuel des données (**MCD**) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

1. Entités et classe d'entité :

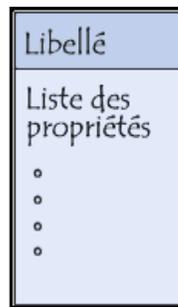
Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire.

On appelle classe d'entité un ensemble composé d'entités de même type, c'est-à-dire dont la définition est la même. Le classement des entités au sein d'une classe s'appelle classification (ou abstraction). Une entité est une instantiation de la classe. Chaque entité est composée de propriétés, données élémentaires permettant de la décrire.

Prenons par exemple une Ford Fiesta, une Renault Laguna et une Peugeot 306. Il s'agit de 3 entités faisant partie d'une classe d'entité que l'on pourrait appeler voiture. La Ford Fiesta est donc une instantiation de la classe voiture. Chaque entité peut posséder les propriétés couleur, année et modèle.

Les classes d'entités sont représentées par un rectangle. Ce rectangle est séparé en deux champs :

- le champ du haut contient le libellé. Ce libellé est généralement une abréviation pour une raison de simplification de l'écriture. Il s'agit par contre de vérifier qu'à chaque classe d'entité correspond un et un seul libellé, et réciproquement.
- le champ du bas contient la liste des propriétés de la classe d'entité

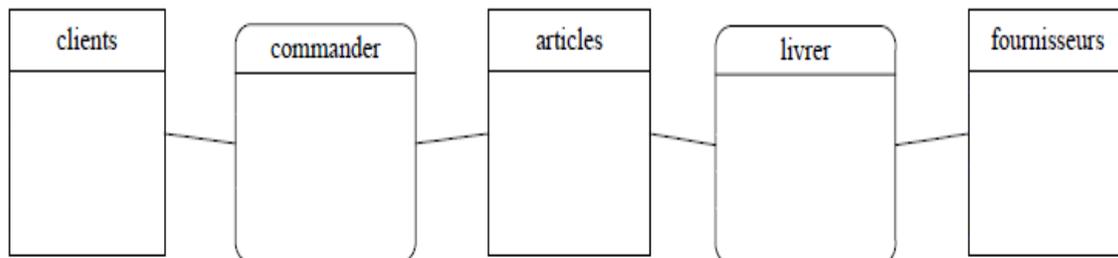


2. Relations et classes de relation :

e relation (appelée aussi parfois association) représente les liens sémantiques qui peuvent exister entre plusieurs entités. Une classe de relation contient donc toutes les relations de même type (qui relient donc des entités appartenant à des mêmes classes d'entité). Une classe de relation peut lier plus de deux classes d'entité. Voici les dénominations des classes de relation selon le nombre d'intervenants :

- une classe de relation **récursive** (ou réflexive) relie la même classe d'entité
- une classe de relation **binaire** relie deux classes d'entité
- une classe de relation **ternaire** relie trois classes d'entité
- une classe de relation **n-aire** relie n classes d'entité

Les classes de relations sont représentées par des hexagones (parfois des ellipses) dont l'intitulé décrit le type de relation qui relie les classes d'entité (généralement un verbe). On définit pour chaque classe de relation un identificateur permettant de désigner de façon unique la classe de relation à laquelle il est associé.

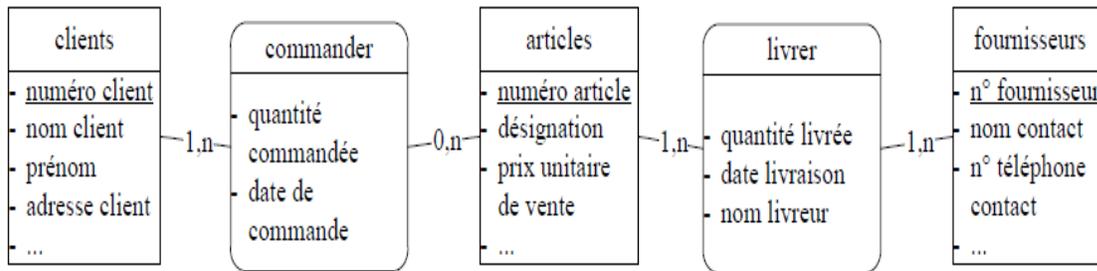


On peut éventuellement ajouter des propriétés aux classes de relation.

3. Cardinalités :

Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. La cardinalité d'une relation est composée d'un couple comportant une borne maximale et une borne minimale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur :

- la borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation
- la borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation



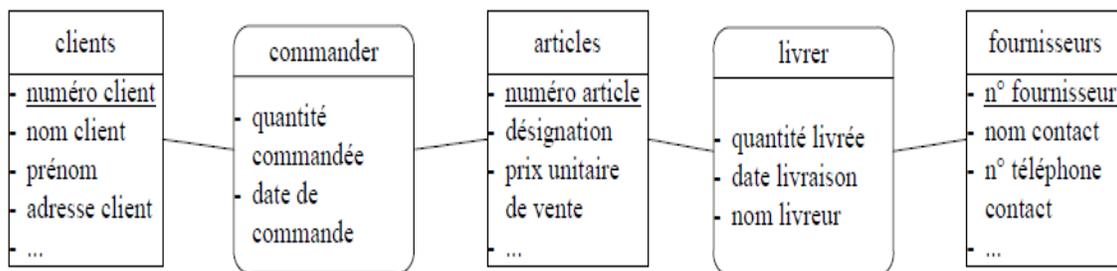
Une cardinalité 1.N signifie que chaque entité appartenant à une classe d'entité participe au moins une fois à la relation. Une cardinalité 0.N signifie que chaque entité appartenant à une classe d'entité ne participe pas forcément à la relation.

4. Les identifiants :

Un identifiant est un ensemble de propriétés (une ou plusieurs) permettant de désigner une et une seule entité.

Les attributs d'une classe d'entité permettant de désigner de façon unique chaque instance de cette entité sont appelés **identifiants absolus**.

Le modèle conceptuel des données propose de faire précéder d'un # les identifiants (parfois de les souligner).



Ainsi, chaque classe d'entité doit posséder au moins un attribut identifiant, et l'ensemble de ses attributs identifiants doivent être renseignés à la création de l'entité.

Conclusion :

Cette annexe a proposé une présentation brève des principaux concepts de modélisation d'UML et son extension pour le web ainsi que leur notation et quelques notions de bases sur le MCD (Modèle Conceptuel de Données).

Bibliographie

- [1] Larry L. PETERSON, Bruce S. DAVIE, "Réseaux d'ordinateurs : une approche système". Vuibert, 1998.
- [2] David James Clarke, CNE Guide de formation.
- [3] Andrew TANNENBAUM, LES RESEAUX, Prentice Hall, 1999.
- [4] George et Olivier GARDARIN, "Le Client-Serveur".Eyrolles, 2000.
- [5] <http://www.wikipedia.org>.
- [6] <http://www.Algerie-dz.com>.
- [7] <http://www.vitamedz.com>.
- [8] <http://www.univ-avignon.fr>.
- [9] Arrêté n° 136 du 26 Jomada Ethania 1430 H correspondant au 20 juin 2009 fixant les règles d'organisation et de gestion pédagogiques commune aux études conduisant aux diplômes et de Master. Et Arrêté n° 137 du 20 juin 2009 correspondant au 26 Jomada Ethania 1430 H portant modalités d'évaluation, de progression et d'orientation dans les cycles d'études conduisant aux diplômes de licence et de master.
- [10] <http://www.ummt0.dz/spip.php?rubrique261>
- [11] COURS UML livre de Frédéric Di Gallo/ Laurent Piechocki.
- [12] UML, le langage de modélisation objet unifié Par Laurent Piechocki.