



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE



Mémoire

De fin d'études

En vue de l'obtention de diplôme de Master
(LMD) en informatique.

Option : Ingénierie en système d'information

Thème:

**Conception et réalisation d'une application
web avec J2EE pour la gestion de scolarité**

Proposé et dirigé par :

M^RY.CHAIEB

Réalisé par :

M^{elle} AMRANE SAMIA

&

M^{elle} AMRANE HASSINA

Promotion : 2012/2013

REMERCIEMENTS



En premier lieu, nous remercions Dieu qui nous a procuré ce succès.



Nous tenons à remercier vivement notre promoteur Mr – Chaieb pour ses conseils précieux et pour toutes les commodités et aises qu'il nous a apportées durant notre étude et réalisation de ce projet.



Nos remerciements les plus vifs s'adressent aussi aux messieurs le président et les membres de jury d'avoir accepté d'examiner et d'évaluer notre travail.



que nos chers parents et familles, et bien avant tout, trouvent ici l'expression de nos remerciements les plus sincères et les plus profonds en reconnaissance de leurs sacrifices, aides, soutien et encouragement afin de nous assurer cette formation de master dans les meilleures conditions.



Et enfin, nous exprimons également notre gratitude à tous les professeurs et enseignants qui ont collaboré à notre formation depuis notre premier cycle d'étude jusqu'à la fin de notre cycle universitaire.

Hassina et Samia



Dédicaces

*Je dédie ce modeste travail
À mes chers parents, mes chers frères et mes chères sœurs.
À tous mes amis.
À toute la promotion ISI
À mon binôme et toute sa famille
À tous ceux qui m'ont aidé durant ma vie universitaire.*

Samia

*Je dédie ce modeste travail
À mes chers parents, mes chers frères et mes sœurs.
À tous mes amis, on particulier M.Amani qui était toujours
là pour m'encourager et pour me soutenir...
À toute la promotion ISI
À mon binôme et toute sa famille
À tous ceux qui m'ont aidé durant ma vie universitaire.*

Hassina

Sommaire


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

Chapitre I

Introduction	2
I. L'architecture logicielle.....	2
1. Vue Logique	2
2. Vue Composants.....	2
3. Vue Processus.....	2
4. Vue Déploiement.....	2
5. Vue Use-Cases.....	3
II. Architectures distribuées.....	3
III. Les architectures client/serveur	4
III.1.Définition.....	4
III.2. Notions de bases.....	4
III.3.Fonctionnement d'un système Client/serveur.....	4
III.4. Caractéristiques de l'architecture Client-serveur.....	5
III.5. Types de serveurs.....	6
III.6.Les différentes architectures client/serveur	7
III.6.1. L'architecture à deux niveaux.....	7
III.6.2. L'architecture à 3 niveaux.....	8
III.6.3. Architecture multi niveaux.....	9
III.7. Les avantages et les inconvénients de l'architecture Client/ Serveur	9
III.8. Middleware.....	10
III.8.1. Définition.....	10
VI. Internet et le web.....	12
IV.1. l'Internet.....	12
IV.1.1. Définition.....	12
IV.1.2. Les protocoles d'Internet.....	12
IV.1.3. les services internet.....	13
IV.2. Intranet-Extranet.....	14
IV.3. Le web.....	15
IV.3.1. Définition.....	15
IV.3.2. Les principaux termes du web.....	15
IV.3.3. Les sites web.....	16
IV.3.3.1. Les sites web statiques.....	16
IV.3.3.2. Les sites web dynamiques.....	16
VI.3.4. Les technologies de programmation web.....	16
VI.3.4.1. Côté client	16
VI.3.4.2. Côté serveur.....	17
Conclusion.....	17

Sommaire

Chapitre II

Introduction	18
I. Un bref historique de J2EE.....	18
II. Présentation de J2EE	18
III. L'architecture de J2EE.....	19
IV. Les couches logicielles JEE.....	20
V. Spécifications de la plate forme JEE.....	22
V.1. Les APIs.....	22
V.1.1 Les composants	22
V.1.2. Services d'infrastructures.....	26
V.1.3. Services de communication.....	27
 Les conteneurs.....	27
V.2. Les serveurs d'applications.....	28
V.2.1. Le rôle d'un serveur d'application.....	28
VI Les avantages d'utiliser JEE.....	29
Conclusion	30

Chapitre III

Introduction	32
I. Présentation d'UML.....	32
I. 1. Définition	32
I .2. Les diagrammes UML.....	32
II. Analyse	33
II.1. Identification des besoins	33
II.2. Identification des acteurs.....	33
II.3. Identification des cas d'utilisation	34
II.4. Diagramme de contexte.....	35
II.5. spécification des scénarios	36
II.6. Spécification de quelques cas d'utilisation.....	39
II.7. Diagramme de cas d'utilisation	43
III. La conception.....	46
III.1. Diagramme de séquence de réalisation des cas d'utilisation	46
III.2. Diagrammes d'activités.....	53
III.2.1. Diagramme d'activités du cas d'utilisation « S'authentifier »	53
III.2.2. Diagramme d'activités du cas d'utilisation « Consulter les PV globaux »	54
III.2.3. Diagramme d'activités du cas d'utilisation « Enregistrement des notes des étudiants ».....	55
III.2.4. Diagramme d'activités du cas d'utilisation « Changement de mot de passe » ...	56
III.2.5. Diagramme d'activités du cas d'utilisation « ajout d'un module ».....	57

Sommaire

III.2.6. Diagramme d'activités du cas d'utilisation « édition des relevés de notes»	58
III.2.7. Diagramme d'activités du cas d'utilisation « création d'un compte»	59
III.3. Diagramme de classe globale de donnée	60
III.4. Création de la base de données	59
III.4.1 définition de la base de données.....	61
III.4.2. Passage au modèle relationnel	61
Conclusion	64

Chapitre IV

Introduction.....	66
I. Les langages de programmation.....	66
I.1. HTML.....	66
I.2. Le langage JAVA.....	66
I.3. Java Script.....	67
II. Environnement et outils de développement	67
II.1. Macromedia Dreamweaver.....	67
II.2. Macromedia flash	67
II.3. IDE NetBeans 6.8.....	68
II.4. Les Serveurs	69
II.4.1. Présentation de MySQL Serveur 5.4.....	69
II.4.2. Le serveur apache	70
II.4.3. Le module Tomcat	70
III. Présentation de quelques interfaces et fonctionnalités de l'application	71
Conclusion.....	77
Conclusion générale.....	79
Annexe A.....	81
Annexe B.....	88

Liste des figures

Chapitre I

Figure (I.1) : Vues de l'architecture logicielle selon P. Kruchten	3
Figure (I.2) : Principe de fonctionnement du Client-serveur.....	5
Figure (I.3) : L'architecture client /serveur à 2 niveaux.....	8
Figure (I.4) : L'architecture client /serveur à 3 niveaux.....	8
Figure (I.5) : L'architecture client /serveur à multi niveaux.....	9
Figure (I.6) : Organisation d'un intergiciel.....	11

Chapitre II

Figure (II.1) : L'architecture de J2EE.....	20
Figure(II.2) : Architecteur trois tiers.....	21
Figure(II.3) : Cycle de vie d'une servlet.....	24

Chapitre III

Figure (III.1): La démarche de modélisation de l'application.....	33
Figure (III.2) : Diagramme de contexte de l'application.....	36
Figure (III.3) : Diagramme global de cas d'utilisation relatif à l'étudiant.....	43
Figure(III.4) : Diagramme global de cas d'utilisation relatif à l'enseignant	44
Figure (III.5) : Diagramme global de cas d'utilisation relatif à l'Agent de scolarité.....	44
Figure(III.6) : Diagramme global de cas d'utilisation relatif au chef de département	45
Figure (III.7) : Diagramme global de cas d'utilisation relatif à l'administrateur	45
Figure(III.8) : Diagramme de séquence de réalisation de cas d'utilisation « Authentification Etudiant ».....	46
Figure(III.9) : Diagramme de séquence de réalisation de cas d'utilisation « Consultation des PV modulaire».....	47
Figure(III.10) : Diagramme de séquence de réalisation de cas d'utilisation « enregistrement des notes ».....	48

Liste des figures

Figure(III.11) : Diagramme de séquence de réalisation de cas d'utilisation « Changement de mot de passe ».....	49
Figure(III.12) : Diagramme de séquence de réalisation de cas d'utilisation «Ajout d'un module ».....	50
Figure(III.13) : Diagramme de séquence de réalisation de cas d'utilisation « édition des relevés de notes».....	52
Figure(III.14) : Diagramme de séquence de réalisation de cas d'utilisation « ajouter un compte».....	53
Figure (III.15) : Diagramme d'activités du cas d'utilisation « authentification étudiant».....	54
Figure(III.16) : Diagramme d'activités du cas d'utilisation «Consulter les PV modulaire »...55	
Figure(III.17) : Diagramme d'activités du cas d'utilisation «enregistrement des notes des étudiants».....	56
Figure(III.18) : Diagramme d'activités du cas d'utilisation «changement de mot de passe».....	57
Figure(III.19) : Diagramme d'activités du cas d'utilisation «Ajout d'un module ».....	58
Figure(III.20) : Diagramme d'activités du cas d'utilisation « édition de relevé de notes »....	60
Figure(III.21) : Diagramme d'activités du cas d'utilisation « création d'un compte».....	59
Figure(III.22) : Diagramme de classe globale	60

Chapitre IV

Figure(IV .1) : Interface Principale de Dreamweaver8.....	67
Figure(IV.2): Interface Macromedia flash.....	68
Figure (IV.3): Interface de Neatbeans6.8.....	69
Figure (IV.4): Interface principale MySQL Serveur 5.4.....	70
Figure(IV.5): Interface d'Apache Tomcat.....	71
Figure (IV.6): Page d'accueil Principale.....	72
Figure (IV.7): Page authentification de l'enseignant.....	72

Liste des figures

Figure (IV.8) : consulter PV modulaire.....	73
Figure (IV.9) : Affichage de PV modulaire.....	73
Figure(IV.10): Consulter relevé de notes.....	74
Figure(IV.11): édition d'un relevé de notes.....	74
Figure (IV.12): Ajout d'un module.....	75
Figure (IV.13) : Ajout d'un enseignant sans remplir tout les champs.....	75
Figure (IV.14): Ajout d'un enseignant.....	76
Figure (IV.15): Liste des enseignants	76

Liste des tableaux

Chapitre III

Tableau (III .1) : Les différentes tâches des acteurs	35
Tableau (III .2) : Les Scenarios des acteurs	39

Introduction générale

Introduction générale

Il ne fait désormais plus aucun doute que l'informatique représente la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité en ce siècle passé. En effet, loin d'être un éphémère phénomène de mode, ou une tendance passagère, l'informatique vient nous apporter de multiples confort à notre mode de vie. Aucun domaine n'est resté étranger à cette stratégie qui offre tant de services aussi bien pour l'entreprise ou l'administration que pour le personnel.

Mais, au delà de l'utilisation individuelle de l'informatique, c'est surtout la mise en communication des ordinateurs, qui a permis de révolutionner les méthodes de travail. Ainsi, on a assisté à l'émergence des réseaux. Ce nouveau progrès offre aux utilisateurs de nouveaux outils de travail et leur permet d'améliorer leur rentabilité et leur productivité.

La révolution des nouvelles technologies s'impose tant dans la vie domestique que professionnelle ; elle touche pratiquement tous les secteurs de la société, de l'économie,...etc. Cette évolution a aussi touchée l'univers des études et en plus particulier le monde de l'université.

C'est dans ce cadre d'idées que s'inscrit notre projet de fin d'études : Concevoir et développer une application web pour prendre en charge les enseignements et les étudiants au sein du département mathématique, cette application permettra aux enseignants l'édition des PV modulaires, voir les affectation, elle permettra au chef de département d'affecte les enseignants, consultation des PV de délibérations et changer son mot de passe, elle permettra à l'agent de scolarité l'édition des listes des étudiants, l'édition des PV globaux de délibérations, l'édition des relevés de notes et elle permettra aussi à l'administrateur la création de comptes, modification de comptes, et de permettre à l'étudiant de consulté ses notes et même ses relevés de notes et de changer son mot de passe.

Afin de mener à bien notre projet, nous avons réparti le contenu de notre travail en quatre chapitres, comme suit :

- Le premier chapitre sera une présentation générale des nouvelles technologies : présentent quelques notions de base concernant l'architecture logicielle, l'Internet, le web et le modèle Client/serveur.
- Le deuxième chapitre, consiste à présenter la plateforme J2EE.
- Le troisième chapitre est consacré à l'analyse et la conception de notre système à travers son architecture globale puis détaillée en définissant les différentes fonctionnalités attendues du système. Et pour permettre de détailler toutes les étapes de travail, une modélisation est faite avec le langage UML.
- Le quatrième chapitre, consiste à présenter la réalisation du système. On commence par décrire les outils et l'environnement de développement utilisés puis les différentes interfaces et une brève explication pour chacune d'elle pour donner une vue générale et complète sur le système.

Nous terminerons par une conclusion générale et deux annexes :

Annexe A : UML.

Annexe B : Généralité sur les réseaux.

Chapitre 1

Les nouvelles technologies

Introduction :

Les nouveaux réseaux de télécommunication rapides et à large bande passante (RNIS, ATM, High Performance Ethernet) ont permis le regroupement des ordinateurs en réseaux, d'abord de taille réduite (entreprise), puis moyenne (nationale) puis finalement mondiale avec Internet. Tout ceci a favorisé la démocratisation des communications informatiques (à commencer par les messageries électroniques et la navigation sur le Web) puis, plus récemment, l'émergence des applications distribuées.

Le présent chapitre est une introduction aux nouvelles technologies de l'information. Dans un premier temps, Nous allons définir c'est quoi l'architecture logicielle 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).

I. L'architecture logicielle : [01]

L'architecture logicielle est un domaine dans lequel s'est épanoui au cours des dix dernières années le paradigme objet ainsi que les langages et méthodologies associés (UML, xUP, Design- Patterns, XP...). L'état de l'art de la matière est maintenant relativement stable, et les méthodologies d'ingénierie logicielle dignes des méthodologies de production industrielle.

D'après P. Kruchten²⁷, l'architecture logicielle se représente selon « 4+1 » vues :

1. **Vue Logique** : elle se concentre sur la modélisation des principaux éléments d'architecture et mécanismes logiciels, la définition des éléments du domaine (objets métier, diagrammes d'états de ces objets...). Elle comprend en particulier les modèles d'analyse et de conception du système, c'est la vue des équipes de conception et développement.
2. **Vue Composants** (ou vue Implémentation) : elle permet d'identifier les modules (composants logiciels) qui implémentent les éléments définis dans la vue logique, de les regrouper en composants logiciels, d'identifier les dépendances d'intégration entre ces composants ; c'est la vue des équipes d'intégration.
3. **Vue Processus** : en environnement multi-tâches, cette vue de l'architecture permet de définir les processus, la coordination et la synchronisation des processus, les threads d'exécution. Cette vue est optionnelle et n'est utile que dans le cas d'architectures complexes multi-tâches.
4. **Vue Déploiement** : elle précise l'architecture de production (ressources matérielles, implantation des composants, pilotage...)

Cette vue permettra par exemple de s'assurer que l'application répond aux contraintes de déploiement, aux exigences de qualité de service (montée en charge, temps de réponse, haute-disponibilité...) et s'intègre aux infrastructures de supervision, etc.

5. **Vue Use-Cases** (vue « +1 ») : elle se concentre sur un sous-ensemble des cas d'utilisation qui ont une influence significative sur l'architecture du système. Ces use-cases structurants permettent d'identifier les fonctionnalités et contraintes importantes,

les risques majeurs de l'architecture, ce sont eux qui guident l'élaboration des quatre autres vues de l'architecture, de la conception à la mise en production de l'application.

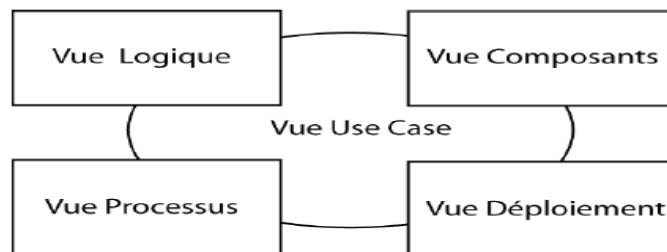


Figure (I.1): Vues de l'architecture logicielle selon P. Kruchten

II. Architectures distribuées [02]:

Qu'est-ce qu'une architecture distribuée? Alors que le terme n'a pas été défini de manière officielle ni par la NIST (National Institute of Standards and Technology) ni par l'AFNOR (Association française de normalisation), nous nous sommes engagés à explorer quel est le vrai sens de ce terme.

Définie par Wikipedia comme toute architecture dont les ressources ne se trouvent pas au même endroit ou sur la même machine, le concept d'architecture distribuée (ou informatique distribuée) s'oppose habituellement à celui d'architecture centralisée représentée par les architectures traditionnelles client-serveur.

Les applications à architectures distribuées sont des applications dont les fonctions sont réparties entre plusieurs systèmes. On les appelle aussi architectures multi-tiers (ou multicouches)

Chaque système contient une partie de l'application, les parties manquantes sont exécutées sur les autres systèmes participants à l'application et les informations sont échangées par le réseau.

Une application répartie est constituée d'un ensemble de composants qui communiquent entre eux par l'échange de requêtes. Cette communication est assurée par l'Intergiciels, qui offre des mécanismes de communication et de transport pour la localisation et l'activation des composants distribués. L'Intergiciels est le vecteur d'interaction des composants distribués.

III. Les architectures client/serveur [03]:

III.1 Définition :

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

III.2. Notions de bases :

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 est au milieu assure les dialogues entre les clients et les serveurs souvent hétérogènes.

III.3. Fonctionnement d'un système Client/serveur[04]

Un système client/serveur fonctionne selon le schéma suivant :

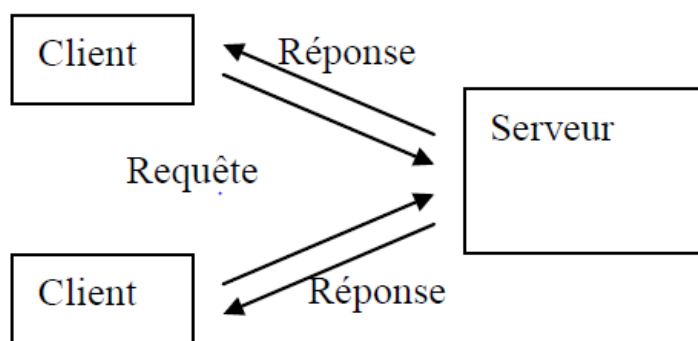


Figure (I.2) : Principe de fonctionnement du Client/Serveur

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.

III.4. Caractéristiques de l'architecture Client/serveur [05]

Le système client/serveur partage les caractéristiques suivantes :

- **Service** : Le modèle Client/serveur est essentiellement une relation entre des processus. Le processus serveur est un fournisseur de services et le client en est le consommateur ; Le modèle Client/serveur établit ainsi une séparation claire des rôles à partir de la notion du service.
- **Partage de ressources** : Un serveur peut traiter plusieurs clients à la fois et leurs accès aux ressources.
- **Asymétrie des protocoles** : La relation entre le client et le serveur est de type plusieurs vers un, toutefois, le client est le déclencheur du dialogue en demandant un service alors que le serveur attend passivement les requêtes.
- **Assemblage multi-vendeur** : Le Client-serveur est indépendant de la plate forme matérielle ou du système d'exploitation. On doit toujours pouvoir mélanger et apparier les plates formes Client-serveur.
- **Echange de messages** : Le client et le serveur sont des systèmes à liaison épisodique qui interagissent au moyen de messages. Le message est un mécanisme d'émission de demandes de services et de réponses à celles-ci.
- **Encapsulation des services** : Le serveur est un spécialiste, un message lui indique quel service est requis et c'est à lui de décider comment rendre ce service. Les serveurs peuvent être mis à niveau sans effet sur les clients tant que l'interface des messages reste la même.
- **Intégrité** : Le code et les données du serveur sont gérés de façon centralisée, ce qui garantit un moindre coût de maintenance et une meilleure intégrité des données tandis que les clients restent individuels et indépendant.
- **Souplesse et adaptabilité** : On peut modifier le module serveur sans toucher au module client et vice versa. Si une station est remplacée par un modèle plus récent, on modifie le module client sans modifier le module serveur.

III.5. Types des serveurs :

Les différents types de serveurs de l'architectures client / serveur peuvent être classées en fonction du service qu'ils fournissent aux utilisateurs :

- **Serveur de fichier (NFS, FTP):**

Dans le cas de serveur de fichiers, le client requiert des enregistrements de fichiers en émettant des requêtes sur le réseau en direction d'un serveur de fichier.

Caractéristiques :

- Très utilisé à ce jour (partage de fichiers sur le réseau).
- Forme primitive de service de données.
- Nombreux échanges de messages sur le réseau pour obtenir le résultat.
- Indispensable pour les banques de documents, d'images etc.

- **Serveurs de bases de données :**

Dans le cas d'un serveur de bases de données, le client émet des requêtes SQL sous forme de message (qui contiennent des commandes SQL) et le serveur renvoie le résultat de chaque requête (les données reçues sont le résultat de l'exécution qui se fait par le serveur de ces commandes).

Caractéristiques :

- Meilleure répartition de la puissance : le serveur utilise sa capacité de traitement (SGBD) pour sélectionner les réponses au lieu de transmettre tous les articles au client et de le laisser en faire la sélection.
- Nécessité d'écrire du code pour l'application cliente.
- C'est la base des systèmes d'aide à la décision.

Ex. : Oracle, Microsoft Server SQL

- **Serveur de transactions :**

Dans ce modèle, le client invoque des procédures distantes résidant (résidentes) sur le serveur qui comporte un moteur de bases de données SQL. Chaque procédure exécute un ensemble d'instructions SQL. L'échange sur le réseau consiste en un seul message requête / réponse (contrairement à l'application serveur base de données pour laquelle le message requête / réponse est émis pour chaque instruction).

Pour ce type de serveurs l'application client / serveur nécessite du code source au niveau du serveur. Il existe deux formes de transactionnel :

- Transactionnel léger : les procédures sont fournies par l'éditeur du SGBD.
- Transactionnel lourd : moniteurs transactionnels fournis par l'éditeur d'applications OLTP.

- **Serveur de groupware :**

Ce type de serveur s'intéresse à la gestion d'informations semi-structurées. Il est fondé sur 5 technologies de base :

- gestion de documents multimédia.
- workflow.
- courrier électronique.
- gestion de conférences.
- planification de réunions.

- **Serveur d'objets :**

Dans ce type de serveur, l'application client / serveur est écrite sous forme d'objets communicants. Les objets clients communiquent avec les objets serveurs au moyen d'un négociateur de requêtes objet ou ORB (Object Request Broker).

Fonctionnement :

- Le client appelle une méthode appartenant à une classe du serveur objet.
- L'ORB localise une instance de la classe, appelle la méthode demandée et renvoie le résultat à l'objet client. Les serveurs d'objets doivent bien sûr traiter le partage des objets.

- **ServeurWeb :**

Le serveur web est un processus capable de traiter des requêtes http issues des clients web ;c'est-à-dire des navigateurs. Ce processus est présent sur un serveur physique accessible par les ordinateurs du réseau via TCP / IP. Ce serveur contient les informations qui peuvent être mises à disposition des utilisateurs sous forme de fichiers HTML localisés dans un ou plusieurs périphériques de stockage, généralement des disques durs.

III.6. Les différentes architectures client/serveur : [06] :**III.6.1.L'architecture à deux niveaux**

Dans une architecture à deux niveaux (deux tiers), encore appelée client/serveur de première génération ou encore client/serveur de données, le poste client se contente de déléguer la gestion de données à un service spécialisé.

Le cas typique de cette architecture est une application de gestion fonctionnant sous Windows ou Linux et exploitant un SGBD centralisé. Le client demande un service au serveur et ce dernier l'exécute et lui fournit la réponse directement sans faire appel à un serveur intermédiaire.

Cette architecture fonctionne suivant le schéma suivant :

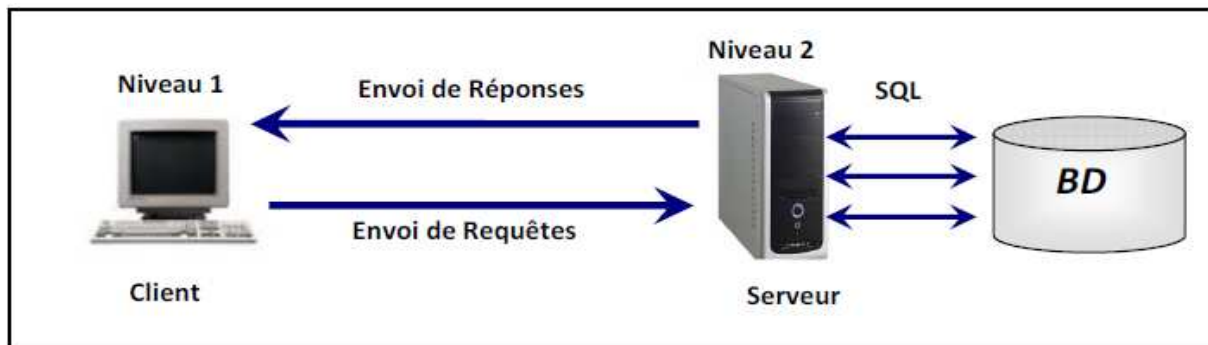


Figure (I. 3) : L'architecture client /serveur à 2 niveaux

III.6.2. L'architecture à 3 niveaux

Dans l'architecture à trois niveaux (appelée architecture 3 tiers), il existe un niveau intermédiaire, c'est à dire que l'on a généralement une architecture partagée entre le client, le serveur d'application (appelé aussi middleware) qui est le serveur chargé de fournir la ressource mais faisant appel à un autre serveur, et le serveur secondaire (généralement

unserveur de bases de données) fournissant un service au premier serveur. Les applications de l'architecture client/serveur à trois niveaux sont plus faciles à déployer et à gérer sur le réseau, elles essaient de minimiser les échanges sur le réseau en créant des niveaux de service. Elle est recommandée pour diverses raisons dont on cite celle de l'Internet. L'architecture 3 tiers fonctionne selon le schéma suivant :

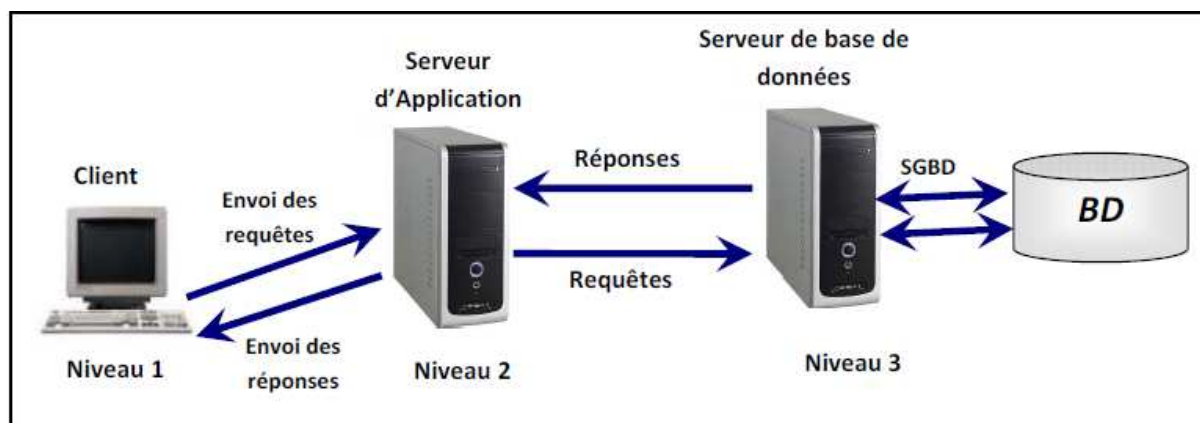


Figure (I.4) : L'architecture client /serveur à 3 niveaux

III.6.3. Architecture multi niveaux :

L'architecture multi niveaux fonctionne comme l'architecture trois niveaux sauf qu'il peut y avoir des niveaux en plus, c'est-à-dire que le serveur secondaire fait appel à un autre serveur pour fournir les ressources.

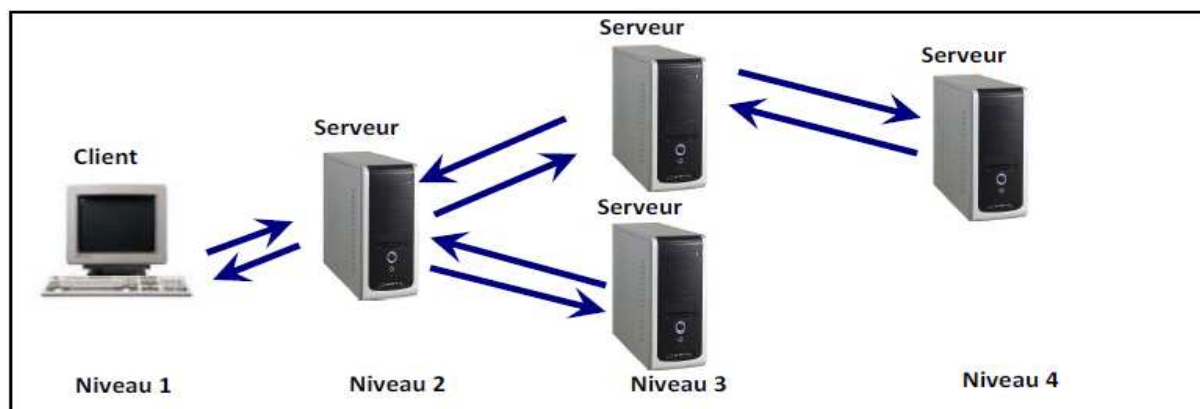


Figure (I.5) : L'architecture client /serveur à multi niveaux

III.7. Les avantages et les inconvénients de l'architecture Client/ Serveur :

a- Les avantages :

- Capacité de traitement** : Le modèle Client/ Serveur s'appuie sur les progrès réalisés dans le monde des microprocesseurs. Les techniques qu'il met en œuvre nécessitent des capacités de traitement importantes que les microprocesseurs d'aujourd'hui sont capables de fournir.

- **Standardisation** : En s'appuyant sur les normes standards, le modèle Client/ Serveur assure la portabilité des applications entre systèmes distincts et garantit leurs inter portabilité.
- **Capacité de mémorisation** : Dans une architecture Client/ Serveur, les postes serveurs doivent gérer simultanément les contextes applicatifs de plusieurs postes clients. Cette architecture sera autant mieux adapter à l'environnement transactionnel lourd où les serveurs disposent d'une grande capacité mémoire.
- **Productivité du développeur** : L'environnement Client/ Serveur facilite le processus de développement applicatif. Cela est dû notamment à la séparation fonctionnelle qui constitue les principes de base du modèle Client/ Serveur.
- **Portabilité et interopérabilité** : Le modèle Client/ Serveur assure le partage d'une même application sur différentes plates formes, échanger des informations et faire interagir plusieurs applications dans des environnements hétérogènes.
- **Des ressources centralisées** : Le serveur peut gérer les ressources communes des utilisateurs, car il est au centre du réseau. Ses ressources peuvent êtres logicielles (base de donnée,...) ou matérielles (imprimante,...).
- **Une meilleure sécurité** : Le nombre de nœuds permettant l'accès aux données du serveur est limité.
- **Une administration au niveau serveur** : Les clients ayant peut d'importance dans ce modèle, ils ont moins besoin d'être administrés.
- **Un réseau évolutif** : Cette architecture nous permet de supprimer ou d'ajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures.

b- Les inconvénients :

L'architecture Client/ Serveur a tout de même quelques lacunes parmi lesquelles, dont on peut citer :

- **Complexité de mise en œuvre** : Une complexité qui se manifeste notamment lors du découpage des traitements entre partie client et partie serveur.
- **Fragilité relative du système** : Dans un environnement centralisé, les données sont concentrées sur un seul endroit, le contexte client/serveur peut nécessiter l'installation des données sur des serveurs distincts. Cette répartition géographique des données et l'utilisation même d'un réseau, fragilise l'ensemble du système et pose alors le problème de la sécurité des données en environnement réparti.
- **Un cout très élevés** : La mise en place d'une architecture client /serveur nécessite des efforts financiers importants, investissement en matériels et logiciels tel que : le micro, interface graphique et matériels de connexion.

III.8. Middleware:[07]

III.8.1. Définition :

L'intergiciel (middleware en anglais) est un ensemble de logiciels ou detechnologies informatiques qui servent d'intermédiaire entre les applicationset le transport des données via le réseau. Ils offrent des services de hautniveau liés aux besoins de communication des applications (temps réel,sécurisation, sérialisation, transaction informatique,etc.)

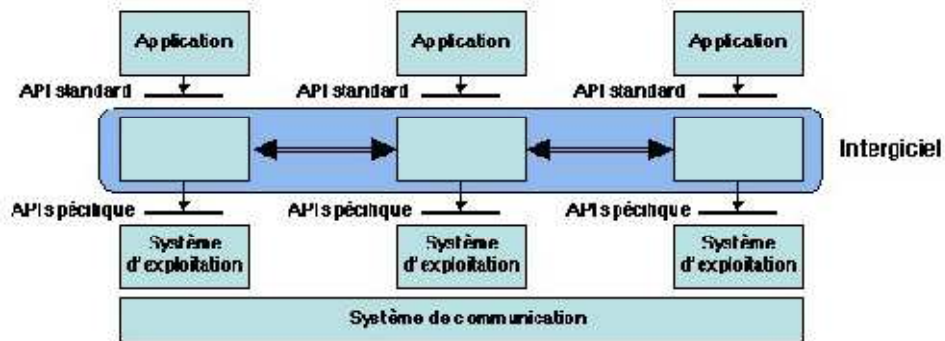


Figure (I.6) : Organisation d'un intergiciel

La définition du Middleware généralement admise est la suivante :

Un Middleware est un logiciel de communication qui permet à plusieurs processus s'exécutant sur une ou plusieurs machines d'interagir à travers un réseau.

Il est un complément de services du réseau permettant la réalisation du dialogue client/serveur :

- Prend en compte les requêtes de l'application cliente ;
- Les transmet de manière transparente a travers le réseau jusqu'au serveur ;
- Prend en comptes les données résultantes du serveur vers l'application cliente .

L'objectif essentiel d'un middleware (intergiciel) est d'offrir une interface unifiée permettant l'accès à l'ensemble des services disponibles sur le réseau. Il réalise les taches suivantes :

1. cacher la répartition, c'est-à-dire, le fait qu'une application est constituée de parties interconnectées s'exécutant a des emplacements géographiquement repartis ;
2. cacher l'hétérogénéité des composants matériels, des systèmes d'exploitation et des protocoles de communication utilises par les différentes parties d'une application ;
3. fournir des interfaces uniformes, normalisées, et de haut niveau aux équipes de développement et d'intégration, pour faciliter la construction, la réutilisation, le portage et l'interopérabilité des applications ;

4. fournir un ensemble de services communs réalisant des fonctions d'intérêt général, pour éviter la duplication des efforts et faciliter la coopération entre applications.

III. 8.2 Types d'intergiciel:

- **Général :**
 - Protocoles de communication, répertoires répartis, services d'authentification, RPC, etc.
 - Services répartis de type NOS (Networked OS) : services de fichiers, services d'impression.
- **Spécifique :**
 - de bases de données : Oracle Net, JDBC
 - de groupware : MAPI, Lotus Notes
 - d'objets : CORBA, COM/DCOM, Java RMI
 - de composants : EJB, Web services à messages MOM (Message-Oriented Middleware) : IBM MQSeries, MS Message Queue Server MSMQ.

IV. Internet et le web :

IV.1. L'Internet : [08] :

IV.1.1. Définition: Internet signifie réseaux interconnectés (*interconnected networks*). Il désigne un ensemble de réseaux informatiques privés et publics interconnectés fonctionnant sur la base de protocoles dont le plus connu est TCP-IP. Ce protocole constitue un service sans ouverture de connexion au préalable et sans garantie de fiabilité.

IV.1.2. Les protocoles d'Internet

- **Le protocole TCP /IP** Ce protocole est en fait une combinaison de deux protocoles agissant chacun à un niveau différent.
 - **TCP :** Fonctionne en mode déconnecté, c'est-à-dire qu'une liaison est établie entre deux ordinateurs. C'est un protocole fiable dans la mesure où il est capable de déterminer si un paquet est arrivé ou s'il est interrompu, s'occupe de la transmission des paquets, vérifie l'ordre d'arrivée et l'intégrité des données.
 -
 - **IP :** est le nœud de transmission des données au dessous de TCP. Il s'occupe du transport des données, mais il ne contient aucun outil de contrôle de données ni de vérification des paquets, il permet d'adresser les paquets au bon endroit.
- **HTTP (HyperText Transport Protocol) :** C'est un protocole léger et rapide, utilisé pour délivrer des fichiers multimédia et hypertextes appelés plus généralement ressources, en utilisant Internet. Il gère la totalité des échanges réalisés entre le client et le serveur. Pour chaque demande d'accès à l'URL, une requête http est émise indépendamment par le client.
- **DNS (Domain Name Service) :** C'est un ensemble de protocoles permettant aux utilisateurs d'un réseau TCP/IP d'accéder aux hôtes à l'aide des noms conviviaux hiérarchisés, le serveur DNS assure la transformation des noms des domaines en

adresses IP et inversement, car la structure des adresses IP est complexe à manipuler pour cela on utilise le DNS, sans celui-ci les utilisateurs devraient mémoriser les adresses IP.

IV.1.3. les services d'Internet :

Internet offre aujourd'hui plein de services aux milliers d'utilisateurs connectés, tels l'accès au web, la messagerie électronique, les forums de discussions, le téléchargement de données, la recherche d'informations, la téléphonie, le e-commerce et bien d'autres.

❖ le courrier électronique :

C'est tout simplement un message écrit, échangé entre un ou plusieurs internautes ; Le courrier électronique permet de faire parvenir des messages à n'importe quel endroit dans le monde, de façon quasi instantanée ; il rend possible l'envoi de message à toute heure, sans avoir à se soucier de la disponibilité du destinataire.

❖ Les news d'Usenet :

Usenet (également connu sous le nom **Netnews**) est un système en réseau de forums, inventé en 1979 et basé sur le protocole NNTP. Il a rapidement été rendu utilisable via Internet où il reste au début du XXI^e siècle en usage.

Certains le trouvent moins simple d'emploi que les forums sur le web apparus dans les années 1990 mais il a pour principaux avantages de permettre une lecture plus rapide (car moins personnalisée et perturbée visuellement) et de faciliter la réplication des articles sur les serveurs du monde entier.

❖ Les moteurs de recherche :

Un moteur de recherche est un logiciel permettant de retrouver des ressources pages web, associées à des mots quelconques (clés). Certains sites Web offrent un moteur de recherche comme principale fonctionnalité ; on appelle alors moteur de recherche le site lui-même ; les plus connus s'appellent Google, Alta Vira, Lycos, Wanadoo, Yahoo ... Mais il y en a des milliers..... Un moteur de recherche permet de trouver les occurrences d'un mot à l'intérieur d'une base de données.

❖ Le transfert de fichiers :

L'Internet est utilisé comme une grande bibliothèque distribuée à travers la planète. Vue comme une bibliothèque de fichiers de tout type et en particulier de logiciels mis à la disposition de tous, ou qu'ils soient, grâce au service de transfert de fichier FTP (File Transfer Protocol). Ce service trouve son utilité dans les milieux universitaire et de la recherche.

IV.2. Intranet-Extranet [09]

IV.2.1. Définitions

Intranet :

L'intranet est un réseau informatique utilisé à l'intérieur d'une entreprise ou de toute autre entité organisationnelle utilisant les techniques de communication d'Internet (IP, serveurs HTTP). Dans les grandes entreprises, l'intranet fait l'objet d'une gouvernance particulière en raison de sa pénétration dans l'ensemble des rouages des organisations.

Un intranet repose généralement sur une architecture à trois niveaux composée : de clients (navigateur internet généralement), d'un ou plusieurs serveurs d'application (middleware): un serveur web permettant d'interpréter des scripts PHP, ASP ou autres, et les traduire en requêtes SQL afin d'interroger une base de données et d'un serveur de bases de données. De cette façon, les machines clientes gèrent l'interface graphique, tandis que les différents serveurs manipulent les données. Le réseau permet de véhiculer les requêtes et les réponses entre clients et serveurs.

Un intranet dans une entreprise permet de mettre facilement à la disposition des employés des documents divers et variés; cela permet d'avoir un accès centralisé et cohérent à les données de l'entreprise. De cette façon, il est généralement nécessaire de définir des droits d'accès pour les utilisateurs de l'intranet aux documents présents sur celui-ci, et par conséquent une authentification de ceux-ci afin de leur permettre un accès personnalisé à certains documents.

Des documents de tous types (textes, images, vidéos, sons, ...) peuvent être mis à disposition sur un intranet. De plus, un intranet peut réaliser une fonction de groupware très intéressante, c'est-à-dire permettre un travail coopératif. Voici quelques unes des fonctions qu'un intranet peut réaliser :

- Mise à disposition d'informations sur l'entreprise (panneau d'affichage)
- Mise à disposition de documents techniques
- Moteur de recherche de documentations
- Un échange de données entre collaborateurs
- Annuaire du personnel
- Gestion de projet, aide à la décision, agenda, ingénierie assistée par ordinateur
- Messagerie électronique
- Forum de discussion, liste de diffusion, chat en direct
- Visioconférence
- Portail vers internet
- Planification des réunions.

Ainsi, un intranet favorise la communication au sein de l'entreprise et limite les erreurs dues à la mauvaise circulation d'une information. L'information disponible sur l'intranet doit être mise à jour en évitant les conflits de version.

Extranet

Un extranet est une extension de l'entreprise à des partenaires situés au-delà du réseau et ce de manière sécurisée (authentification par un mot de passe). De cette façon un extranet n'est ni un Intranet ni un site web Internet, il s'agit d'un système supplémentaire offrant par exemple aux clients, à ses partenaires ou à des filiales un accès privilégié à certaines ressources informatiques de l'entreprise par l'intermédiaire d'une interface Web.

IV.3. Le web :[10]

Le Web à été inventé plusieurs années après Internet, mais c'est lui qui a contribué à l'explosion de l'utilisation d'Internet par le grand public, grâce à sa facilité d'emploi. Depuis, le Web est fréquemment confondu avec Internet alors qu'il n'est en réalité qu'un de ses services.

IV.3.1. Définition : 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.

IV.3.2. Les principaux termes du web

- **Page web** : Est un document destiné à être consulté avec un navigateur Web. Une page Web est toujours constituée d'une ressource centrale (généralement un document HTML) et d'éventuelles ressources liées automatiquement accédées (typiquement des images).
- **Site web** : Est un ensemble de pages Web et d'éventuelles autres ressources, liées dans une structure cohérente, publiées par un propriétaire (une entreprise, une administration, une association, un particulier, etc.) et hébergées sur un ou plusieurs serveurs Web.
- **URL (Uniforme Resource Locator)** : Pointe sur une ressource. C'est une chaîne de caractères permettant d'indiquer un protocole de communication et un emplacement pour toute ressource du Web.
- **HTML (HyperText Markup Language)** : C'est le langage utilisé pour écrire un document afin de le diffuser sur le web. Il est constitué d'un jeu de commandes (tags ou balises) assez simples qui déterminent la structure du document. Il permet de définir les éléments du document, mais pas son formatage et c'est au navigateur d'effectuer un formatage optimal en fonction de ses possibilités.
- **Hypertexte** : L'hypertexte est une manière d'organiser et de présenter l'information dans laquelle certains éléments du texte, appelés liens, permettent de se déplacer vers d'autres zones du texte ou vers une autre page.
- **Navigateur web** : C'est un logiciel client qui interprète les adresses des pages web, d'afficher et d'exploiter les liens hypertextes à l'intérieur de celles-ci. Il existe un certain nombre de navigateurs web mais les plus connus restent *Internet Explorer* et *Netscape*.

IV.3.3. Les sites web :

Un site web est un ensemble de pages web liées entre elles par une structure cohérente, publiées par un propriétaire, une entreprise, une administration, une association, un particulier ...etc. Qui sont hébergées sur un serveur particulier. On distingue deux types de sites web statiques et les sites web dynamiques.

IV.3.3.1. Les sites web statiques :

Ce sont des sites réalisés uniquement à l'aide des langages HTML et CSS. Ils fonctionnent très bien mais leur contenu ne peut pas être mis à jour automatiquement : il faut que le propriétaire du site (le webmaster) modifie le code source pour y ajouter des nouveautés. Ce n'est pas très pratique quand on doit mettre à jour son site plusieurs fois dans la même journée ! Les sites statiques sont donc bien adaptés pour réaliser des sites « vitrine », pour présenter par exemple son entreprise, mais sans aller plus loin. Ce type de site se fait de plus en plus rare aujourd'hui, car dès que l'on rajoute un élément d'interaction (comme un formulaire de contact), on ne parle plus de site statique mais de site dynamique

IV.3.3.2. Les sites web dynamiques :

Plus complexes, ils utilisent d'autres langages en plus de HTML et CSS, tels que PHP et MySQL. Le contenu de ces sites web est dit « dynamique » parce qu'il peut changer sans l'intervention du webmaster ! La plupart des sites web que vous visitez aujourd'hui, y compris le Site du Zéro, sont des sites dynamiques. Le seul pré requis pour apprendre à créer ce type de sites est de déjà savoir réaliser des sites statiques en HTML et CSS

VI.3.4. Les technologies de programmation web :

VI.3.4.1. Côté client :

- **HTML (L'Hypertext Markup Language) :** est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables tels que des applets. Les principales fonctionnalités du langage HTML sont :
 - Le support des tableaux.
 - Le support des formulaires de saisie.
 - La gestion des images maps (images cliquables).
 - L'insertion de scripts (Java Script, VB Script, CGI Script, ...).
- **XML (Extensible Markup Language) :** Tout comme HTML, le langage XML représente un langage de balises décrivant la structure et la nature des données d'un document. Mais la différence du langage HTML, statique, XML est davantage orienté traitement des données et publication des documents complexes.

VI.3.4.2. Côté serveur

- **JSP :** technologie développée par Sun Microsoft dans le but de créer des pages dynamiques avec le langage Java. Le JSP permet d'ajouter du code Java dans la page HTML qui sera interprétée par le serveur.
- **ASP :** technologie développée par Microsoft à partir de 1996 dans le but de créer des pages dynamiques. L'ASP permet d'ajouter du code dans les pages HTML qui sera interprété par le serveur. La partie ADO (ActiveX data object) de l'ASP permet de se connecter à une base de données.

- **Les servlets** : le nom vient d'une analogie possible à faire avec les applets. Il s'agit donc de programmes créés en Java et tournant sur le serveur web. L'exécution du programme génère les pages web renvoyées au client.

Conclusion

Dans ce chapitre, nous avons abordé les notions de base sur les nouvelles technologies de l'information et de la communication (Internet, Client-serveur, Le Web). Ces notions seront utilisées pour le développement de notre application. Dans le chapitre suivant, nous aborderons Les notion de base de la plate formeJ2EE.

Chapitre II

J2EE

Introduction :

La technologie Java est aujourd'hui incontournable, tant pour le développement d'applications autonomes que pour l'utilisation des outils orientés web. La technologie Java est ici employée afin de générer des pages HTML dynamiques (DHTML). Elles permettent l'intégration de l'ensemble des fonctionnalités de Java Enterprise Edition (XML, bases de données, services web, systèmes distribués CORBA, ...etc.), contrairement à d'autres langages web (tel que PHP par exemple).

Dans ce chapitre nous allons présenter l'essentiel des connaissances permettant de développer des applications web en s'appuyant sur JavaEE.

I. Un bref historique de J2EE [11]

L'environnement Java pour l'entreprise a commencé à émerger assez rapidement après les débuts de Java au milieu des années 90. A son origine, Java était destiné aux environnements contraints (par exemple des petits équipements électroniques). Il a en fait percé dans l'environnement du Web, notamment dans les navigateurs pour les supports d'interfaces graphiques riches la notion d'appliquette Java (en anglais applet). Les premières déclinaisons de Java dans l'environnement des serveurs sont apparues en 1997 avec les servlets, dont l'objectif est la construction programmatique de pages Web, puis avec les Enterprise Java Bean (EJB) dont l'objectif est le support de code métier (API) nécessitant un contexte d'exécution transactionnel.

Après ces premiers pas et un relatif succès de ces technologies, Sun a structuré l'offre technique autour des serveurs d'applications Java à travers le standard J2EE. L'objectif de ce dernier est de fédérer dans un cadre cohérent toutes les technologies nécessaires à la mise en œuvre des applications de l'entreprise (applications orientées <serveur>). Sun a décidé de rebaptiser son langage « Java 2 » et d'introduire la première version 1.2 des spécifications de J2EE en 1999. Cette version est suivie alors de la version 1.3 de J2EE en 2001, puis la version 1.4 en 2003 incluant un support complet des standards XML et le support des services Web. Mais pour des raisons de marketing, Sun a décidé de rebaptiser cette plateforme « Java Enterprise Edition » (en supprimant le 2) à compter de la version 1.5, qui elle-même été changée en « 5 » et elle est devenu JEE5.

II. Présentation de J2EE : [12]**Qu'est ce que J2EE ?**

J2EE (Java 2 Enterprise Edition) est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants.

On parle généralement de la «plate-forme J2EE» pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution. J2EE comprend notamment :

- Les spécifications du serveur d'application, c'est-à-dire de l'environnement d'exécution : J2EE définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Ces recommandations permettent ainsi à des entreprises tierces de développer des serveurs d'application conformes aux spécifications ainsi définies, sans avoir à redévelopper les principaux services.
- Des services, au travers d'API, c'est-à-dire des extensions Java indépendantes permettant d'offrir en standard un certain nombre de fonctionnalités. Sun fournit une implémentation minimale de ces API appelée J2EE SDK (J2EE Software Development Kit).

Dans la mesure où J2EE s'appuie entièrement sur le Java, il bénéficie des avantages et inconvénients de ce langage, en particulier une bonne portabilité et une maintenabilité du code.

De plus, l'architecture J2EE repose sur des composants distincts, interchangeables et distribués, ce qui signifie notamment :

- Qu'il est simple d'étendre l'architecture.
- Qu'un système reposant sur J2EE peut posséder des mécanismes de haute disponibilité, afin de garantir une bonne qualité de service.
- Que la maintenabilité des applications est facilitée.

III. L'architecture de J2EE: [15]

J2EE ajoute des nombreuses couches de niveau entreprise au-dessus de la plate-forme J2SE – Java Standard Edition. Chaque couche est conçue pour supporter une différente technologie de développement.

- **Technologie web application:** technologies liées à la production des interfaces web dynamiques, par exemple JSP (Java Server Pages) et servlet.
- **Technologie entreprise application:** technologies plus directement liées à la logique de business : EJB (Enterprise Java Bean), JavaMail, JMS (Java Message Service), JTA (Java Transaction), etc.
- **Technologie web services:** technologies utiles au développement des applications adhérentes au paradigme SOA (Service Oriented Architecture) : web services, JAX-WS (java API for XML-based web services), JAX-RPC (Java API for XML-Based RPC).
- **Technologie management and security:** technologies liées à la gestion de la technologie d'entreprise afin de réaliser l'accès et l'échange d'information entre machines et services distribués : JAAS (Java Authentication and Authorization Service), JCA (Java Connector Architecture)

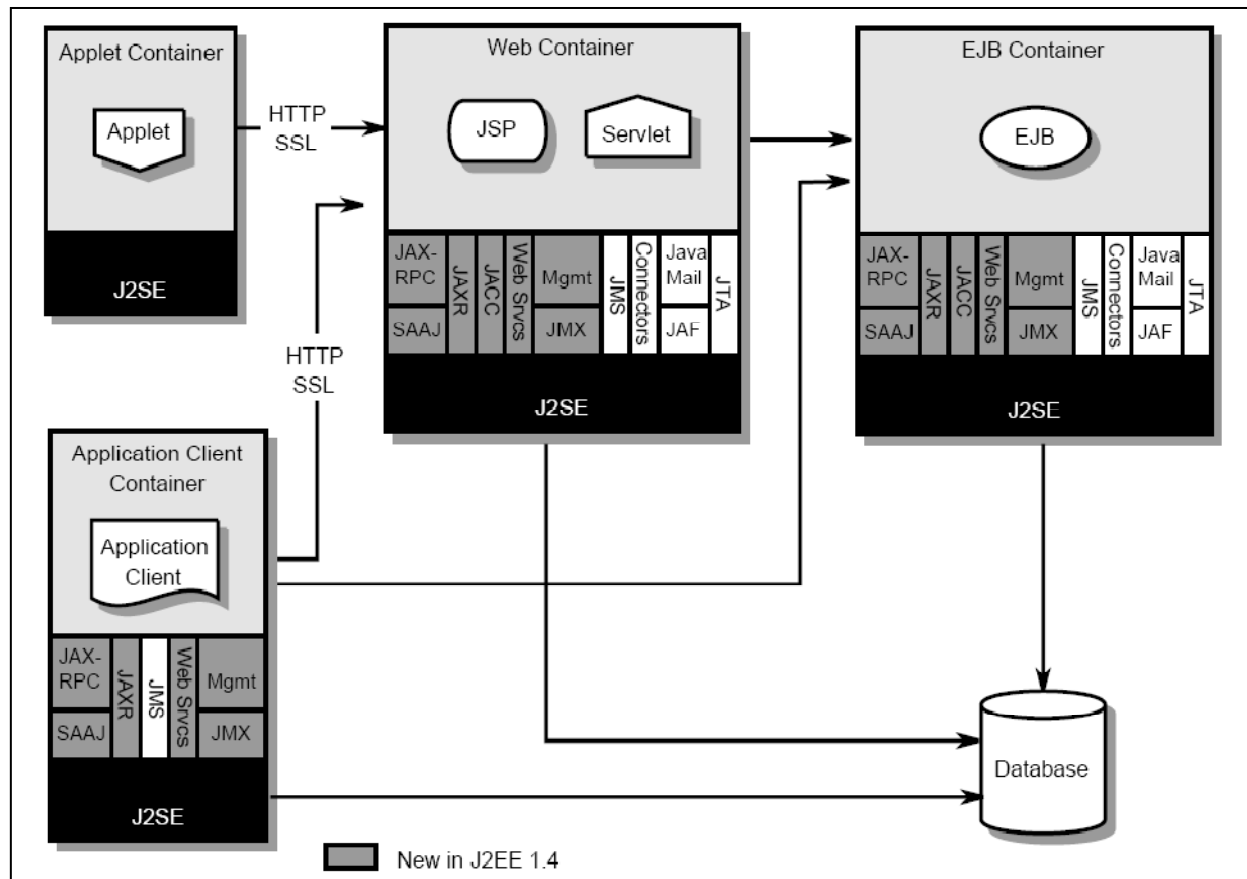


Figure (II.1) : L'architecture de J2EE

IV. Les couches logicielles JEE [11]

La plateforme JEE utilise un modèle d'application distribuée multi tiers pour des applications d'entreprise. La logique d'application est divisée en composants selon la fonction. Les composants divers qui composent l'application JEE sont installés sur des machines différentes en fonction du niveau de l'environnement JEE multi tiers auquel le composant d'application appartient, la figure suivante montre deux types d'applications JEE réparties dans des niveaux décrits comme suis :

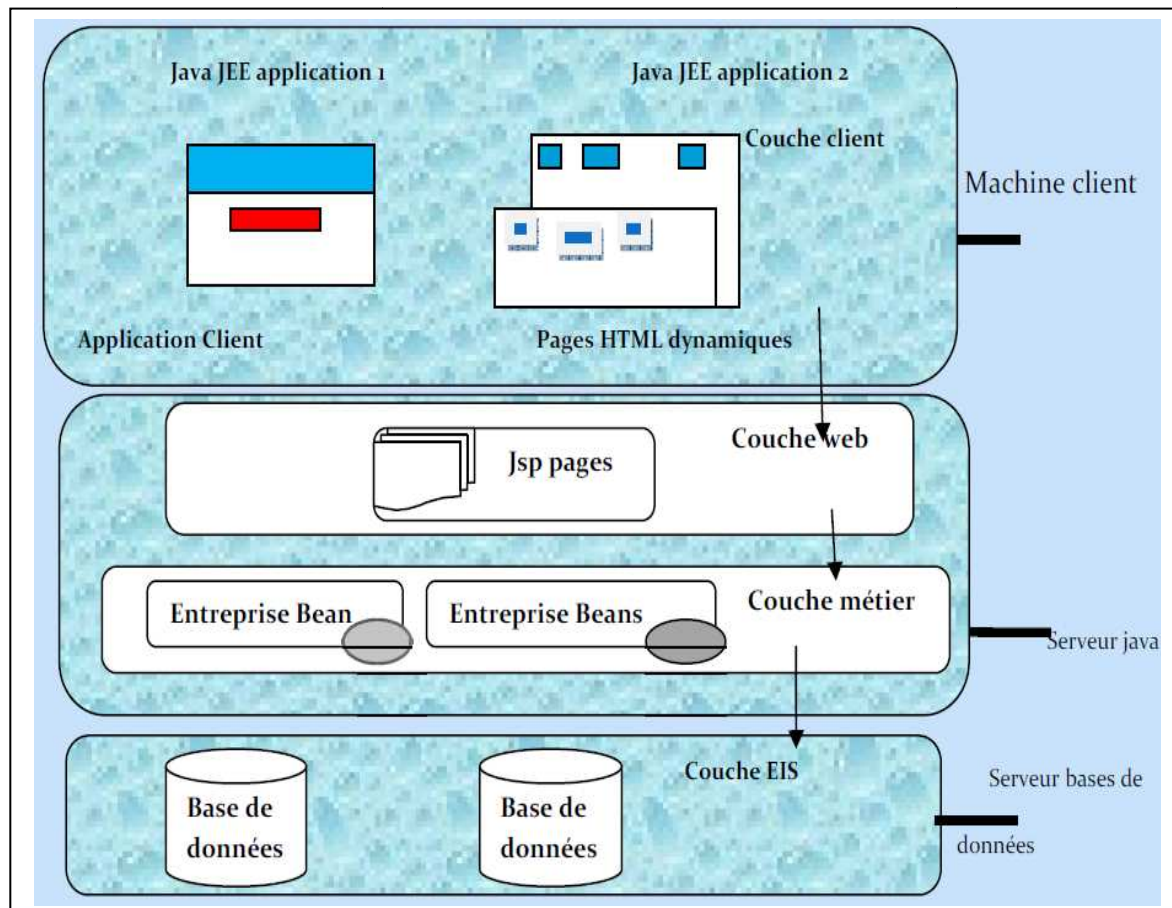
- La couche client : ces composants s'exécutent sur la machine du client.
- La couche WEB : ces composants s'exécutent sur le serveur JEE.
- La couche métier : ces composants s'exécutent sur serveur JEE.
- La couche EIS (Entreprise information System) : s'exécutent sur le serveur EIS.

Bien qu'une application JEE peut être constituée par trois ou quatre couches comme montrées dans Figure 2 suivante. Les applications JEE multi tiers sont généralement considérées comme des applications 3-tiers, parce qu'elles sont distribuées sur trois emplacements :

- La machine client.

- La machine pour serveur JEE.
- Et la machine pour serveur de base de données.

Les applications à 3-tiers qui fonctionnent de cette façon étendent le standard 2-tiers : client/serveur en plaçant un serveur d'application entre l'application de client et le stockage principal.



Figure(II.2) : Architecteur trois tiers

V. Spécifications de la plate forme JEE :

JEE comporte des spécifications qu'on peut regrouper en deux grandes parties :

- + Les APIs ;
- + Les serveurs d'applications.

V.1. Les APIs.

Les APIs sont des extensions Java, forment une collection d'outils logiciels permettant d'offrir en standard un certain nombre de fonctionnalités, Sun fournit une implémentation minimale de ces APIs nommée J2EE SDK (JEE SOFTWARE DEVELOPMENT KIT). On peut regrouper ces APIs en trois groupes principales :

- Composants ; [14]
- Services d'infrastructures ; [12]
- Services de communications. [12]

V.1.1. Les composants :

Un composant est un fichier simple, contenant généralement du code compilé. Il est possible de créer des composants avec la majorité des langages. Aussi il peut être vu comme unité qui regroupe un certain nombre de fonctionnalités qui peuvent être appelées depuis un programme externe ou client.

Remarque: Pour les composants JEE écrite en java et compilées sont assemblés en une application JEE déployés, exécutés et gérés par le serveur JEE.

On peut faire une séparation entre les composants existants selon leur exécution :

- Composants exécutés sur le client (clients et applets).
- Composants exécutés sur le serveur (web et les composants métiers).

1. Composants exécutés sur le client (clients et applets).

Peuvent être application client ou applet.

❖ Client JEE

Un client JEE peut être client web ou application client.

○ Client web

Un client web est appelé client léger qui n'a généralement pas de requête de BDD. L'utilisation de client web permet de charger les opérations lourds au niveau de l'entreprise beans et sont exécutées sur le serveur JEE où ils peuvent exploiter les avantages des technologies JEE (sécurité, la vitesse, les services ...) sur le serveur. Un client web a besoin de deux fonctionnalités principales :

- Les pages web dynamiques avec les différents langages à balisage (HTML, XML,), ces pages sont générés par les composants web au niveau du client web.
- Le navigateur web qui affiche les pages reçus par le serveur.

○ Application client

L'application client fonctionne au niveau de la couche métier, et peut accéder directement à l'entreprise Bean. Généralement les utilisateurs possèdent une interface graphique (GUI) pour exécuter l'application client (on utilisant l'API Swing ou Abstract window toolkit).

❖ Applet

Une applet est une petite application java qui a les caractéristiques de pouvoir s'exécuter via un navigateur .en effet, la plupart des navigateurs aujourd'hui, dispose d'une machine virtuelle capable d'interpréter les bytecode inclus dans une page HTML.

Il existe plusieurs différences entre application et applets .en effet, les applets fournissent un certain nombre de fonctionnalités supplémentaires propres à leur contexte d'utilisation, comme la possibilité de communiquer avec le navigateur ou même entre applets. Mais la plus notable, réside dans la sécurité qui leur est associée.

2. Composants exécutés sur le serveur [13]

Sont des composants web ou bien EJB.

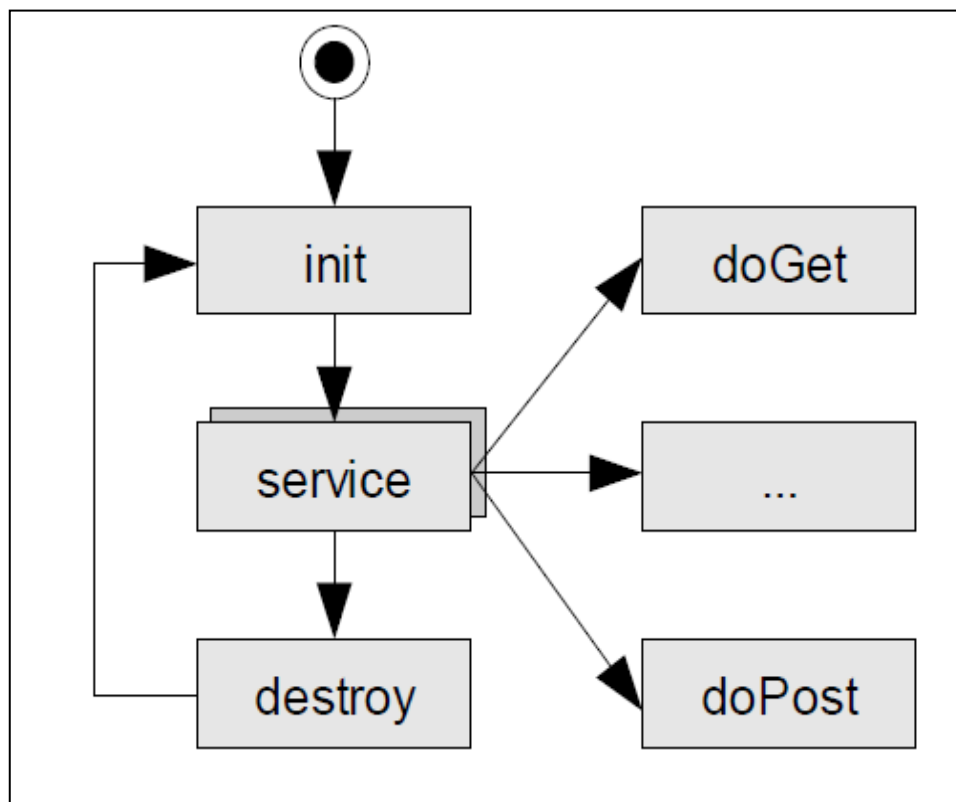
❖ **Composants WEB** : Sont des servlets, JSP ou JSF.

- **Servlet** Ou « un peu d'HTML dans beaucoup de Java... »:

Une servlet est un composant du coté serveur, considérée comme une classe java qui permet une communication avec un protocole « requête / réponse ». Autrement dit, une servlet est un programme qui s'exécute coté serveur en tant qu'une extension du serveur. Elle reçoit une requête du client, elle effectue des traitements et renvoie le résultat.

L'utilisation des servlets permet la génération des pages web dynamiques qui utilisent le protocole HTTP, ou aussi n'importe quel autre protocole reposant sur le principe de servlet. Une servlet est une classe java qui implémente les méthodes dopost et/ou doget .

✓ **Cycle de vie d'une servlet :**



Figure(II.3) : Cycle de vie d'une servlet

- **init()** : initialisation de la servlet chargement du code.
Souvent effectué lors de la première requête cliente (doGet, doPost) Allocation d'un pool de threads
- **doGet ()** : Traitement des requêtes HTTP GET
- **doPut ()** : Traitement des requêtes HTTP PUT
- **doPost ()** : Traitement des requêtes HTTP POST
- **destroy ()** : Destruction de la servlet par le serveur

- La méthode Init est appelé une seule fois au chargement de la servlet.
- La méthode service est appelée à chaque requête http (client)
- La méthode destroy est appelée une seule fois lorsque le conteneur détruit la servlet.

✓ **Avantage et inconvénients des servlets :**

➤ **Avantage :**

- Portabilité (car s'exécute sur une JVM (java vertuel machin)).
- S'exécute cote serveur.
- Maintenabilité.
- Elle maintient des sessions de plusieurs clients http grâce à l'objet http session.
- Multithread/contrairement a PHP exécute plusieurs fois.

➤ **Inconvénients :**

- Très difficile d'utiliser les outils d'édition HTML pour traiter les aspects graphique.

• **JSP : Java Server Page** Ou « *un peu de Java dans beaucoup d'HTML...* »

Les JSPs (pour Java Server Pages) est une technologie qui permet d'écrire très facilement des pages dynamiques en insérant des portions de code java dans une page html. Destiné à l'origine à la présentation des données, une page JSP permet de définir le design d'une page (en HTML) et d'y intégrer les données (grâce aux portions de code java).cette technologie favorise donc la séparation de la présentation et du contenu.

D'un point de vu plus technique, une page java est donc constitue de code HTML dans lequel sont insérées des code java délimitées par des balises particulières .l'intégralité du code est portable puisque tout comme un code java, il sera compilé et exécuté comme Servlet .

A la différence d'une Servlet, il sera par contre compilé au vol (lors de chaque invocation par un moteur de JSP. Enfin, les pages JSP offrent les possibilités d'importer des classes java et d'accéder à des objets java.

- **JSF [14]**

(**Java Server Faces**) Un cadre standard de composants conçu pour construire des interfaces utilisateur riches pour les applications Java. Java Server Faces exécuté sur le serveur, mais sont affichés sur le client. La technologie JSF représente un Framework basé sur les composants utilisateurs utilisés pour construire des applications web. L'API de JSF permet de représenter les composants, de gérer leurs états et leurs comportement. Elle facilite l'écriture d'interfaces utilisateurs en fournissant une bibliothèque de contrôles

- Simples : zones de saisie classiques, boutons, liens hypertextes.
- Complexes : tableaux de données ...
- Elle permet même de créer ses propres composants, ou d'enrichir les composants existants.

JSF permet principalement de:

- Faciliter le développement de pages web grâce à un Framework basé sur les composants. représenter les composantes UI et gérer leurs états.
- Gestion d'événements.
- Validation coté serveur.
- Conversion de données.
- Définition de la navigation des pages.
- gérer les composantes et les lier aux objets coté serveur.

- ❖ **Composants métiers (EJB) [13]**

Enterprise JavaBeans (EJB) est une architecture de composants logiciels côté serveur pour la plateforme de développement JEE.

Cette architecture propose un cadre pour créer des composants distribués (c'est-à-dire déployés sur des serveurs distants) écrit en langage de programmation Java hébergés au sein d'un serveur applicatif permettant de représenter des données (EJB dit *entité*), de proposer des services avec ou sans conservation d'état entre les appels (EJB dit *session*), ou encore d'accomplir des tâches de manière asynchrone (EJB dit *message*). Tous les EJB peuvent évoluer dans un contexte transactionnel.

L'architecture EJB prévoit en effet de manipuler les composants EJB dans un environnement distribué, assurant une complète compatibilité avec tous types de clients : servlet, page jsp, autre EJB, applet ...

Pour assurer la communication réseau entre différents composants distribués les EJB utilisent un mécanisme basé sur RMI. ce mécanisme est implémenté dans une API qui offre à une machine virtuelle java les mécanismes nécessaires pour invoquer des méthodes sur d'autre JVM, qu'elles soient ou non sur le même réseau.

- **Les différents types d'EJB :**

Il existe trois types des EJB :

- **Les beans entités**

Ça correspond à des objets réels tels que ces objets sont généralement des enregistrements persistants stockés dans une BDD. On peut aussi dire que les beans entités sont des moyens de modélisation des concepts métier pouvant être désignés par des noms, par exemple : un bean entité peut représenter un client ou un objet stocké.

- **Les beans session**

Ceux sont une extension de l'application client et ils sont responsables de la gestion des processus ou des tâches.

- **Les beans orientés message**

Un bean session fournit une interface distante afin de définir les méthodes que l'on peut invoquer, alors que bean orienté message attend des messages asynchrones spécifiques auquel il répond.

Alors généralement, un EJB récupère également les données stockées, les traitent et les envois au programme du client.

V.1.2. Services d'infrastructures [12]

On trouve les services suivants :

- **JDBC « Java DataBase Connectivity »**

C'est une API qui permet aux programmes java d'interagir avec les bases de données SQL. Les serveurs fournissent en plus un mécanisme qui permet de réutiliser les connexions créées avec les utilisations.

- **JTA/JTS « Java Transaction Api / Java Transaction Services »**

C'est une API définissant des interfaces standard avec un gestionnaire de transaction, c'est_à dire permettant à des applications et à des serveurs JEE d'accéder à des transactions.

- **JCA « JEE Connector Architecture »**

C'est une API de connexion au système d'information de l'entreprise, notamment aux systèmes dits "legacy" tel que les ERP (Entreprise Ressource Planning).

- **JMX « Java Management Extension »**

Permet de développer des applications WEB de supervision d'applications à partir des extensions.

- **JNDI « Java Naming and Directory Interface»**

C'est une extension java standard qui fournit une API permettant de localiser et d'utiliser des ressources; il peut y avoir un appel à des services CORBA, DNS, NIS, LDAP...

V.1.3. Services de communication :

- **JMS « Java Message Service »**

A pour rôle l'envoi et la réception des messages entre deux composants d'application JEE.

- **JAAS « Java Authentication And Autorisation Service »**

Permet aux applications JEE d'authentifier et autoriser l'exécution par les utilisateurs.

- **Java Mail TM Technology**

Permet l'envoi des e-mails d'une manière indépendante de la plate forme et du protocole, donc envoi de courrier électronique.

- **RMI-IIOP « Remote Method Invocation/Internet Inter-Orb Protocol »**

Cette API permet la communication synchrone entre objets.

- **Web Services**

Les web services permettent de partager un ensemble de méthodes qui pourront être appelées à distance. Cette technologie utilise XML ce qui permet d'être utilisé par n'importe quel langage et n'importe quelle plate forme.

- **XML**

N'est pas vraiment une API de JEE mais son utilisation dans cette plate forme est de plus en plus important comme son utilisation pour les fichiers de configurations. XML est la base d'un nouveau mode de communication entre les applications.



Les conteneurs (en anglais containers) :

Un conteneur est une entité qui fournit aux composants la gestion de cycle de vie, la sécurité, les services de déploiement et d'exécution. Chaque type de conteneur (EJB, WEB, JSP...) fournit aussi des services spécifiques à ces composants.

Il existe plusieurs types de conteneurs qu'on peut citer :

- **Conteneur web**

Il est composé de deux grandes parties : un moteur de servlets et un moteur de JSP, donc ce conteneur permet d'exécuter les composants JSP et les servlets de l'application JEE.

- **Conteneur EJB**

Ce type de conteneur permet de gérer l'exécution d'EJB.

- **Conteneur d'application client**

Il permet de gérer l'exécution des composants de l'application client. Il s'exécute sur le client.

- **Conteneur d'applets**

C'est une composition de deux parties : navigateur web et java plug-in qui fonctionnent ensemble sur le client, donc ce conteneur permet de gérer l'exécution des applets.

V.2. Les serveurs d'applications. [15]

Les API J2EE ont été projetés pour travailler avec un particulier type de serveur appelé J2EE Application Server. Un serveur d'application est une couche software résidente sur une machine serveur qui fournit les services que la technologie J2EE nécessite. Il y a plusieurs serveurs d'applications. Parmi les produits commerciaux on rappelle **Bea WebLogic**, **IBM Websphere**, **Sun Application Server**, ...etc. Parmi les produits libres le plus connu est **JBOSS**. Les différences principales entre les différents serveurs d'applications sont relatives aux opérations de **déploy**, **clustering**, ...etc. Par contre toutes les fonctionnalités qui concernent strictement le fonctionnement des applications J2EE adhèrent aux spécifications proposées par la Sun.

Un serveur d'application s'installe et se lance comme un normal serveur web (de fait il met à disposition des services typiques d'un serveur web). En plus il possède un panneau d'administration accessible par un poste distant à travers lequel on peut définir les différents services. On peut considérer le serveur d'application comme une application Java standalone exécutée par une J2SE qui dessert les requêtes provenant des différents clients.

Les serveurs d'application se sont développés depuis la création de J2EE. On peut distinguer principalement 2 grandes catégories de serveurs :

- **Open Source** : évolue grâce à la communauté par exemples **Tomcat : Apache**, **Jonas : ObjectWeb**, **JBoss : JBoss ...**
- **Propriétaire** : évolue selon l'éditeur par exemples **WebSphere : IBM**, **WebLogic : BEA**, **WebObject : Apple**, **Oracle Application Server : Oracle ...**

V.2.1. Le rôle d'un serveur d'application (ou bien les services) : [11]

Le rôle d'un serveur d'application est d'héberger des applications distribuées, fabriquées à base de composants Java (Servlet, JSP, EJB) et de les rendre accessibles à des clients web (navigateurs) et à des applications d'entreprise écrites en Java. Le serveur d'application doit être performant et fiable pour satisfaire aux exigences des applications d'entreprise. Le serveur offre différents services utiles pour les applications :

a. Service de nommage :

C'est le service le plus important dans un serveur d'application implémentant l'interface SPI (Service Provider Interface) et utilise généralement le protocole LDAP (Lightweight Directory Access Protocol). Ce dernier fournit des mécanismes de recherche, de localisation et de liaison.

b. Services de gestion de transaction :

Il permet de gérer des transactions locales mais surtout il prend en charge la gestion des transactions distribuées, indispensable pour les applications JEE.

c. Service de gestion de la disponibilité des applications :

Il permet de gérer un grand nombre d'accès et de requêtes simultanés.

d. Service de sécurité :

L'administrateur du serveur d'application peut configurer des domaines de sécurité en définissant des utilisateurs (nom, mot de passe), des groupes d'utilisateurs, puis les permissions accordées sur différents éléments par rapport aux composants et/ou aux ressources de l'application JEE.

e. Service d'administration :

L'administrateur a la possibilité de consulter, de modifier, et d'ajouter des éléments à la configuration des applications d'entreprises et du serveur d'application.

f. Service d'accès aux données :

Ce service est une implémentation des extensions de L'API JDBC

g. Service de gestion de messages :

C'est une implémentation de l'API JMS, il permet de gérer une file de messages de traitements asynchrones.

VI. Les avantages d'utiliser JEE :

L'utilisation de JEE pour développer et exécuter une application représente plusieurs avantages :

- Une architecture d'application basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement.
- La possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA...
- La possibilité de choisir les outils de développement et les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.

Conclusion

Dans ce chapitre, nous avons abordé les notions de base de la plate forme J2EE, certaines d'entre eux seront utilisées pour le développement de notre application. Nous allons présenter dans le chapitre suivant les détails de conception de notre application en utilisant le langage de modélisation UML.

Chapitre IV

Analyse et conception

Introduction :

Dans le but d'une meilleure organisation et une bonne maîtrise du travail, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou démarche bien définie.

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.

Pour ce faire, notre démarche va s'appuyer sur le langage UML, conçu pour la visualisation, la spécification et la construction des systèmes logiciels.

I. Présentation d'UML : [16]**I. 1. Définition :**

UML (*Unified Modeling Language*) est un langage de modélisation unifié (ou universel) et non pas une méthode. UML a été conçu pour permettre la modélisation de tous les phénomènes de l'activité de l'entreprise (processus métier, systèmes d'informations, systèmes informatiques, composants logiciels...) indépendamment des techniques d'implémentations (langage de programmation...) employés par la suite.

I. 2. Les diagrammes UML : [17]

Le langage UML est décrit par un méta-modèle, c'est-à dire une représentation simple, mais rigoureuse, de ses constituants élémentaires, avec leurs règles d'utilisations et de syntaxe. Un diagramme est défini par les auteurs d'UML comme la représentation graphique d'un ensemble sélectionné de constituants UML. Ils ont élaboré treize diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel.

Une classification courante distingue les diagrammes qui traduisent la structure et les diagrammes qui présentent le comportement du système étudié.

Parmi ces diagrammes, on a choisit les suivants qui sont suffisant pour étudier et analyser la situation existante:

- ✓ Le diagramme de cas d'utilisation.
- ✓ Le diagramme de classes.
- ✓ Le diagramme de séquence.
- ✓ Le diagramme d'activité.

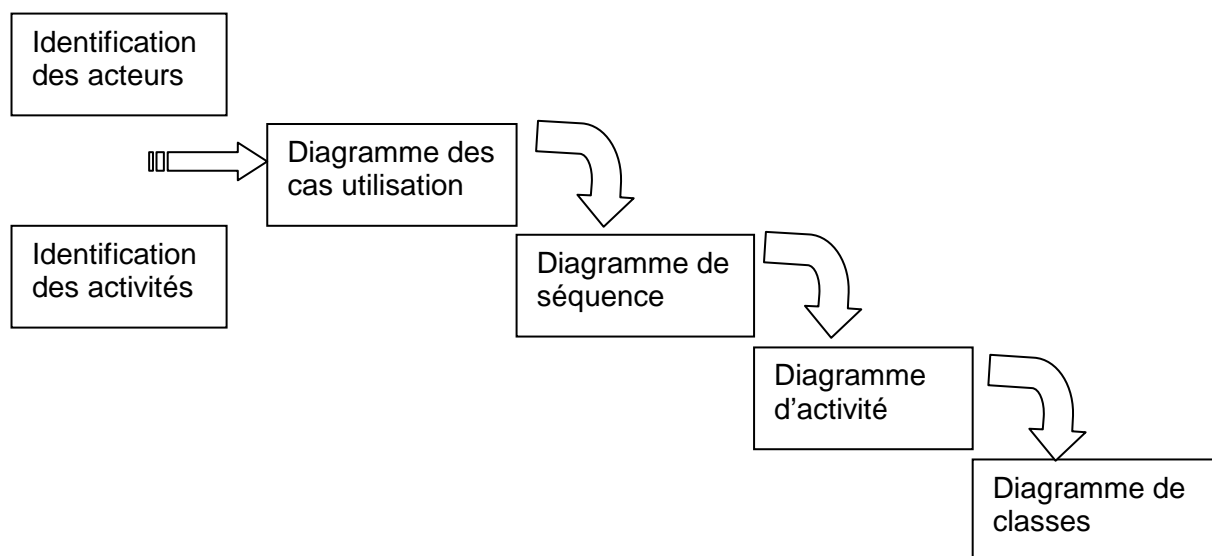
II. Analyse

L'objectif de cette partie est la spécification de manière claire et concise de l'application. On va de ce fait procéder à la définition des utilisateurs, ainsi que leurs besoins respectifs auxquels l'application est tenue d'apporter des réponses.

II.1. Identification des besoins :

Notre travail a pour objectif de développer une application qui permettra de faciliter les fonctions de la scolarité que ce soit pour les administrateurs de la scolarité, les enseignants et les étudiants.

Dans le but d'une meilleure organisation et une bonne maîtrise du travail, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou une démarche bien définie. La figure ci-dessous montre la représentation graphique de la démarche de modélisation choisie pour concevoir notre application :



(Figure III.1): La démarche de modélisation de l'application

II.2. Identification des acteurs :

Définition :

Un acteur représente le rôle d'une entité externe (utilisateur humain ou non) interagissant avec le système. Il n'appartient pas au système mais il fournit de l'information en entrée et/ou la reçoit en sortie.

Nous avons identifié les quatre types d'acteurs suivants :

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 de département.
3. **L'Agent de la scolarité** : c'est un rôle qui peut être joué par tous les personnes chargée des services de la scolarité
4. **Le Chef de département** : c'est un rôle qui peut être joué par la personne qui s'occupe de la direction du département.
5. **Administrateur** : c'est un rôle qui peut être joué par la personne qui est chargée de l'administration de la base de données.

II.3. Identification des cas d'utilisation :

Définition [18]:

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

Acteur	Les tâches
Etudiant	T1 : Se connecter au site T2 : S'authentifier T3 : Consulter (imprimer) le relevé de notes. T4 : Consulter les PV globaux de délibérations. T5 : Changer son mot de passe.
Enseignant	T1, T6: S'authentifier T7: Gérer des notes des étudiants T8 : Consulter les PV globaux de délibérations. T9 : Changer son mot de passe.
Agent de la scolarité	T1, T10 : S'authentifier. T11 : Editer les listes des étudiants. T12 : Editer les relevés de notes et les certificats de scolarité. T13 : Editer les PV globaux de délibérations. T14: Changer son mot de passe.
Chef de département	T1, T15 : S'authentifier. T16 : Affecter les enseignants. T17: Gérer les enseignants (ajouter, supprimer) T18: Gérer les modules (ajouter, supprimer) T19 : Consultation des PV de délibérations. T20 : Changer son mot de passe.
Administrateur	T1, T21: S'authentifier T22: Gérer les comptes des utilisateurs T23: Changer de mot de passe

Tableau (III .1) : Les différentes tâches des acteurs

II.4 Diagramme de contexte : [18]

Le diagramme de contexte est un modèle conceptuel qui permet d'avoir une vision globale des interactions entre le système et les liens avec l'environnement extérieur. Il permet aussi de délimiter le champ d'étude.

Dans notre cas le diagramme de contexte est donné par la figure suivante :

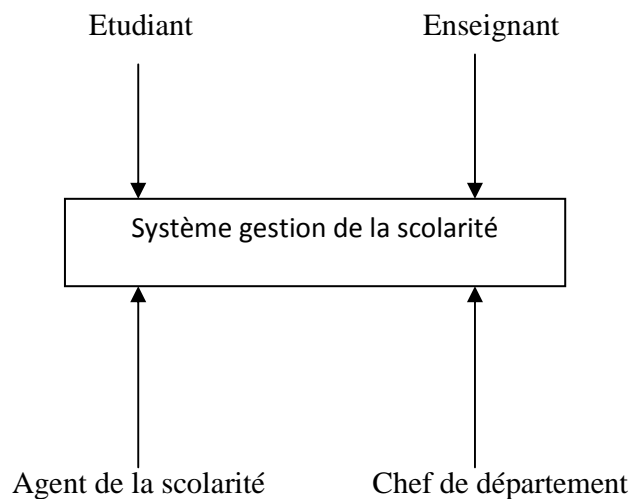


Figure (III.2) : Diagramme de contexte de l'application

II.5. spécification des scénarios :

Chacun des acteurs qu'on a définis effectue des tâches et chaque tâche est décrite par des scénarios qu'on résume comme suit :

Acteur	Les tâches	Scenarios
Etudiant	T1 : se connecter au site	S0 :Saisir l'URL du site
	T2 : S'authentifier	S0 : S1 : Cliquer sur le lien « Etudiant » S2 : Remplir le formulaire et valider
	T3: Consulter (imprimer) le relevé de notes	S0, S1, S2 S3 : Cliquer sur le lien « Relevé de notes »
	T4 : Consulter les PV modulaire	S0, S1, S3 S4 : Cliquer sur le lien « PV de délibérations » S5 : Choisir la section et le cycle puis valider
	T5 : Changer son mot de passe	S0, S1, S2 S6 : Cliquer sur le lien «changement de mot de passe» S7 : Remplir le formulaire puis valider

Enseignant	T1, T6: S'authentifier	S0 S8 : Cliquer sur le lien « Enseignant » S9 : Remplir le formulaire et valider
	T7: Gérer les notes des étudiants	S0, S8, S9, S10 : sur le menu «gérer les notes» S11 : Choisir l'ajout, la suppression ou la modification S12 : remplir le formulaire puis valider
	T8 : Consulter les PV modulaire	S0, S8, S9, S13: Cliquer sur le lien « PV modulaire» S14: Choisir la section et le cycle puis valider
	T9 : Changer son mot de passe.	S0, S8, S9, S15 : Cliquer sur le lien «changement de mot de passe» S16 : Remplir le formulaire puis valider
Agent de la scolarité	T1, T10: S'authentifier	S0, S17 : Cliquer sur le lien « Agent de scolarité » S18 : Remplir le formulaire et valider
	T11 : Editer les listes des étudiants.	S0, S17, S18, S19 : Sur le menu « Gestion des étudiants» S20 : faire le choix (ajouter, consulter, supprimer) S21 : remplir le formulaire et valider
	T12 : Editer les relevés de notes	S0, S17, S18, S22 : Cliquer sur le lien « Relevés de notes» S23 : Remplir le formulaire et valider
	T 13: Editer les PV modulaire	S0, S17, S18, S24 : Cliquer sur le lien «Edition de PV modulaire»
	T14 : Modifier leur mot de passe	S0, S17, S18, S25 : Cliquer sur le lien «Changement de mot de passe» S26 : Remplir le formulaire et valider

Chef de département	T1, T2 T15 : s'authentifier	S0 S27 : Cliquer sur le lien « Chef de département » S28 : Remplir le formulaire et valide
	T16 : Affecter les enseignants.	S0, S27, S28, S29 : Cliquer sur le lien «Affecter les enseignants» S30 : Remplir le formulaire et valider
	T20 : gérer les enseignants (ajouter, consulter)	S0, S27, S28, S 31 : Sur le menu «gestion des enseignants » fait le Choix (ajouter, consulter, supprimer) S32 : Remplir le formulaire et valider
	T21 : Gérer les modules (ajouter, supprimer)	S0, S27, S28, S 33 : Sur le menu «gestion des modules» fait le Choix (ajouter, consulter, supprimer) S34 : Remplir le formulaire et valider
	T22 : Consultation des PV de délibérations.	S0, S27, S28, S35 : Cliquer sur le lien « PV globaux de délibération »
	T23 : Modifier leur mot de passe	S0, S27, S28 S36 : Cliquer sur le lien «Changement de mot de passé» S37 : Remplir le formulaire et valider
Administrateur	T1, T25 : s'authentifier	S0 S38: Cliquer sur le lien « Administrateur » S39 : Remplir le formulaire et valide
	T25: Gérer les comptes des utilisateurs	S0, S38, S39, S40 : Sur le menu « gestion des comptes »faire le choix (consulter, ajouter, supprimer) S41 : Remplir le formulaire et valider
	T26 : Modifier son mot de passe	S0, S38, S39, S42 : Cliquer sur le lien «Changement de mot de passé » S43 : Remplir le formulaire et valider

Tableau (III .2) : Les Scenarios des acteurs

II .6. Spécification de quelques 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 uses cases décrit les objectifs (le but) du système

✚ Etudiant :

❖ Cas d'utilisation: « Authentification »

Use Case : Authentification

Rôle: étudiant.

Scenarios: S0, S1, S2

Description:

- 1 : L'étudiant clique sur le lien « étudiant »
- 2 : Le système affiche le formulaire d'identification
- 3 : L'étudiant remplit le formulaire et valide
- 4 : le système affiche le message d'erreur si le formulaire est mal rempli sinon
- 5 : Le système affiche l'espace étudiant.

❖ Cas d'utilisation: « Consulter les PV modulaire»

Use Case: Consulter les PV modulaire

Rôle: étudiant.

Scenarios: S0, S1, S2, S4, S5

Description:

- 1 : L'étudiant clique sur le lien « Consulter PV »
- 2 : Le système affiche le formulaire pour faire le choix (la section et le cycle)
- 3 : l'étudiant remplit le formulaire et valider
- 4 : Le système affiche le PV de délibération.

✚ Enseignant.

❖ **Cas d'utilisation:** « Gérer les notes des étudiants »

Use Case: gérer les notes des étudiants (ajouter notes)

Rôle: enseignant.

Scenarios: S₀, S₈, S₉, S₁₀, S₁₁, S₁₂

Description:

- 1: L'enseignant clique sur le lien «ajouter des Notes »
- 2 : Le système affiche le formulaire
- 3 : L'enseignant fait choisit la section, le cycle, le module et valide
- 4 : Le système affiche une liste des étudiants.
- 5 : L'enseignant saisit la note de chaque étudiant et valide
- 6 : Le système affiche le message succès.

❖ **Cas d'utilisation:** « Changer son mot de passe»


Use Case: Changer son mot de passe

Rôle: Enseignant.

Scenarios: S₀, S₈, S₉, S₁₅, S₁₆

Description:

- 1: L'enseignant clique sur le lien « Changement de mot de passe »
- 2 : Le système affiche le formulaire de modification
- 3 : L'enseignant remplit le formulaire et valide
- 4 : Le système affiche le message d'erreur si il y a une erreur sinon
- 5 : Le système affiche le message de succès

 Chef de département.

❖ **Cas d'utilisation:** « gérer les modules (ajouter, supprimer) »

Use Case: gérer les modules

Rôle: chef de département.

Scenarios: S0, S27, S28, S33, S34

Description:

1 : Le chef de département clique sur le lien «Gestion des modules »et fait le choix (ajouter, supprimer) un module

2 : Le système affiche le formulaire

3 : Le chef de département remplit le formulaire et valide

4 : Le système affiche le message de succès.

 Agent de la scolarité.

❖ **Cas d'utilisation :** « Editer les relevés de notes »

Use Case: Editer les relevés de notes

Rôle: Agent de scolarité.

Scenarios: S0, S17, S18, S22, S23

Description:

1: L'agent clique sur le lien «relevés de notes »

2 : Le système affiche le formulaire à remplir

3 : L'agent remplit le formulaire et valider

4 : Le système affiche la liste des étudiants

5 : L'agent choisit l'étudiant puis valider

6 : Le système affiche le relevé de notes demandé.

Administrateur :

❖ **Cas d'utilisation** : « Gérer les comptes (créer un compte) »

Use Case: Gérer les comptes des utilisateurs (créer un compte)

Rôle: Administrateur.

Scenarios: S₀, S₃₈, S₃₉, S₄₀, S₄₁

Description:

1 : L'administrateur clique sur le lien « Gestion des comptes » et fait son choix (ajouter, supprimer, consulter) un compte

2 : Le système affiche le formulaire à remplir

3 : L'administrateur remplit le formulaire et valide

4 : Le système affiche le message de succès

II.6. Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel.

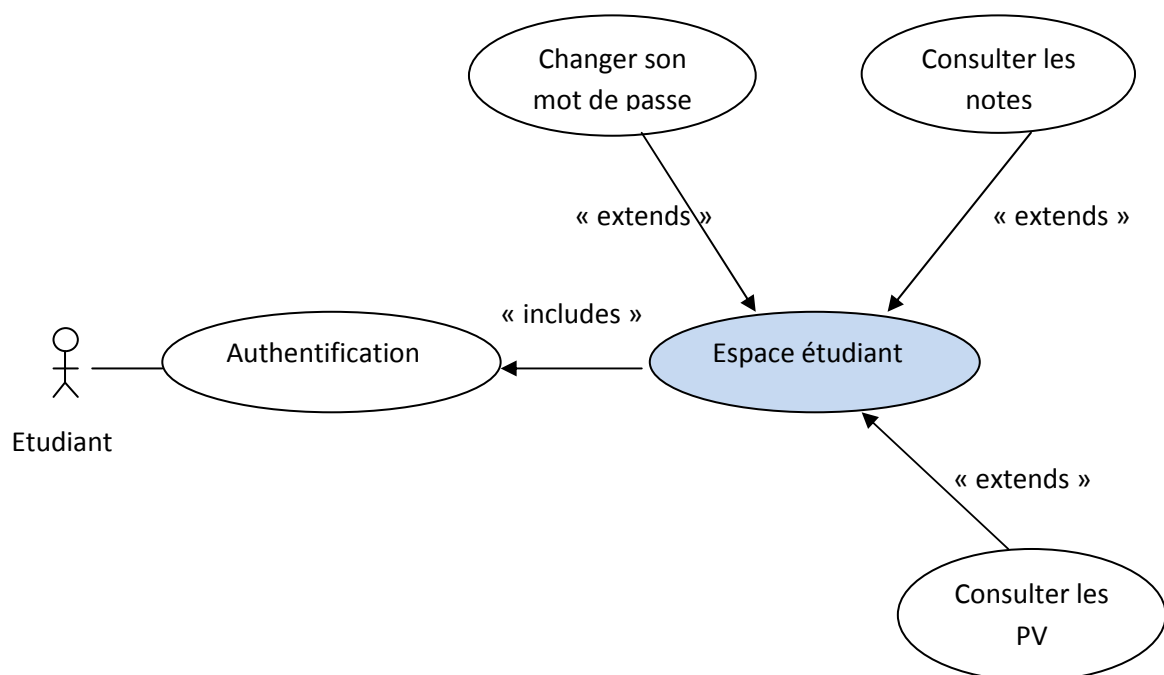
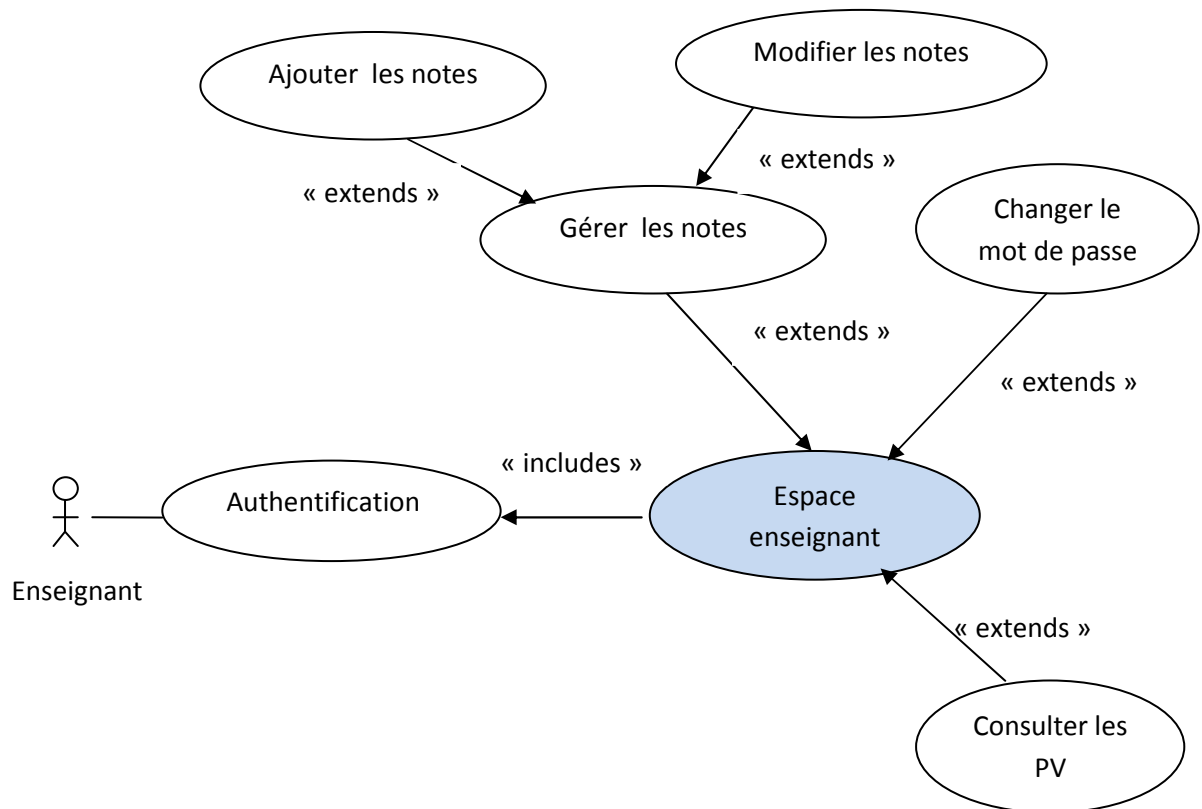


Figure (III.3) : Diagramme global de cas d'utilisation relatif à l'étudiant



Figure(III.4) : Diagramme global de cas d'utilisation relatif à l'enseignant

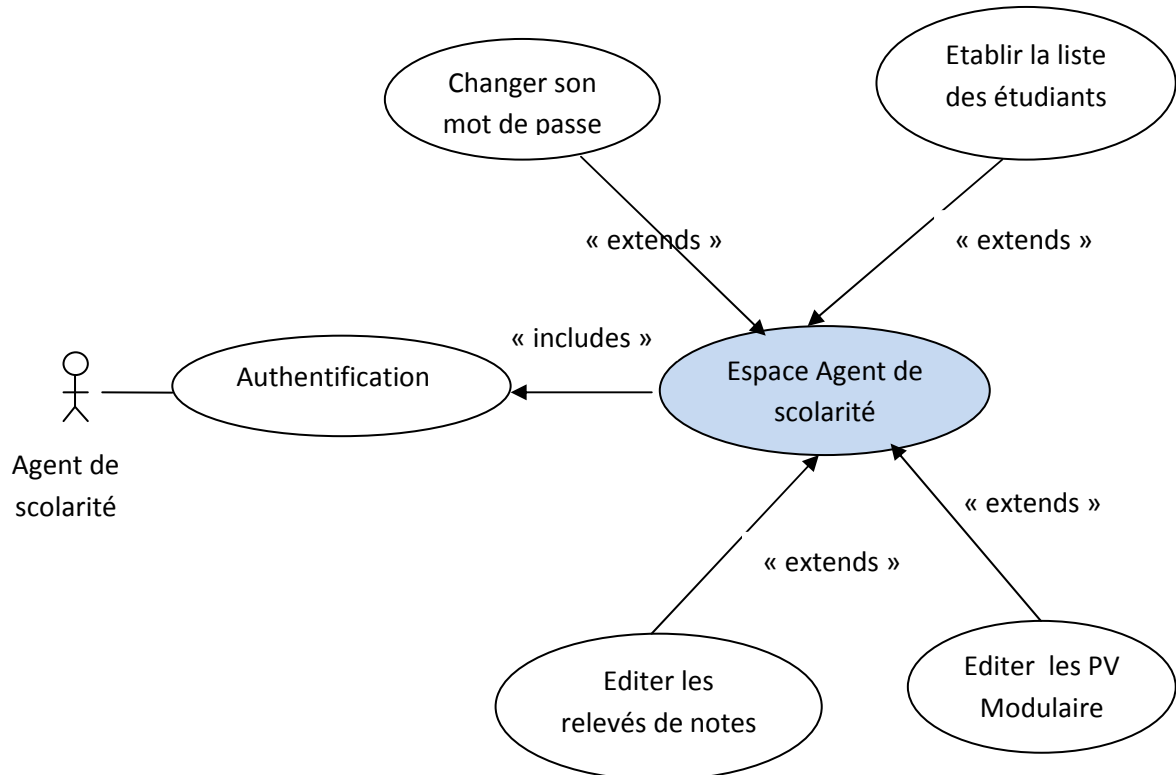
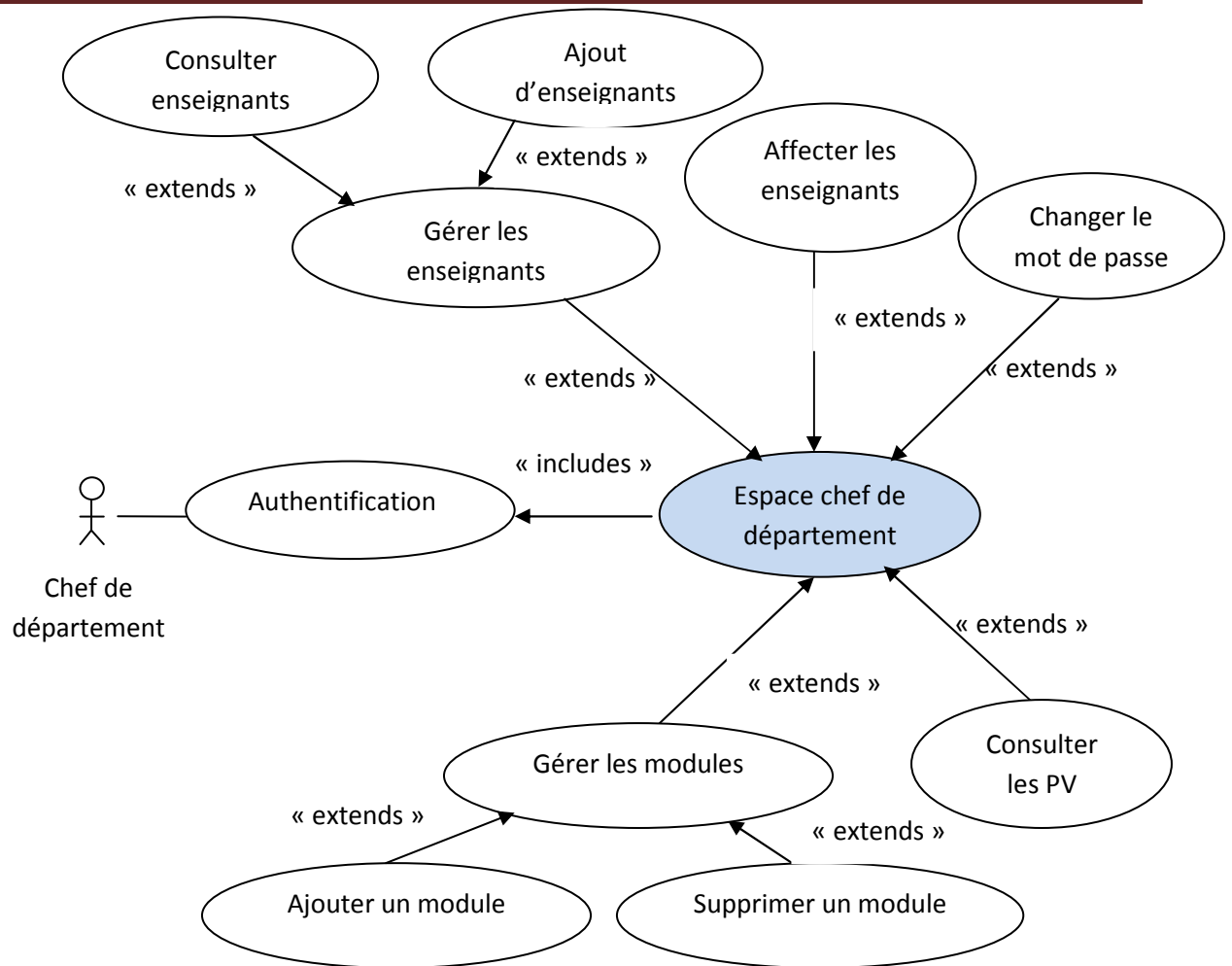


Figure (III.5) : Diagramme global de cas d'utilisation relatif à l'Agent de scolarité



Figure(III.6) : Diagramme global de cas d'utilisation relatif au chef de département

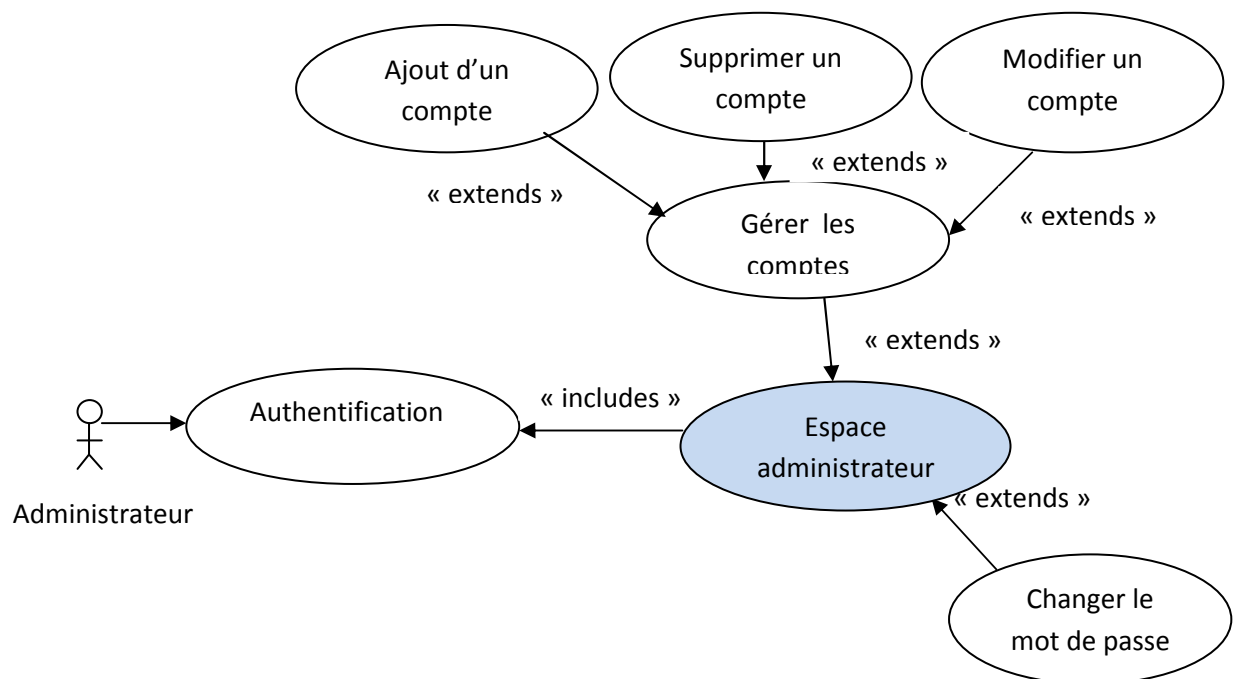


Figure (III.7) : Diagramme global de cas d'utilisation relatif à l'administrateur

III. La conception :

Le but principal de la conception est de rendre le modèle d'analyse réalisable sous forme logicielle, c'est ici que les abstractions métiers viennent pour la première fois, au contact de la réalité logicielle.

Pour modéliser la conception de notre application, nous avons utilisé l'extension d'UML pour le web, en construisant les diagrammes de séquences suivis des diagrammes de classes.

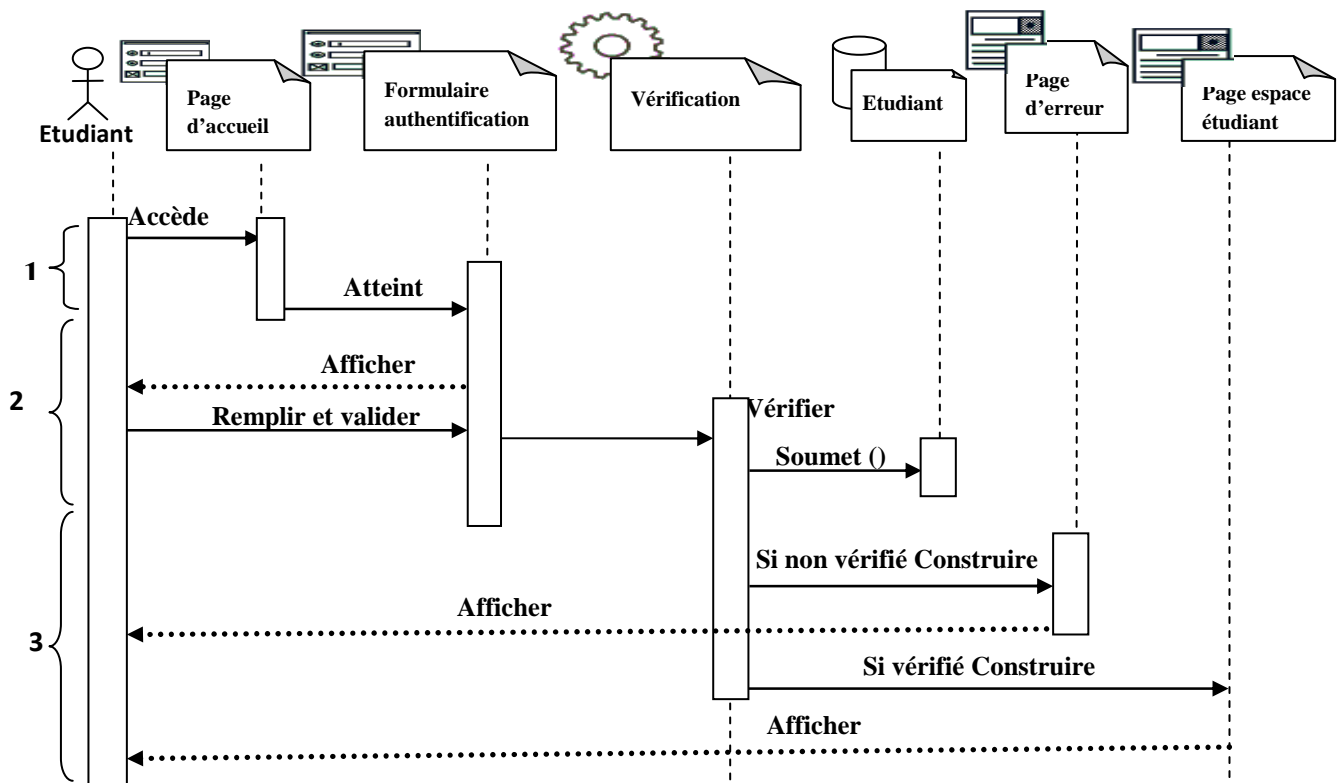
III.1 Diagramme de séquence de réalisation des cas d'utilisation :

Selon le formalisme UML (Unified Modeling Language), les diagrammes de séquences détaillés représentent 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.

✚ Etudiant :

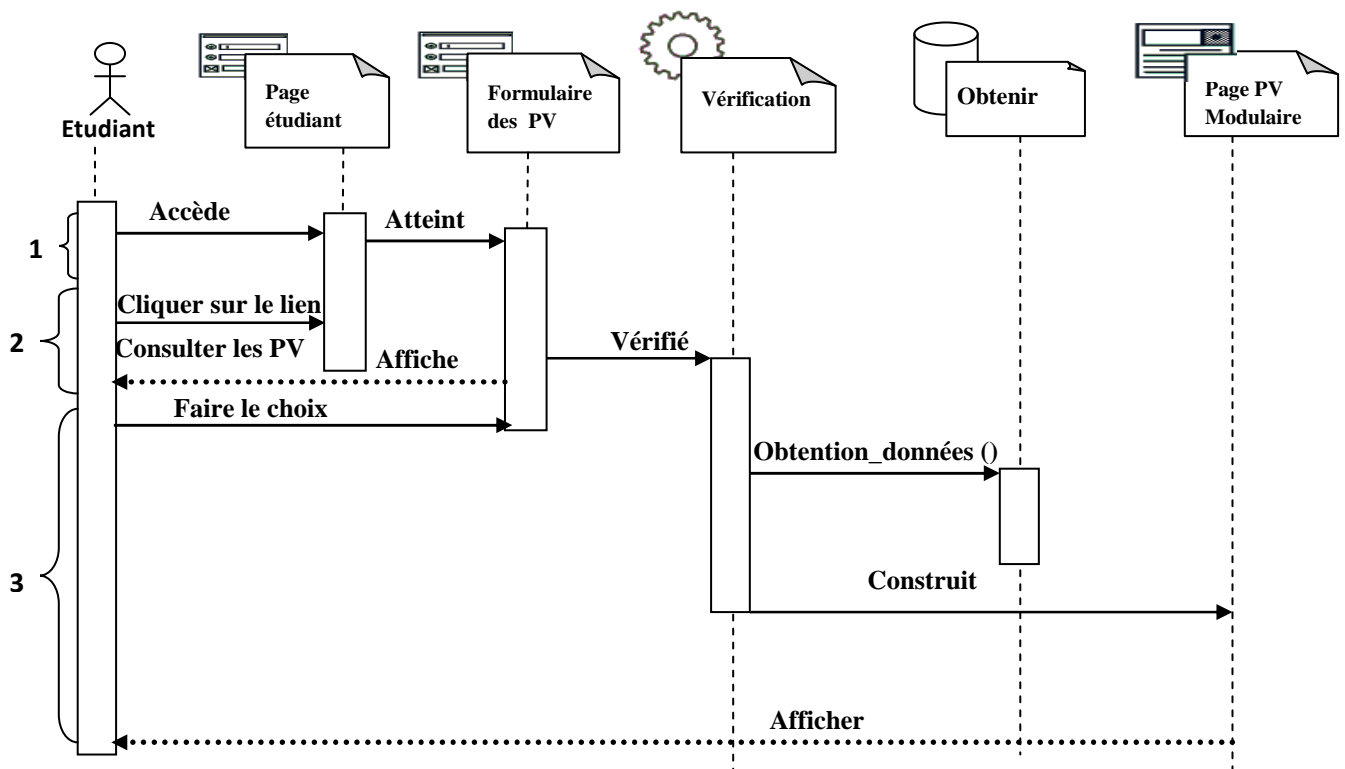
❖ cas d'utilisation « Authentification »



Figure(III.8) : Diagramme de séquence de réalisation de cas d'utilisation « Authentification Etudiant »

- (1) L'acteur étudiant 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 étudiant clique sur le lien étudiant alors le système construit la page authentification étudiant puis l'affiche.
- (3) L'acteur étudiant saisit son login et son mot de passe dans le formulaire authentification puis le valide la saisie le système vérifie alors les données saisies dans la base de données ensuite il construit la page espace Etudiant puis l'affiche .Dans le cas ou les données login et/ou mot de passe ne sont pas juste le système construit un message d'erreur puis l'affiche.

❖ Cas d'utilisation « Consultation des PV modulaire»



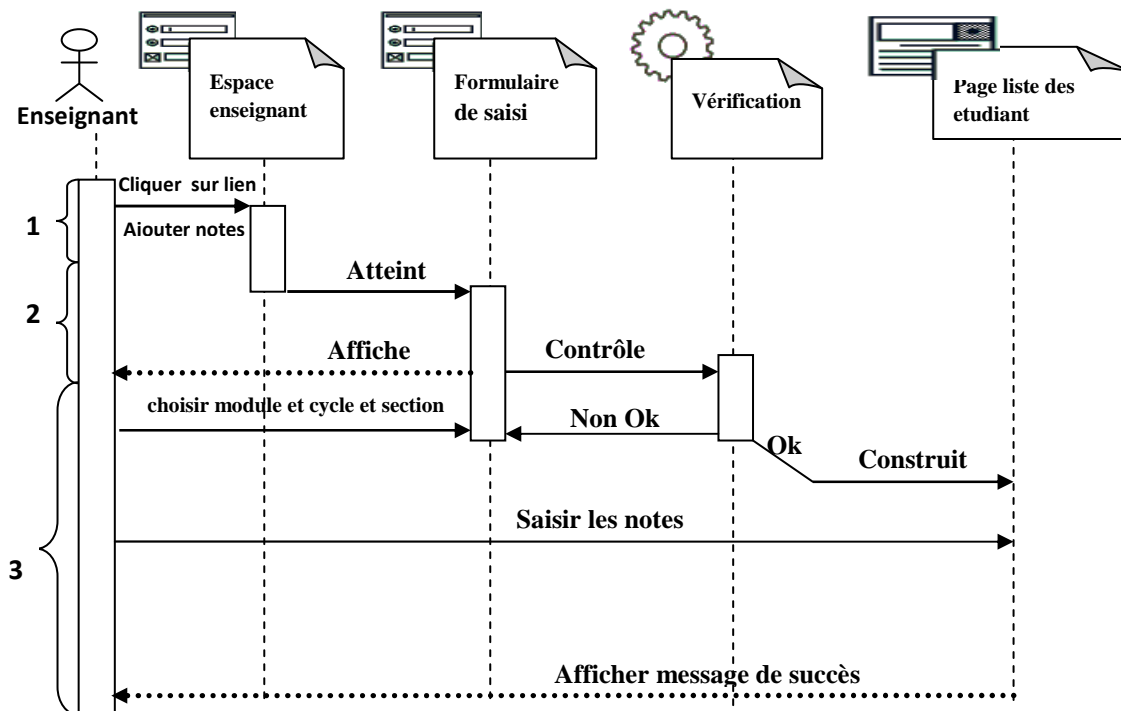
Figure(III.9) : Diagramme de séquence de réalisation de cas d'utilisation « Consultation des PV modulaire»

- (1) L'acteur étudiant accède à son espace le système alors construit la page d'accueil de l'étudiant puis l'affiche.
- (2) L'acteur étudiant clique sur le lien «Consulter PV modulaire » alors le système construit le formulaire de PV modulaire puis l'affiche.

(3) L'acteur étudiant remplit le formulaire puis le valide la saisie le système vérifie alors les données saisies si elles sont dans la base de données ensuite il construit la page PV modulaire puis l'affiche. Dans le cas où les données sont erronées le système construit un message d'erreur puis l'affiche.

Enseignant :

❖ diagramme de séquence de réalisation « enregistrement des notes des étudiants »



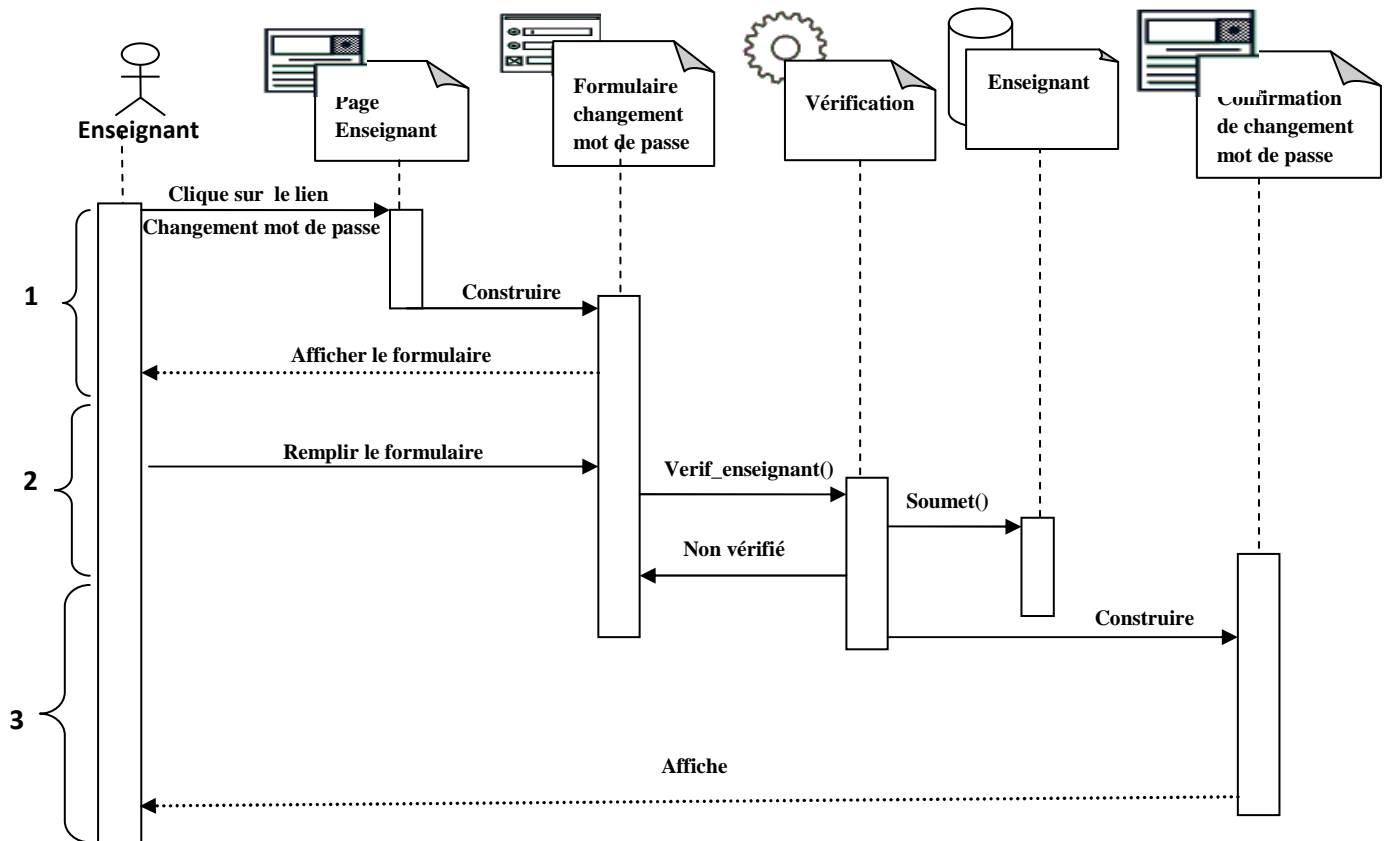
Figure(III.10) : Diagramme de séquence de réalisation de cas d'utilisation « enregistrement des notes »

(1) L'acteur Enseignant clique sur le lien ajouter note qui se trouve dans son espace, le système construit alors le formulaire de choix du module, de cycle et la section puis l'affiche.

(2) L'acteur Enseignant choisit alors le module et le cycle et la section puis le valide en cliquant sur le bouton Valider, le système cherche alors dans la base de données puis construit la liste des étudiants qui convient et l'affiche.

(3) L'acteur Enseignant saisit la notes de chaque étudiant de cette liste et confirme en cliquant sur confirmer, le système affiche un message de succès pour l'enregistrement de ces notes.

❖ Cas d'utilisation « Changement de mot de passe »

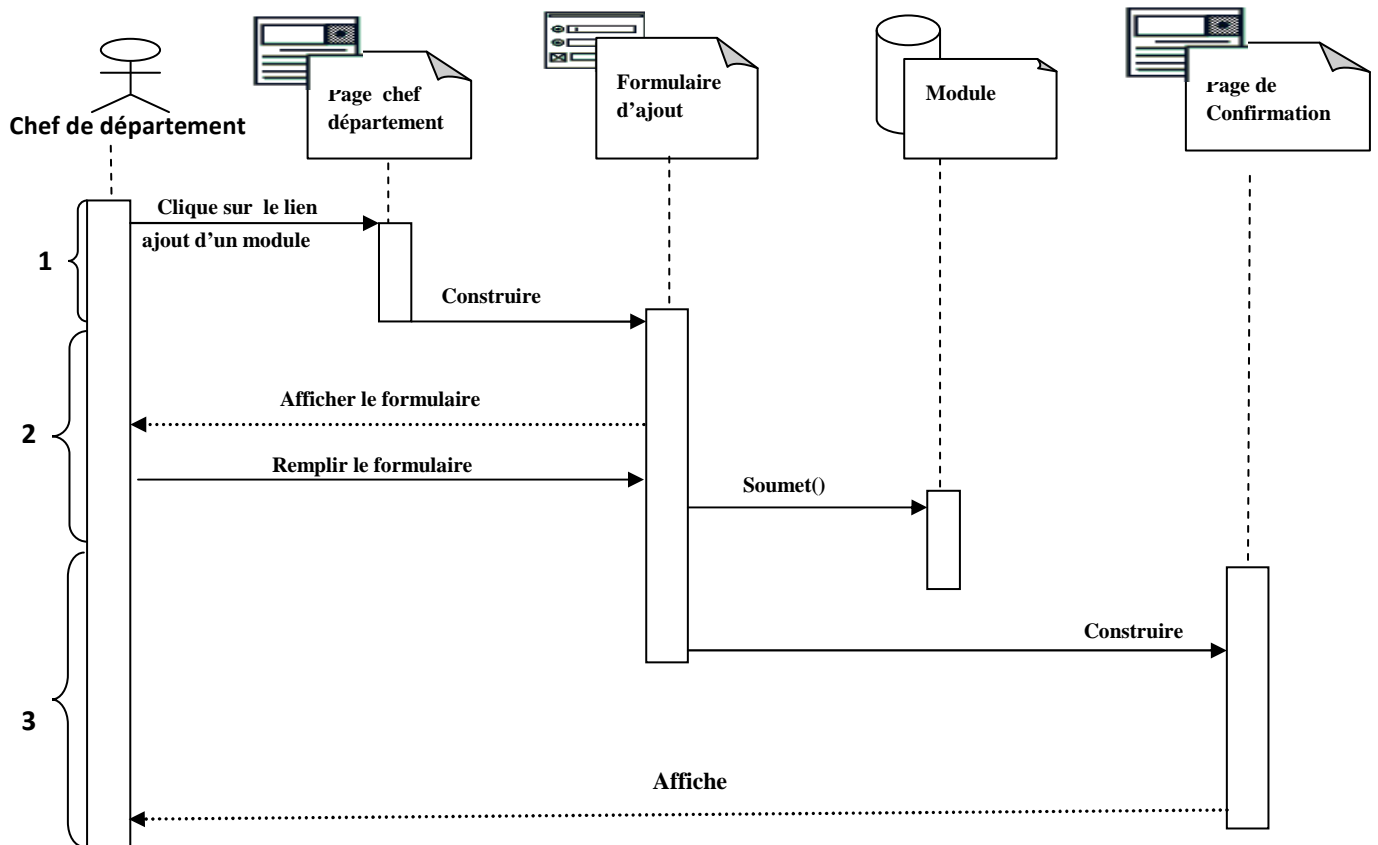


Figure(III.11) : Diagramme de séquence de réalisation de cas d'utilisation « Changement de mot de passe »

- (1) 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.
- (2) 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 du 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.
- (3) Apres avoir modifié le mot de passe dans la base de données le système construit la page de confirmation puis l'affiche.

Chef de département :

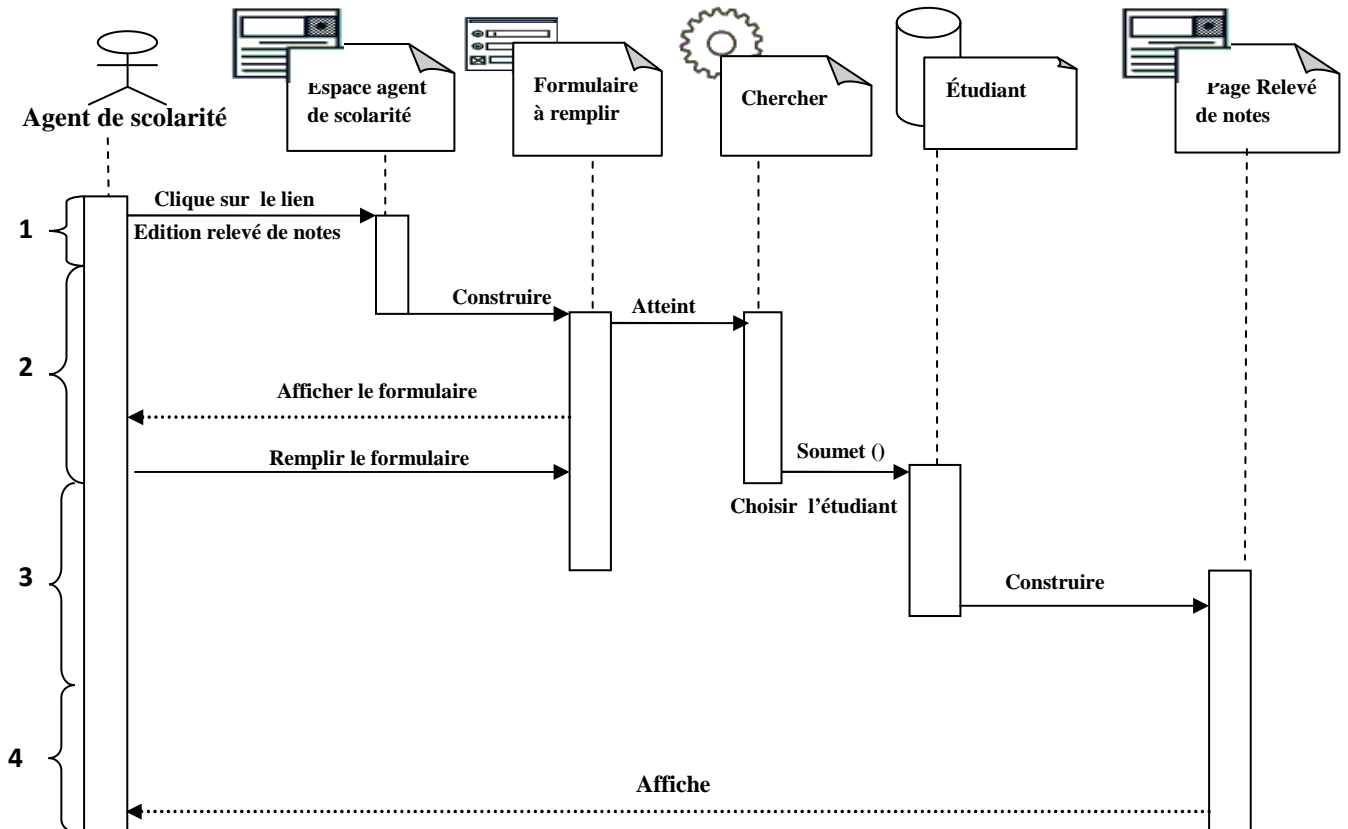
❖ cas d'utilisation « ajout d'un module »



Figure(III.12) : Diagramme de séquence de réalisation de cas d'utilisation «Ajout d'un module »

- (1) L'acteur chef de département accède à son espace, puis il sélectionne le lien « ajout d'un module ».
- (2) Le système lui affiche un formulaire d'ajout, le chef de département choisit et remplit le formulaire, puis valide la saisie en appuyant sur le bouton valider.
- (3) Après avoir fait l'opération dans la base de données le système construit la page de confirmation puis l'affiche.

❖ cas d'utilisation « édition des relevés de notes »

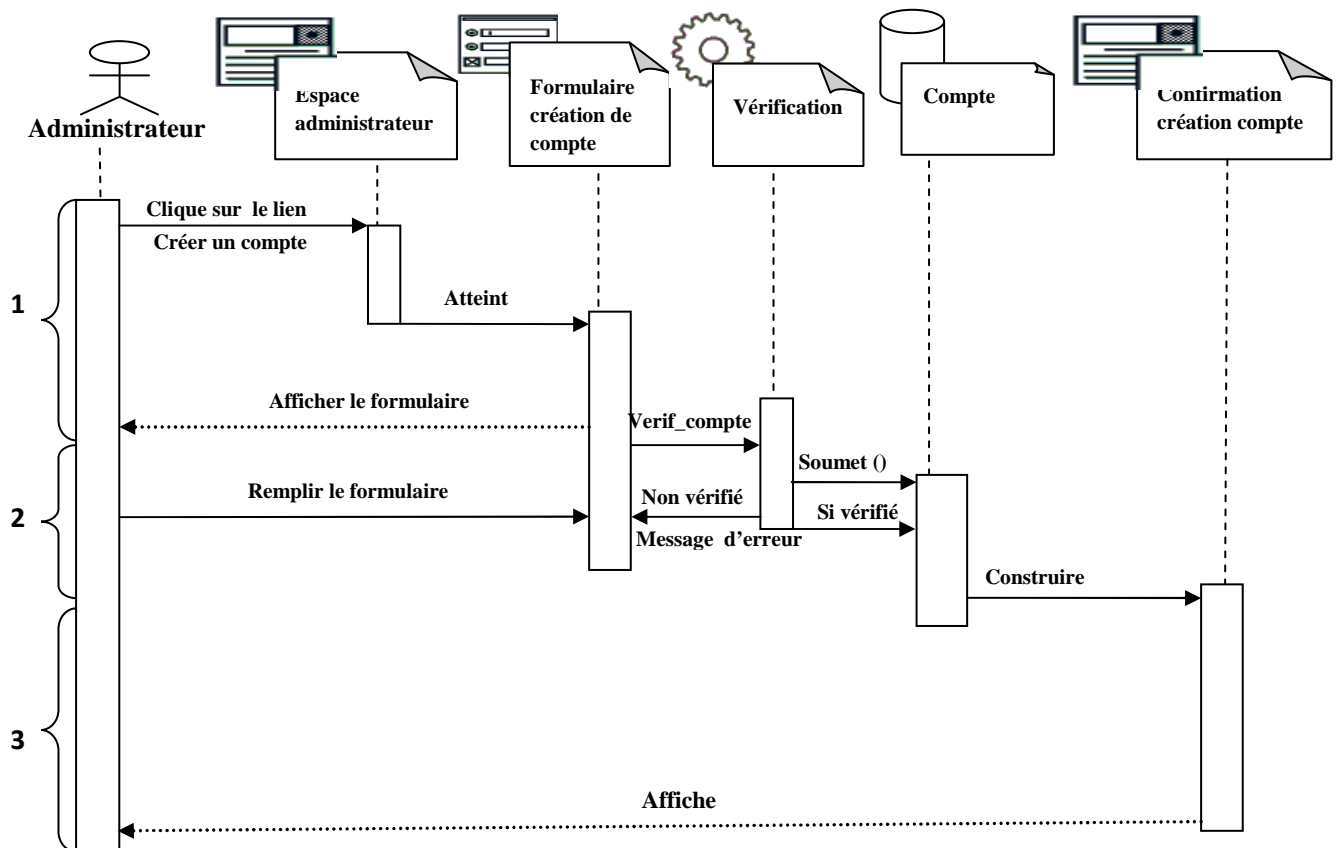


Figure(III.13) : Diagramme de séquence de réalisation de cas d'utilisation « édition des relevés de notes »

- (1) L'acteur agent de scolarité accède à son espace, puis il sélectionne le lien « édition de relevé de notes ».
- (2) Le système lui affiche un formulaire à remplir, l'agent de scolarité remplit le formulaire, puis valide la saisie en appuyant sur le bouton valider.
- (3) Le système affiche la liste des étudiants, l'agent choisit l'étudiant et valide.
- (4) Le système affiche le relevé de notes à imprimer.

Administrateur :

❖ cas d'utilisation « créer un compte »



Figure(III.14) : Diagramme de séquence de réalisation de cas d'utilisation « ajouter un compte »

- (1) L'acteur Administrateur clique sur le lien ajouter compte qui se trouve dans le menu gestion compte 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.

III.2. Diagrammes 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.

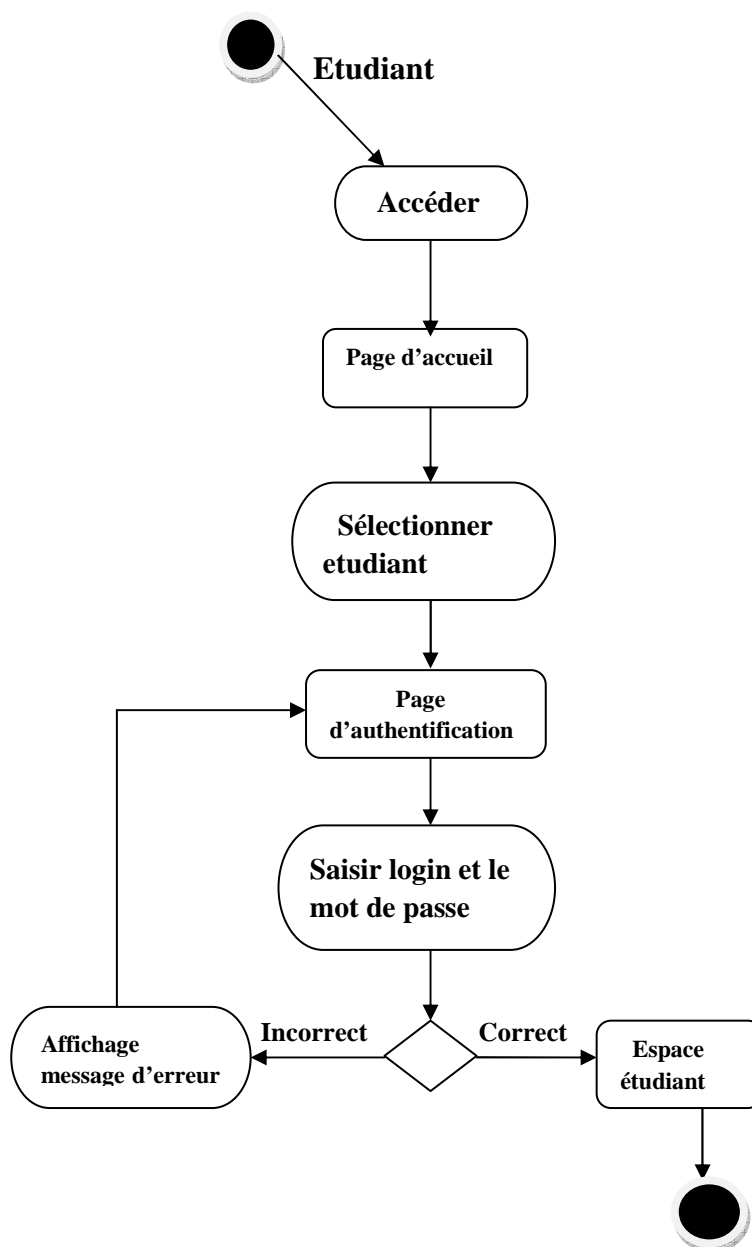
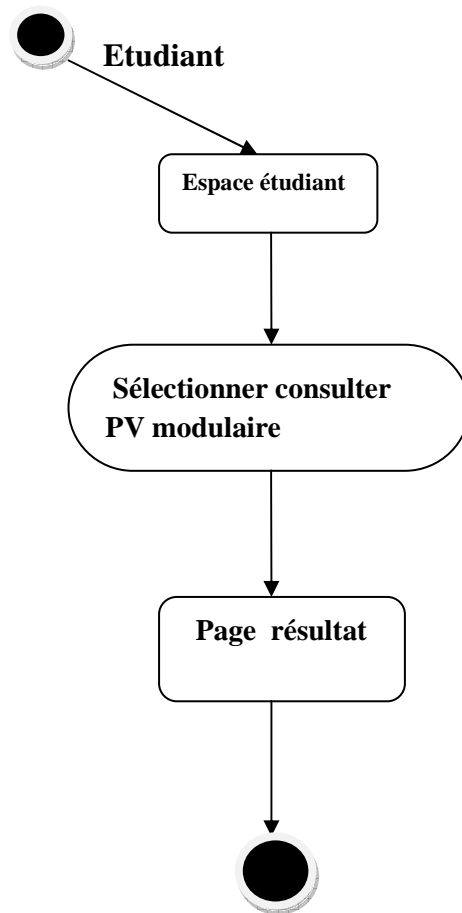
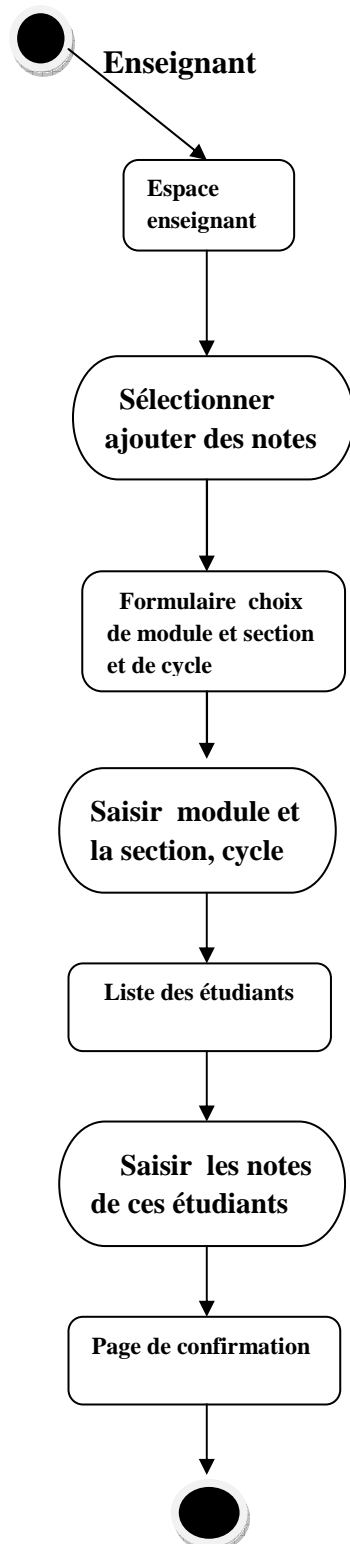
❖ Diagramme d'activités du cas d'utilisation « S'authentifier » pour l'étudiant :

Figure (III.15) : Diagramme d'activités du cas d'utilisation « authentification étudiant»

❖ Diagramme d'activités du cas d'utilisation « Consulter les PV modulaire » :

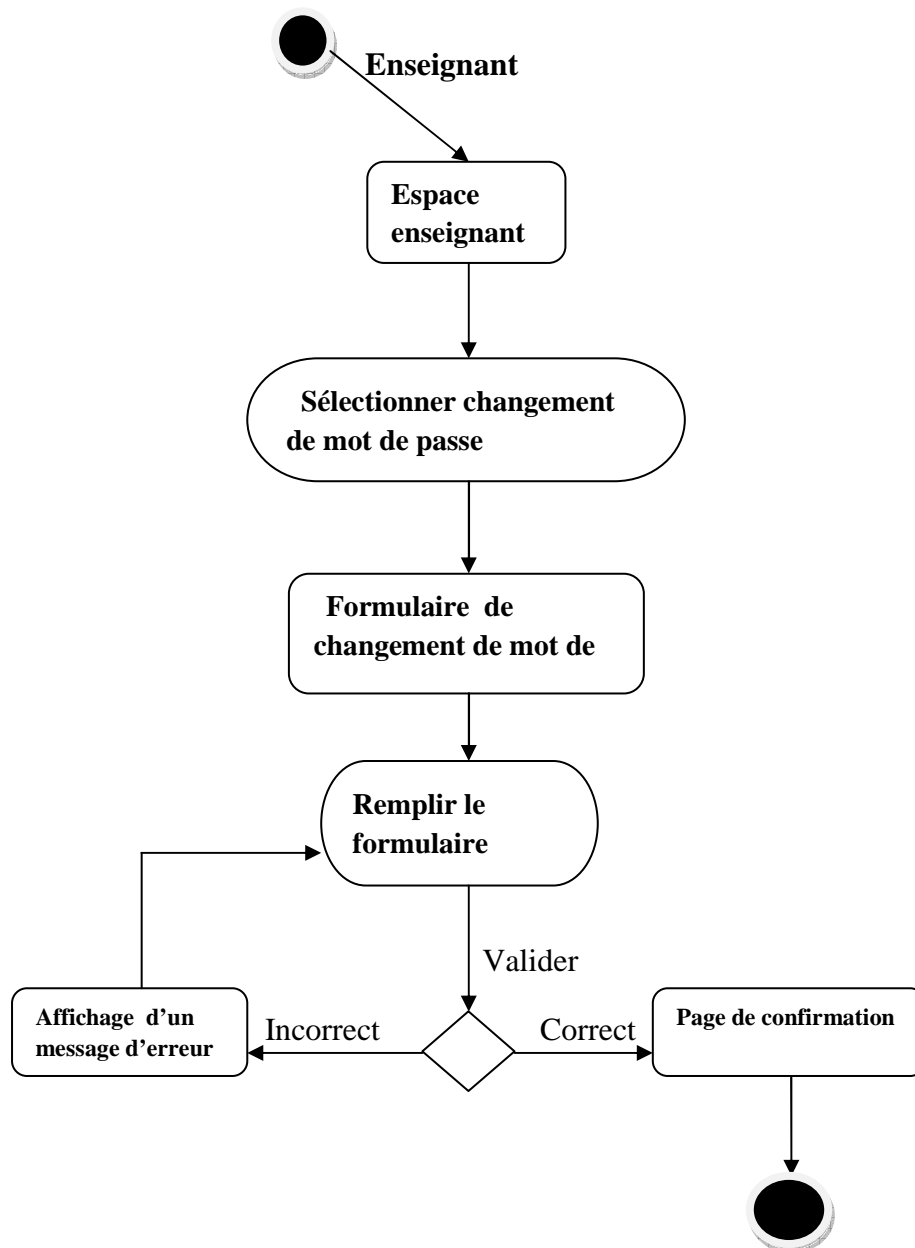
**Figure(III.16)** : Diagramme d'activités du cas d'utilisation «Consulter les PV modulaire »

❖ Diagramme d'activités du cas d'utilisation « Enregistrer les notes des étudiants » :



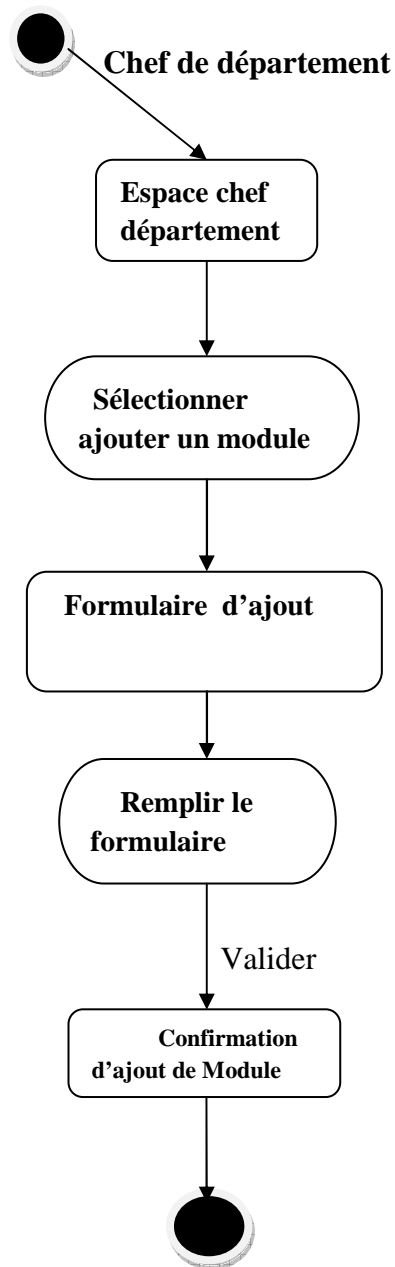
Figure(III.17) : Diagramme d'activités du cas d'utilisation «enregistrement des notes des étudiants»

❖ Diagramme d'activités du cas d'utilisation « Changement de mot de passe » :



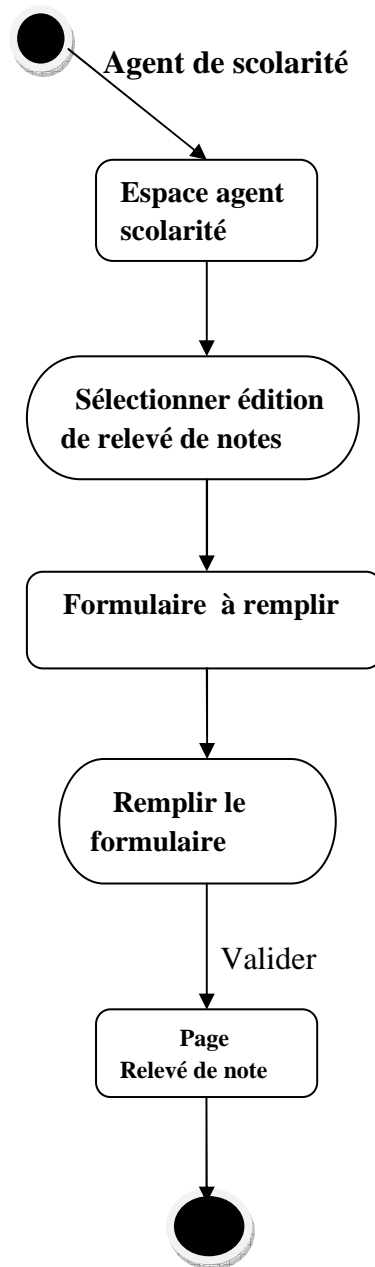
Figure(III.18) : Diagramme d'activités du cas d'utilisation «changement de mot de passe»

❖ Diagramme d'activités du cas d'utilisation « ajouter un module » :

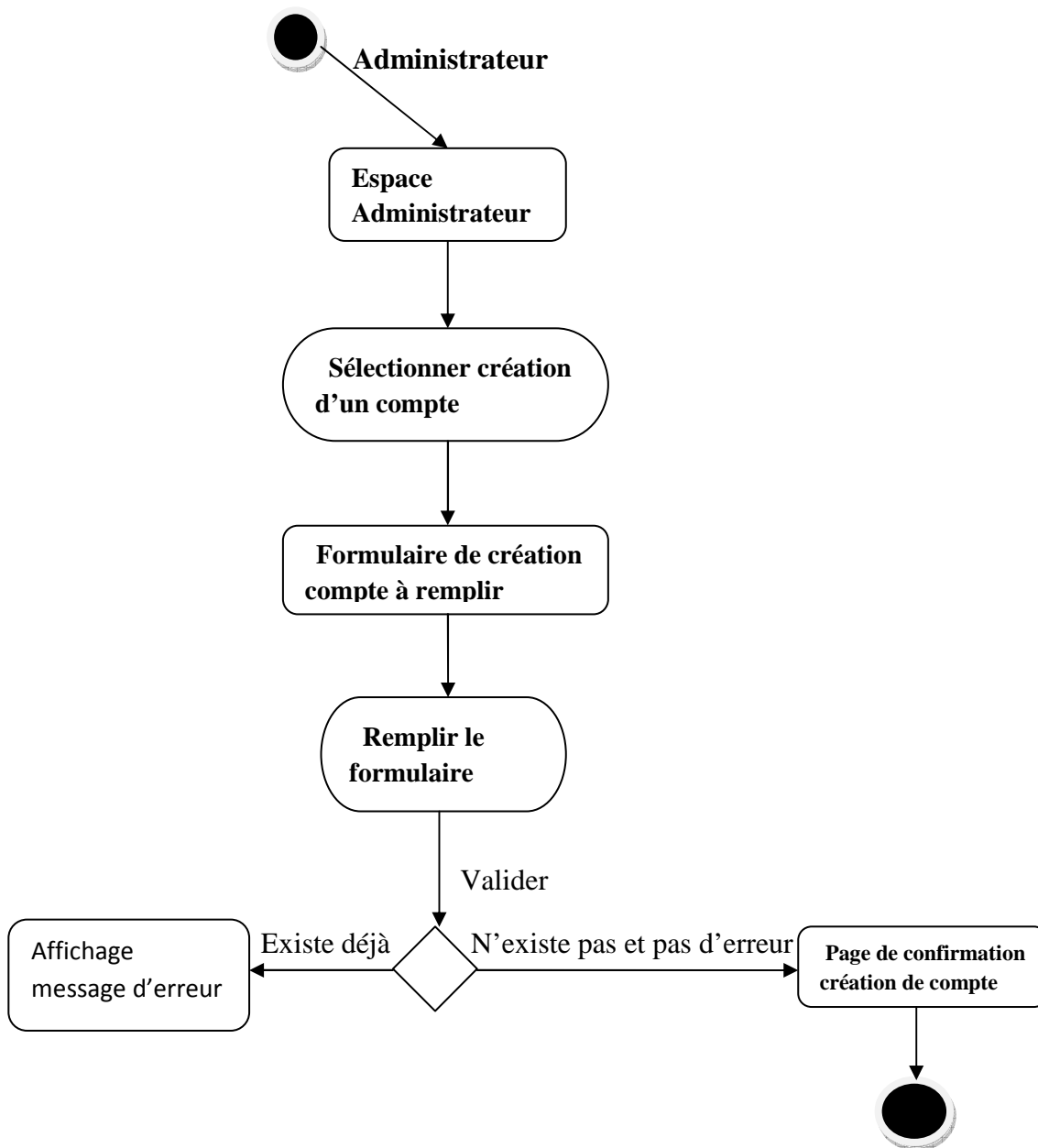


Figure(III.19) : Diagramme d'activités du cas d'utilisation «Ajout d'un module »

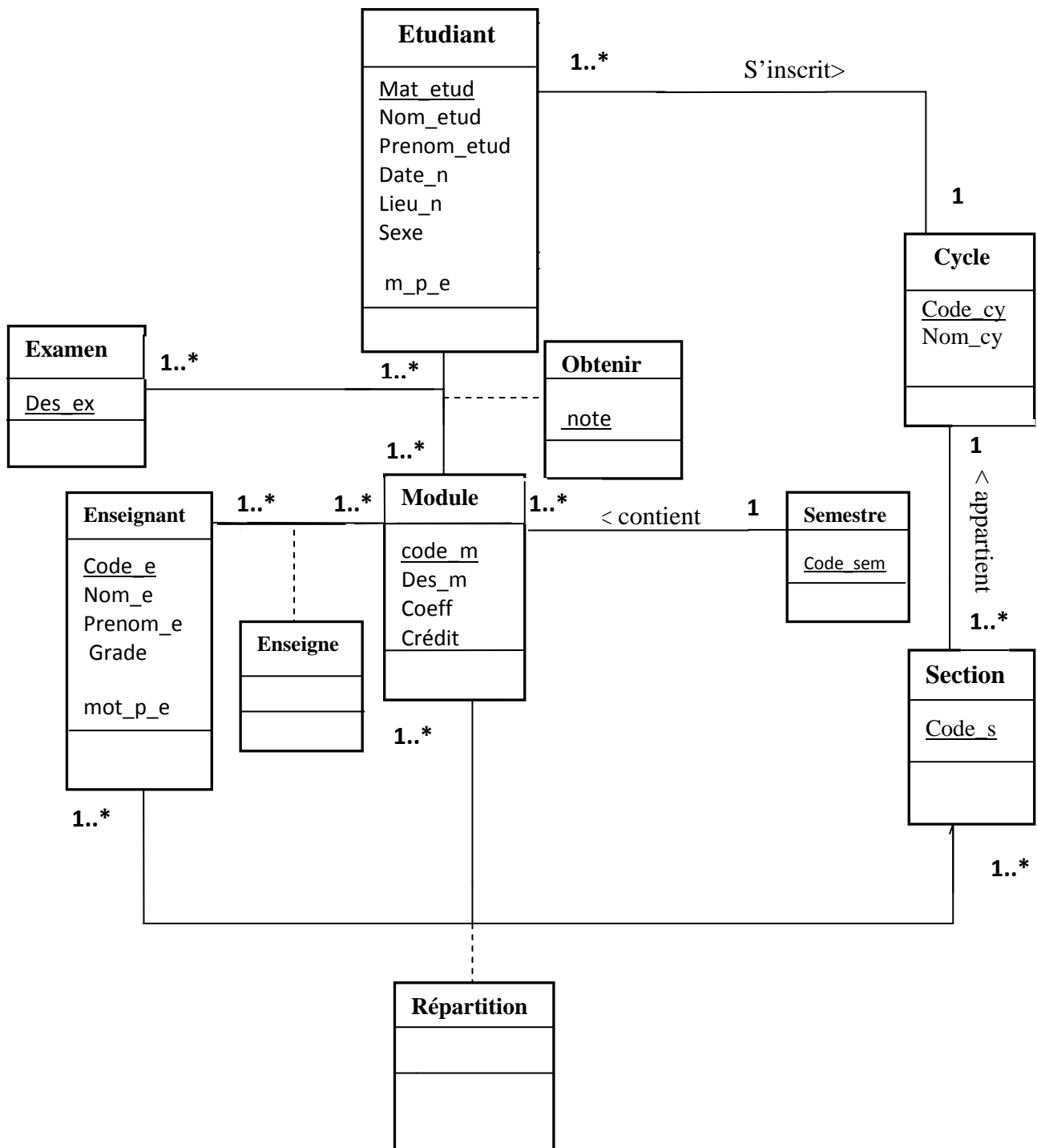
❖ Diagramme d'activités du cas d'utilisation « édition des relevés de notes » :

**Figure(III.20)** : Diagramme d'activités du cas d'utilisation « édition de relevé de notes »

❖ Diagramme d'activités du cas d'utilisation « création d'un compte » :

**Figure(III.21)** : Diagramme d'activités du cas d'utilisation « création d'un compte »

III.3 Diagramme de classe globale de donnée :



Figure(III.22) : Diagramme de classe globale

III.4 Création de la base de données :

III.4.1. Définition d'une base de données :

Une Base de Données (BDD) est un ensemble de données structurées, avec une redondance calculée et accessibles aisément par plusieurs programmes, qui les utilisent selon des objectifs distincts, les propriétés des données ainsi que les relations sémantiques entre ces données sont spécifiées en utilisant les concepts proposés par le modèle de donnée, cette organisation à pour objectif de pouvoir effectuer des inférences sur ces données sous forme de requêtes. Les bases de données sont manipulées par un système de gestion de base de données (SGBD).

III.4.2. Passage au modèle relationnel :

A partir du diagramme de classe globale, nous construisons le modèle relationnel qui va nous permettre de créer la base de donnée sur la quelle s'appuie notre application comme suit :

Cycle (code_cy, nom_cy, ann_etud).

Compte (typecompte, login, motdepasse).

Examen (des_ex).

Enseignant (code_e, nom_e, prenom_e, grade, mot_p_e).

Etudiant (mat_etud, nom_etud, prenom_etud, date_n, lieu_n, sexe, code_cy*, code_s*, m_p_e).

Enseigne (code_e*, code_m*).

Module (code_m, des_m, coeff, credit, code_cy*, code_sem*).

Obtenir (mat_etud*, des_ex*, code_m*, code_sem*, note).

Repartition(code_s*, code_mo*, code_e*, des_ex*, code_cy*) .

Semestre (code_sem).

Section (code_s, code_cy*).

- **Table étudiant :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
mat_etud	La matricule d'étudiant	VARCHAR(10)	primaire
nom_etud	Le nom d'étudiant	VARCHAR(20)	
prenom_etud	Le prénom d'étudiant	VARCHAR(20)	
date_n	La date de naissance d'étudiant	Date	
lieu_n	Lieux de naissance d'étudiant	VARCHAR(25)	
sexe		VARCHAR(10)	
code_cy	Code cycle	VARCHAR(10)	
code_s	Code de section	VARCHAR(10)	étrangère
mot_p_e	Le mot de passe d'étudiant	VARCHAR(10)	étrangère

- **Table module :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
code_m	Code de module	VARCHAR(10)	primaire
des_m	Description de module	VARCHAR(20)	
coeff	Le coefficient de module	Integer	
credit	Le crédit de module	Integer	
code_cy	Code de cycle	VARCHAR(10)	étrangère
code_sem	Code de semestre	VARCHAR(10)	étrangère

- **Table Obtenir :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
mat_etud	La matricule d'étudiant	VARCHAR(10)	étrangère
des_ex	Description d'examen	VARCHAR(10)	étrangère
code_m	Code de module	VARCHAR(10)	étrangère
note	La note d'étudiant	Real	

- **Table Répartition :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
code_s	Code de section	VARCHAR(10)	étrangère
code_mo	Code de module	VARCHAR(10)	étrangère
code_e	Code d'enseignant	VARCHAR(10)	étrangère
des_ex	Description d'examen	VARCHAR(10)	étrangère
code_cy	Code de cycle	VARCHAR(10)	étrangère

- **Table Section :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
code_s	Code de section	VARCHAR(10)	Primaire
code_cy	Code de section	VARCHAR(10)	Etrangère

- **Table Semestre :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
code_sem	Code de semestre	VARCHAR(10)	Primaire

- **Table Cycle :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
Code_cy	Code de cycle	VARCHAR(10)	Primaire
Nom_cy	Nom de cycle	VARCHAR(20)	
Ann_etud	Année d'étude	VARCHAR(10)	

- **Table Examen :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
Des_ex	Description d'examen	VARCHAR(10)	Primaire

- **Table Compte :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
motdepasse	Mot de passe de (administrateur, chef de département, agent de scolarité, étudiant, enseignant)	VARCHAR(10)	primaire
typecompte	Type de compte (administrateur, chef de département, agent de scolarité, étudiant, enseignant)	VARCHAR(25)	
login	Login de (administrateur, chef de département, agent de scolarité, étudiant, enseignant)	VARCHAR(25)	

- **Table Enseignant :**

Nom du champ	Description du champ	Type de donnée	Clé(s)
Code_e	Code de l'enseignant	VARCHAR(10)	primaire
Nom_e	Nom de l'enseignant	VARCHAR(25)	
Prenom_e	Prénom de l'enseignant	VARCHAR(25)	
Grade	Grade de l'enseignant	VARCHAR(10)	
mot_p_e	Mot de passe de l'enseignant	VARCHAR(10)	

- **Table Enseigne:**

Nom du champ	Description du champ	Type de donnée	Clé(s)
code_e	Code de l'enseignant	VARCHAR(10)	étrangère
code_m	Code de module	VARCHAR(10)	étrangère

Conclusion :

Ce chapitre est consacré à l'analyse et la conception de l'application avec le langage de modélisation UML. On a commencé l'analyse et la conception par la description des différentes fonctionnalités et scénarios. Puis on a passé à la description de quelques diagrammes, et en fin on termine ce chapitre par la création de la base de données.

Après avoir effectué l'analyse et la conception de système, on va passer à l'étape de l'implémentation de notre application, qui fera l'objet de prochain chapitre.

Chapitre IV

Réalisation

Introduction

On s'intéressera dans ce chapitre à l'implémentation et la réalisation de notre future application et les différentes fonctionnalités qu'elle nous offre ainsi qu'aux différents outils que nous avons utilisés lors de sa réalisation. En effet, nous y verrons les langages utilisés ainsi que les différentes technologies utilisées pour son développement (Outils de conception web, IDE etc....), et nous aborderons ensuite son fonctionnement et ses différentes interfaces.

I. Les langages de programmation :

I.1.HTML (Hyper Text Markup Language):[19]

HTML est le format de données conçu pour représenter les pages Web. Il permet notamment d'implanter de l'hypertexte dans le contenu des pages et repose sur un langage de balisage, d'où son nom. HTML permet aussi de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables tels que des applets.

HTML permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du Web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade).

I.2. Le langage de programmation Java: [24]

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems. Mais c'est également un environnement d'exécution.

Java peut être séparé en deux parties. D'une part, le programme écrit en langage Java et d'autre part, une machine virtuelle (JVM) qui va se charger de l'exécution du programme Java.

C'est cette plateforme qui garantit la portabilité de Java. Il suffit qu'un système ait une machine virtuelle Java pour que tout programme écrit en ce langage puisse fonctionner.

- **Les Servlets (technique de développement):**

Une servlet est un programme java qui fonctionne sur un serveur Web et dont le rôle consiste à apporter une réponse à une requête. Elle constitue l'outil idéal pour l'implémentation du tiers médian dans les applications 3-tiers. Une servlet peut ainsi recevoir une requête envoyée par un navigateur Web, négocier les informations demandées avec une base de données et renvoyer le résultat de la requête au navigateur.

Pour exécuter une servlet, il suffit de taper son URL dans la zone d'adresse d'un navigateur ou de l'interroger dans une page Web.

Une servlet peut être invoquée plusieurs fois en même temps pour répondre à plusieurs requêtes simultanées. La servlet se positionne dans une architecture Client/serveur trois tiers dans le tiers du milieu entre le client léger chargé de l'affichage et la source de données.

- **Les JSP :**

JSP = Java Server Pages, c'est un fichier contenant du code HTML et des fragments de code Java exécutés sur le moteur de servlets comparable aux langages côté serveur de type PHP, ASP, les pages JSP sont converties en servlet par le moteur de servlets lors du premier appel à la JSP.

I.3.JavaScript :[20]

JavaScript (initialement appelé Live script) a été développé par Netscape puis a été repris par la firme Sun.

Les scripts sont des instructions (des lignes de code) interprétées par les navigateurs Netscape et Internet Explorer. Le JavaScript n'a cessé d'évoluer avec les versions des navigateurs.

JavaScript est un langage de Script qui s'incorpore au langage HTML de présentation des pages web. Les scripts vont permettre de rendre une page web plus dynamique.

II. Environnement et outils de développement :

II.1. Macromedia Dreamweaver :[21]

Macromedia Dreamweaver est un éditeur professionnel de site Web destiné à la conception, au codage et au développement de sites, de pages et d'applications Web. Il fut l'un des premiers éditeurs HTML de type WYSIWYG (What You See Is What You Get : ce que vous voyez est ce que vous obtenez), mais également l'un des premiers à intégrer un gestionnaire de site Web (CyberStudioGoLive étant le premier). Quel que soit l'environnement de travail utilisé (codage manuel HTML ou environnement d'édition visuel), Dreamweaver propose des outils relativement simples d'utilisation qui aident l'utilisateur à développer et gérer des applications dynamiques reposant sur des bases de données sans connaissance préalable des langages de programmation.

L'interface de Macromedia Dreamweaver est illustrée dans la figure suivante :

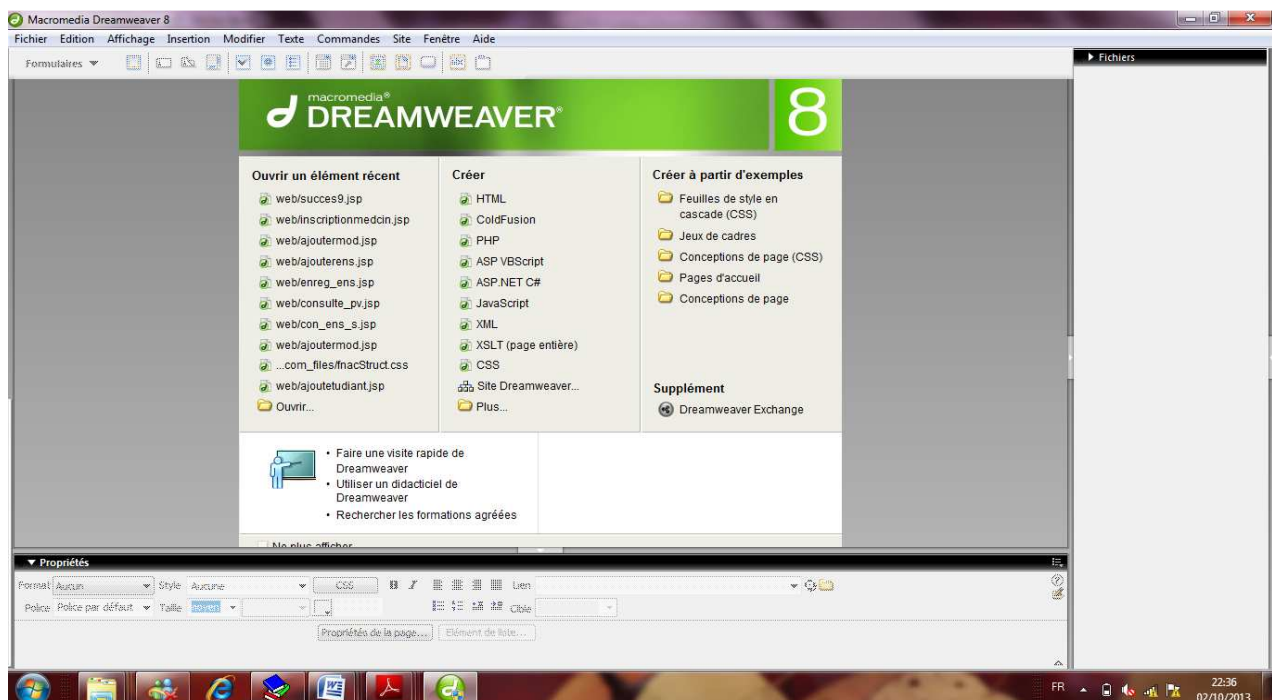


Figure (IV .1): Interface Principale de Dreamweaver8

II.2. Macromedia flash :

Macromedia flash est un logiciel pour créer des animations à base des simples images, comme il permet d'intégrer des vidéos, des sons, et des effets spéciaux pour enrichir le site et aboutir à un meilleur résultat. L'interface de Macromedia flash est illustrée dans la figure :



Figure(VI.2):Interface Macromedia flash

II.3.IDE NetBeans 6.8:[25]

Cet IDE a été créé à l'initiative de Sun Microsystems. Il présente toutes les caractéristiques indispensables à un environnement de qualité, que ce soit pour développer en Java, Ruby, C/C++ ou même PHP.

NetBeans est sous licence OpenSource, il permet de développer et déployer rapidement et gratuitement des applications graphiques Swing, des Applets, des JSP/Servlets, des architectures J2EE, dans un environnement fortement personnalisable.

L'IDE NetBeans repose sur un noyau robuste, la plateforme NetBeans, que vous pouvez également utiliser pour développer vos propres applications Java est un système de plugins performant, qui permet d'avoir un IDE modulable.

A côté de la version complète de l'IDE NetBeans, il existe différentes déclinaisons qui se concentrent sur une plateforme ou un langage précis (Java ME, Java : SE + ME + EE, Ruby, C/C++, PHP).

Enfin cet IDE possède un débogueur de grande qualité ainsi qu'une interface graphique améliorée.

L'explorateur comprend un éditeur de requêtes, un gestionnaire intégré de bases de données MySQL.

L'interface d'IDENetBeans 6.8est illustrée dans la figure suivante :

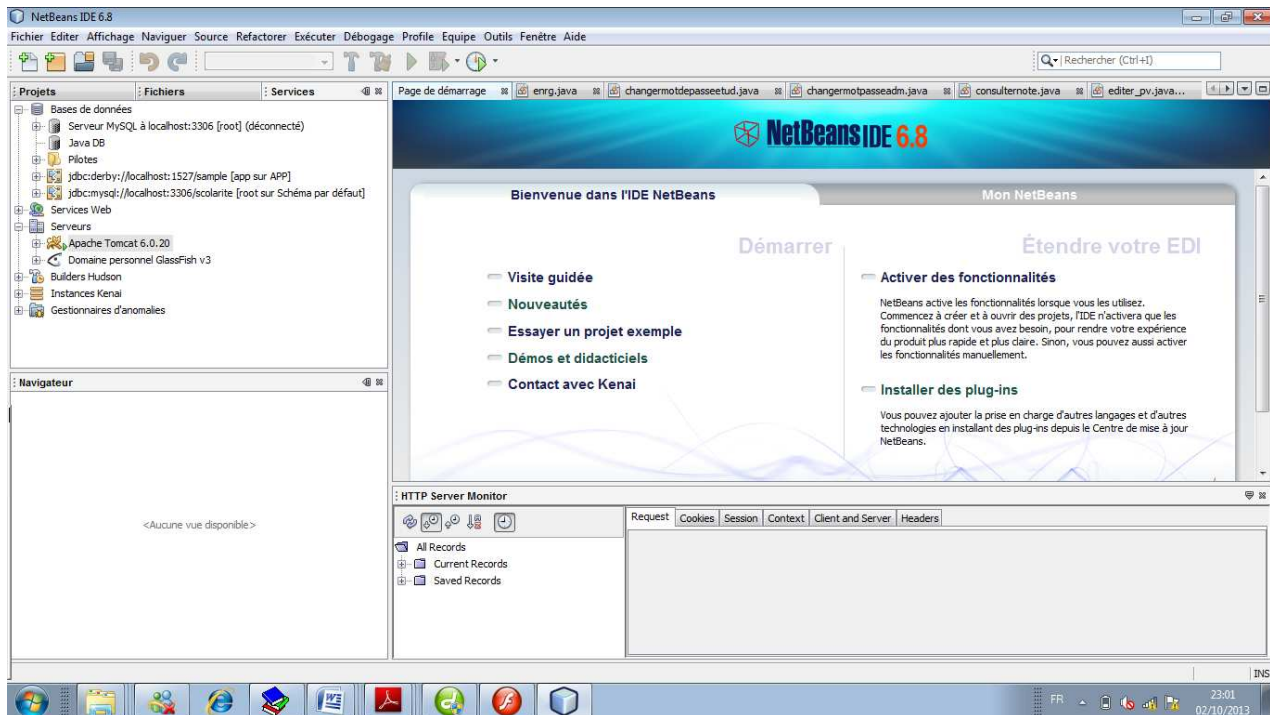


Figure (IV.3): Interface de Neatbeans6.8

II.4. Les Serveurs :

II.4.1. Présentation de MySQL :[22]

Le logiciel MySQL est un serveur de base de données SQL très rapide, multithread, et robuste. Il est le logiciel open source le plus utilisé dans le monde.

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

MySQL est caractérisé par ses fonctionnalités multiples comme le multitraitement en utilisant plusieurs CPU ou bien la diversification de l'outil linguistique qui peut de ce fait afficher au client les messages d'erreurs en plusieurs langues.

Nous avons donc opté pour l'utilisation d'une base de données de type MySQL qui est simple à installer et à utiliser, et pour ce faire nous avons utilisé la version 5.4 de MySQL Serveur.

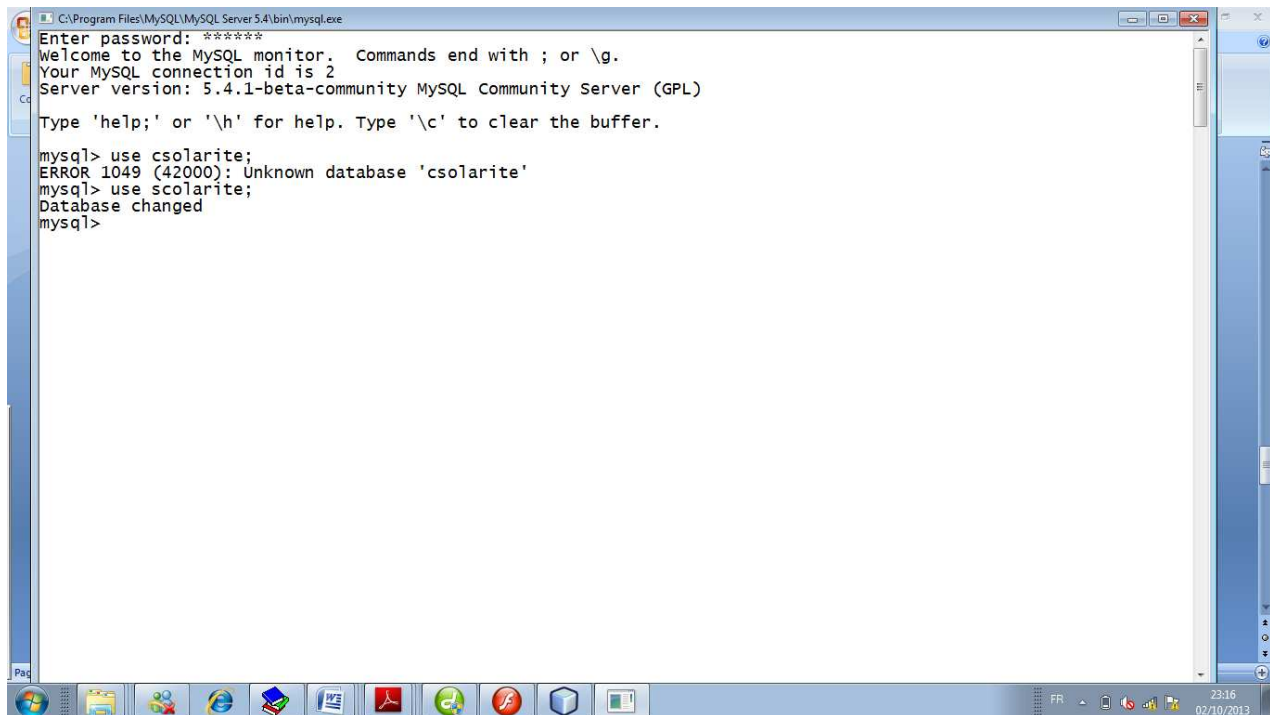


Figure (IV.4): Interface principale MySQL Serveur 5.4

II.4.2. Le serveur apache :[23]

Le serveur http apache est le fruit de travail d'un groupe de volontaires, The Apache Groupe, qui a voulu réaliser un serveur web du même niveau que les produits commerciaux sous forme de logiciels libre .Notre choix pour apache est motivé par les nombreux avantages qu'il présente :

Avantage d'Apache :

- Extrême portabilité (fonctionne sous les différentes plates-formes UNIX et Windows).
- Modestie des exigences matérielles requises pour un niveau élevé de performance.
- Gratuit.
- Extensible, modulaire et configurable.
- Etc.

II.4.3.Le module Tomcat :

Le conteneur de servlets choisi est le moteur Tomcat, (ou moteur d'exécution),il est utilisé par Sun Microsystems pour l'implémentation de référence des servlets etdespages JSP. C'est en fait un serveur HTTP et un serveur de servlets/JSP tout-en-unécrit en Java, ce qui rend cet outil très pratique pour tester un site Internet qui gèredes pages statiques et des pages dynamiques avec des servlets et des pages JSP.

Tomcat peut aussi être intégré au serveur WebApache.

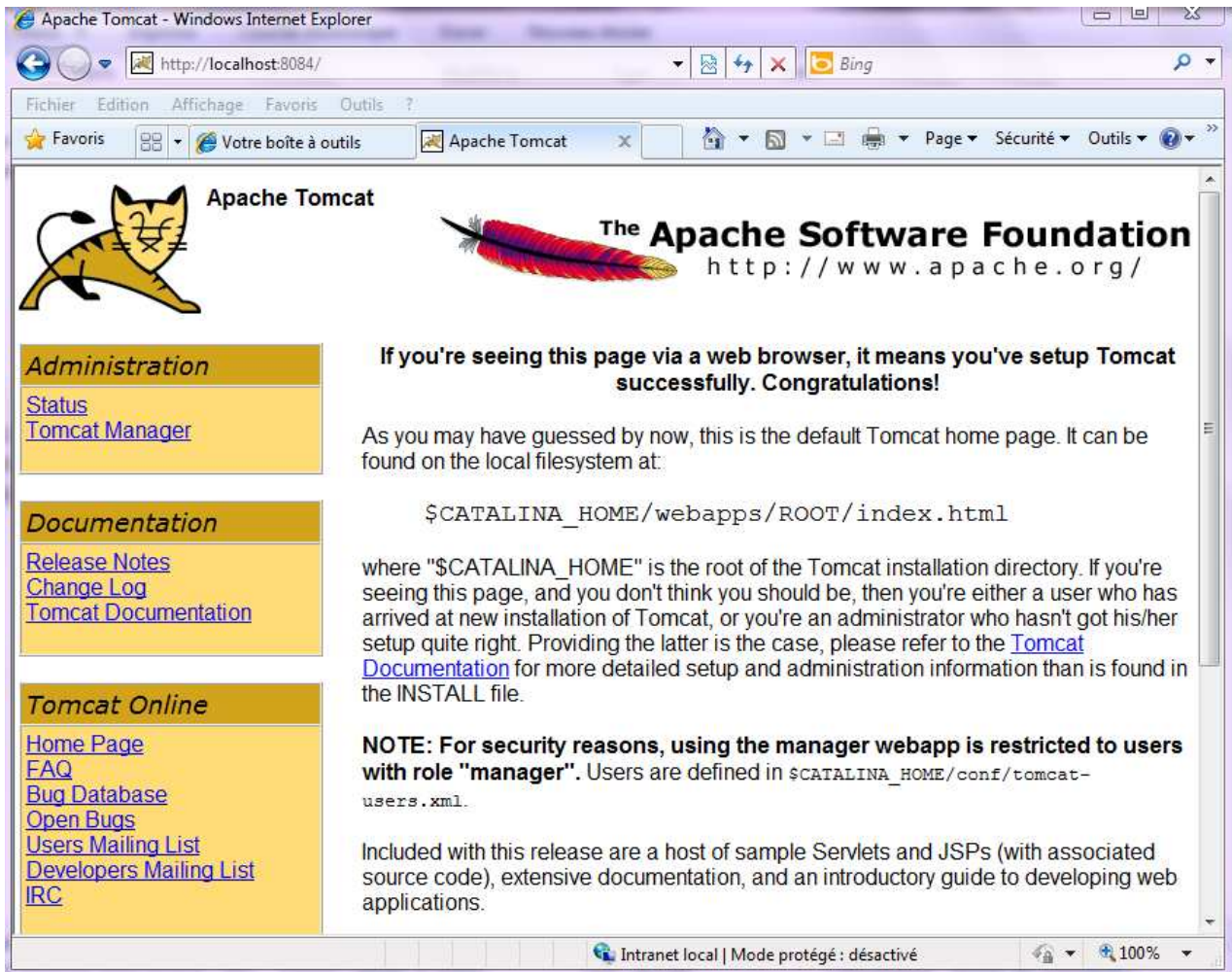


Figure (VI.5) : Interface d'Apache Tomcat.

III. Présentation de quelques interfaces et fonctionnalités de l'application :

Nous allons vous montrer dans cette partie les interfaces principales de notre application :

III.1. L'interface Principale Page d'accueil :

C'est notre interface d'accueil de notre application, qui sera accessible par tous les utilisateurs :



Figure (IV.6): Page d'accueil Principale

III.2. La page Authentification :

Cette page est nécessaire pour l'authentification des différents acteurs qui vont utiliser cette application (étudiant, enseignant, chef de département, agent de scolarité, administrateur), ce qui va leur permettre d'accéder à leur espace personnel et pour cela il suffit de cliquer sur le lien authentification ensuite remplir le formulaire (Login, Mot de passe) et valider la saisie.

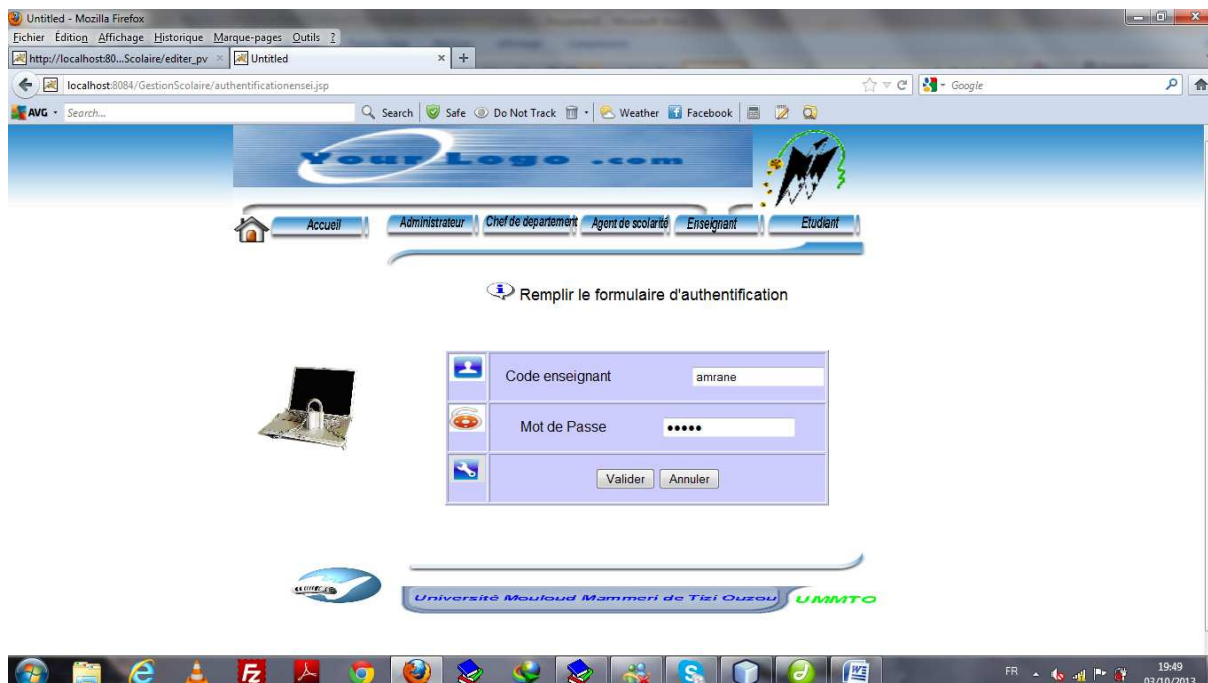
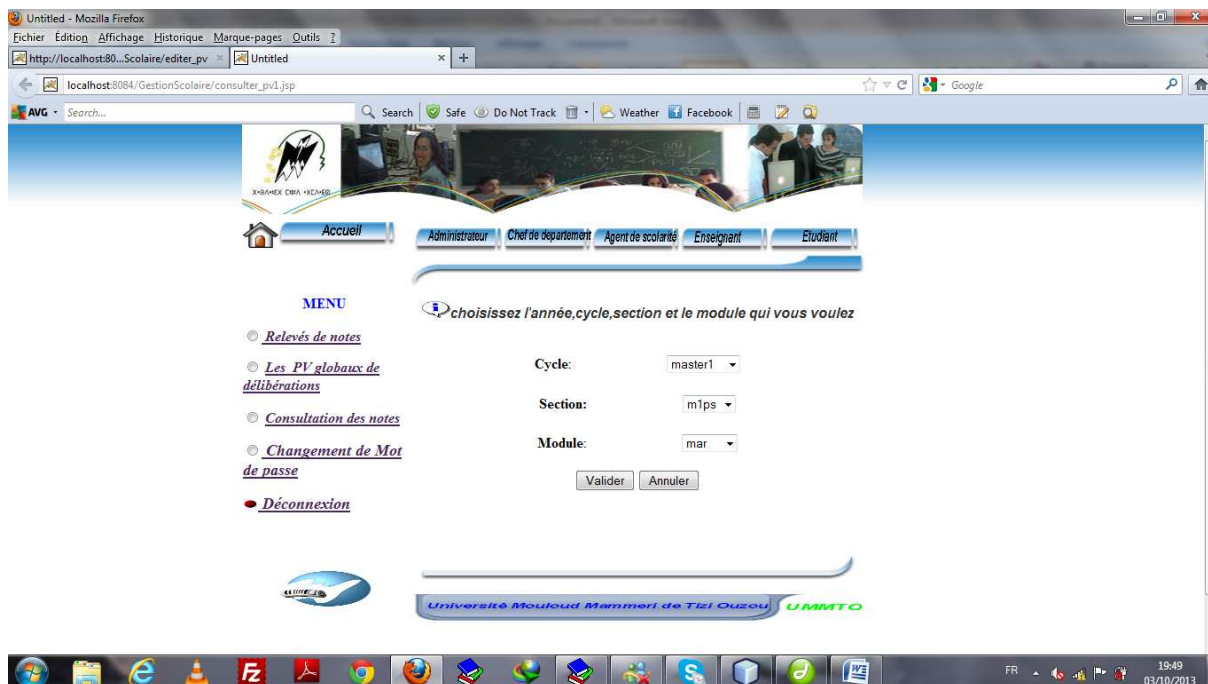


Figure (IV.7): Page authentification de l'enseignant

III.3. La page consultation de PV modulaire :

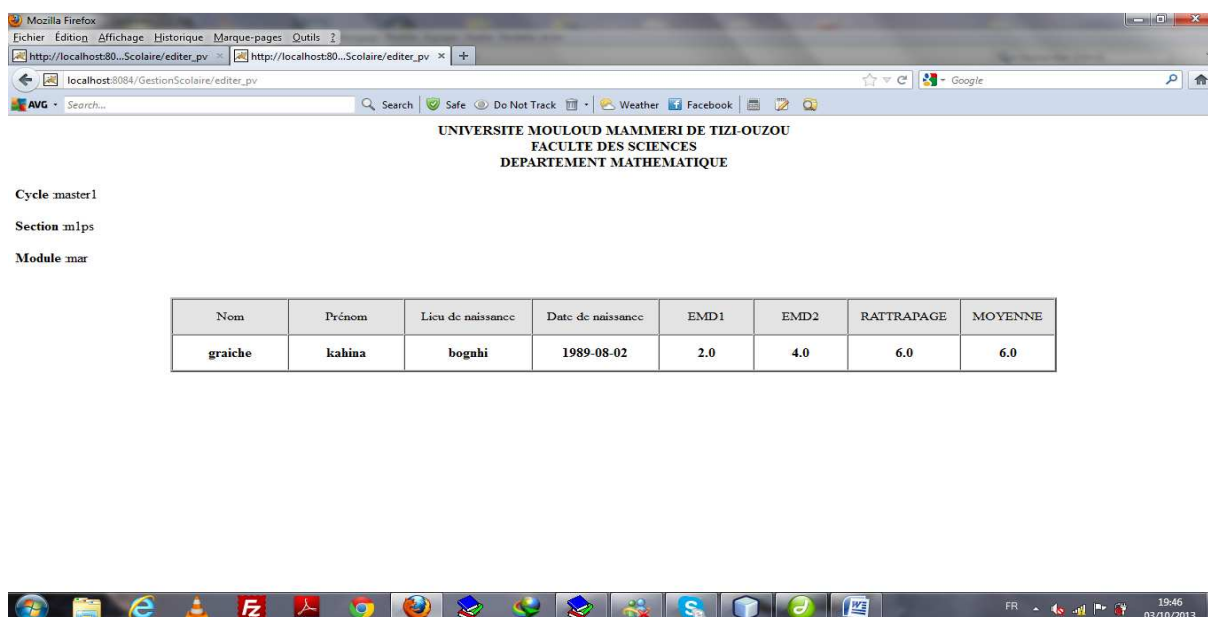
L'étudiant ou le chef de département peut voir les PV modulaire en cliquant sur consulter PV.



Figure(IV.8) : consulter PV modulaire

III.4. La page affichage des PV modulaire :

Après le choix de l'étudiant sur le cycle et la section et le module puis il valide, le système lui affiche le PV modulaire de ce module.



Figure(IV.9) : Affichage de PV modulaire

III.5. La page consulté relevé de notes :



Figure (IV.10): Consulter relevé de notes

III.6. Page d'édition de relevé de notes :

Après avoir choisit la matricule d'étudiant, le système affiche le relevé de notes de cet étudiant :

Faculté: FACULTE DES SCIENCES
Département: DEPARTEMENT MATHÉMATIQUES

RELEVÉ DE NOTES

Le chef de Département Informatique atteste que l'étudiant(e)
Non: graïche Prénom:kahina né le 1989-08-02
Inscrit(e) en 'master1 Option:mlps
a obtenu les résultats suivants durant cette année:

Semestre1						Semestre2					
Module	noteEMD	ratt	Coeff	Moy	Crédit	Module	EMD	Ratt	Coeff	Moy	Crédit
anglai	14.0	-	6	14.0	6	statistique bayesien	13.0	-	10	13.0	10
file d'attente	9.5	13.0	7	13.0	7	statistique multivariee	5.0	8.0	10	8.0	
martingale	3.0	6.0	8	6.0		theorie de langage	14.5	-	10	14.5	10
methode M	11.5	-	9	11.5	9						
avec chaine de M											

Moyenne du semestre 1: 10.883333
Crédit du semestre 1: 22.0

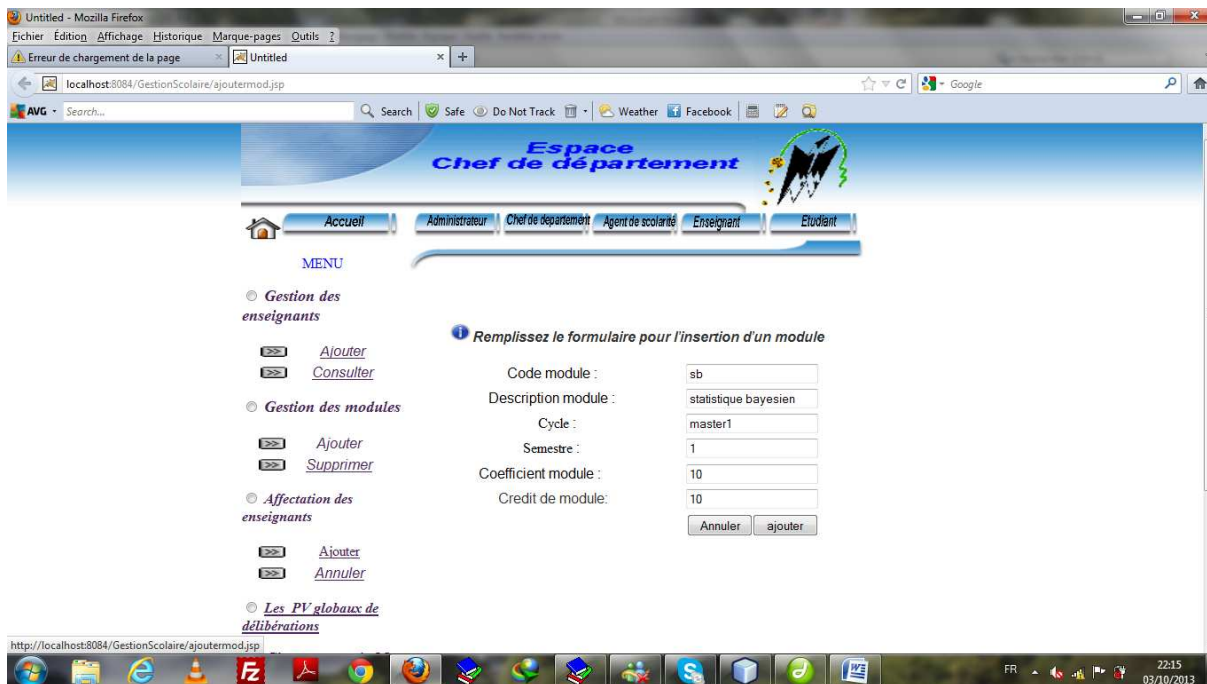
Moyenne du semestre 2: 11.833333
Crédit du semestre: 20.0

Somme des credits:42.0

Figure (IV.11): édition d'un relevé de notes

III.7. Page ajout d'un module :

Cette page peut être atteinte par le chef de département, cette page peut l'atteindre après avoir effectué l'authentification, il pourra ajouter un module où cette interface le montre :



Figure(IV.12): Ajout d'un module

III.8. Page d'ajout d'un enseignant sans remplir tout les champs :

Cette page renvoi un message de manque de l'un des champs après avoir effectuer le contrôle des champs :

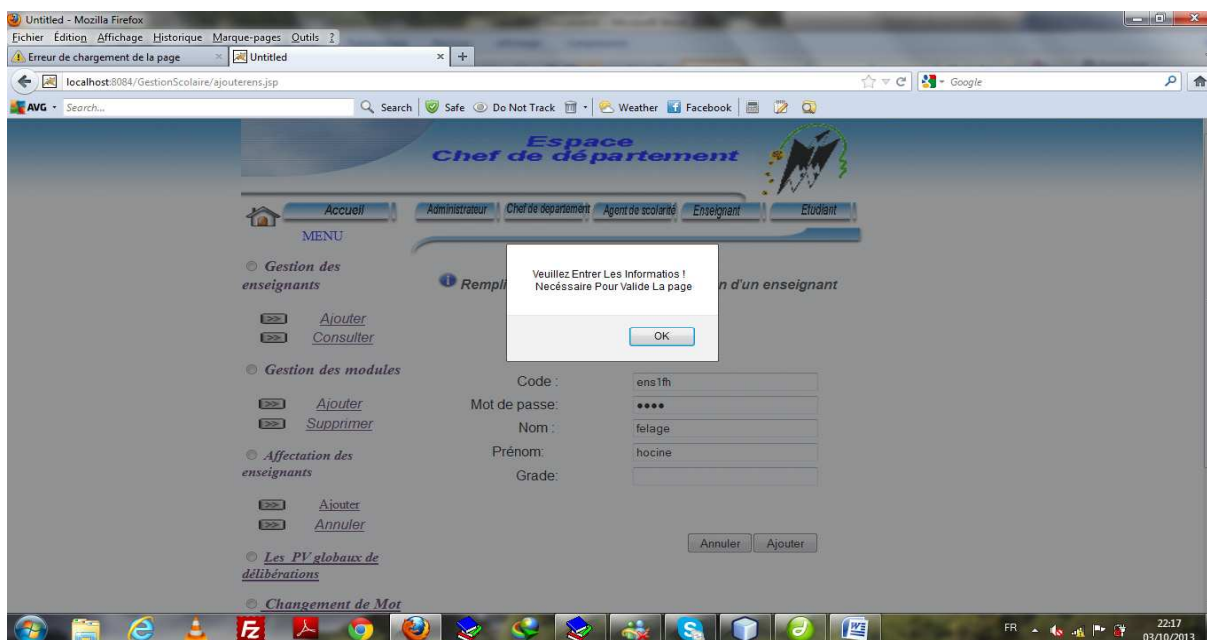


Figure (IV.13): ajout d'un enseignant sans remplir tout les champs

III.9. Page d'ajout d'un enseignant :

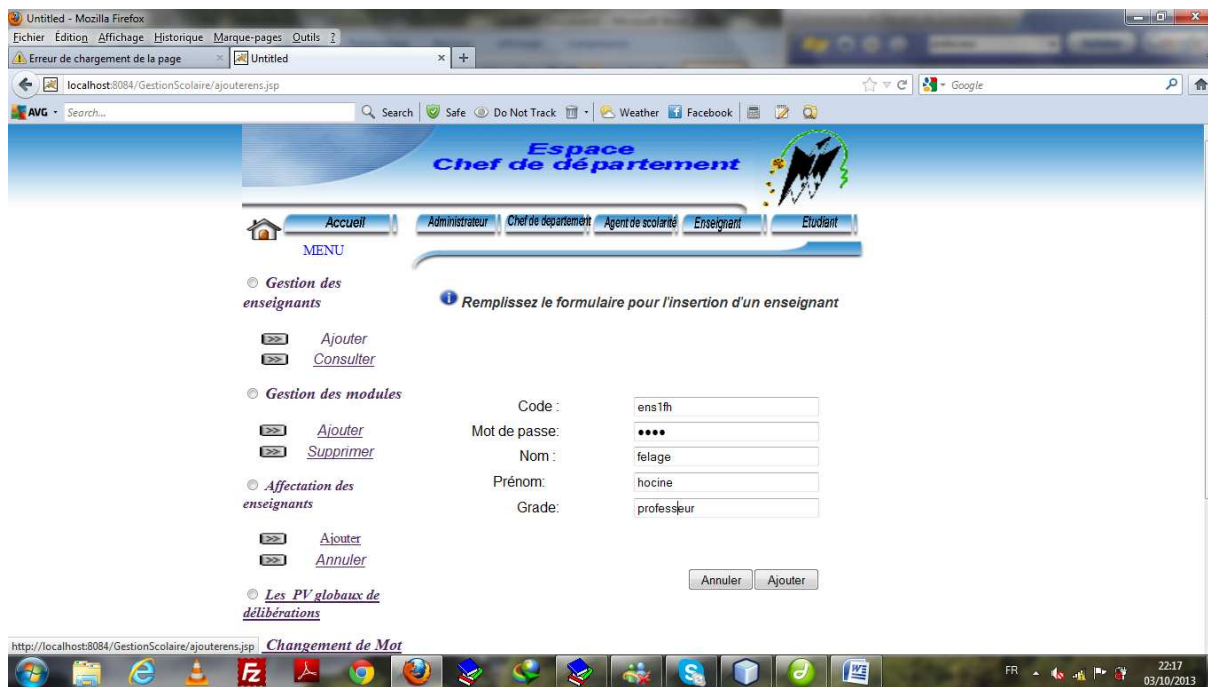


Figure (VI.14): Ajout d'un enseignant

III.7. Page liste des enseignants :

Le chef de département peut voir toute la liste des enseignants existants, en cliquant sur consulter liste enseignant.

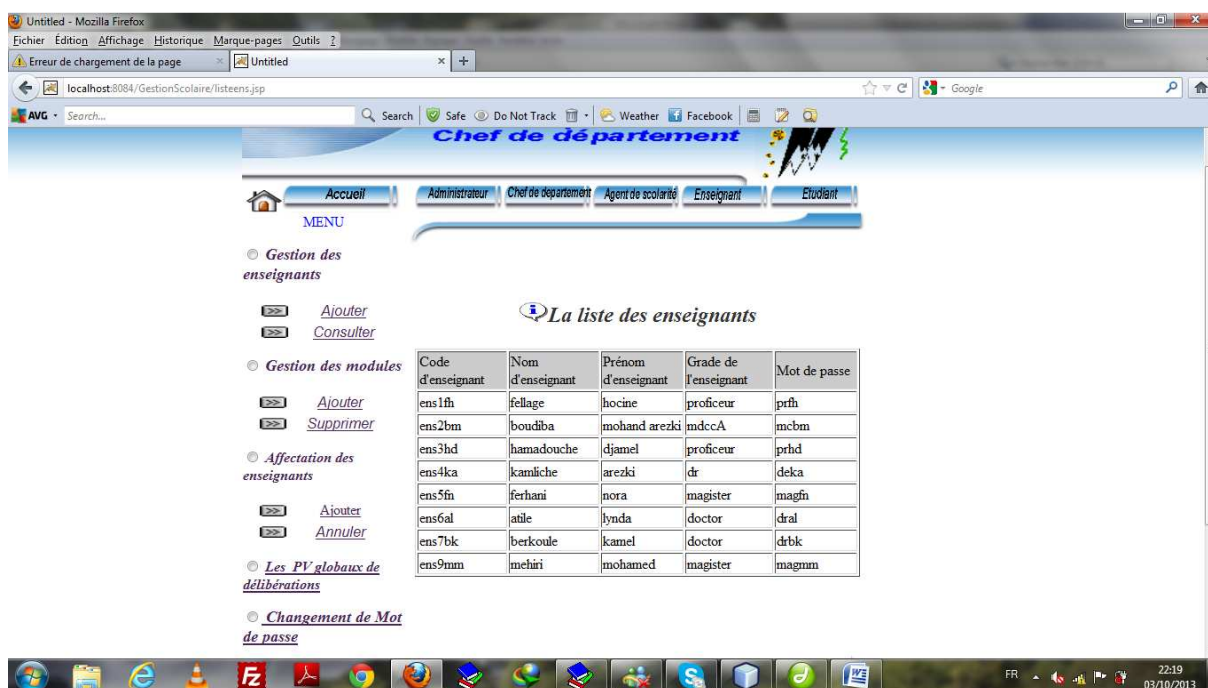


Figure (VI.15): Liste des enseignants

Conclusion :

Dans ce chapitre, nous avons présenté au premier lieu l'environnement de Développement de notre application et les outils utilisés pour implémenter celle-ci.

Puis nous avons présenté les différents langages de programmation utilisés. Ainsi que quelques fonctionnalités qu'elle effectue notre application.

Conclusion générale

Conclusion générale

L'objet de notre travail dans ce projet de fin d'études est d'assurer une partie importante et indispensable pour la mise en place d'une application web pour la gestion de scolarité de l'un des départements de l'université Mouloud Mammeri de Tizi Ouzou.

Notre problématique de départ était de répondre au mieux aux attentes des différents utilisateurs : l'étudiant, l'enseignant, l'administrateur, l'agent de scolarité, chef de département. Par ailleurs, il fallait faciliter l'accès à l'information et éviter les traitements manuels source d'erreurs importantes.

Et pour cela on a essayé d'apporter des solutions à cette problématique comme : la consultation des notes, l'enregistrement et la modification des notes, édition de relevé de notes, consulter les PV modulaire,...etc.

La réalisation de ce travail nous a permis d'acquérir de nouvelles connaissances sur le langage java et la plateforme JEE, Dreamweaver pour la création des interfaces et IDE Netbeans pour la programmation java (des servlets) des contrôles et les JSP, Le système de gestion de bases de données MYSQL server 5.4.

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

Annexes

I- Définition d'un réseau:

Un réseau est un ensemble de matériels et de logiciels permettant de connecter deux ou plusieurs ordinateurs géographiquement dispersés afin d'échanger des données. Les données circulent dans le réseau sous forme de flux de bits. Ces bits sont transférés sous forme de paquets de données qui se déplacent rapidement et qui traversent plusieurs réseaux pour arriver à la destination. Quelque soit la durée de déplacement, l'information doit circuler d'une manière fiable et atteindre sans erreurs la destination voulue.

II- Intérêts d'un réseau informatique :

- Le partage des ressources (fichiers, applications et matériels).
- La communication entre personnes (courrier électronique, discussion en direct).
- La communication entre processus.

III- Les supports de transmission :

Le support de transmission désigne le type du câblage utilisé pour relier l'ensemble des nœuds entre eux. Cet élément est d'une importance capitale car c'est de lui que dépendent, pour une très grande partie les performances du réseau. Les supports de transmission les plus utilisés sont :

III.1- Le câble coaxial :

Un câble coaxial est constitué par une âme centrale (support d'information) et d'un blindage contre les parasites et les inductions magnétiques. C'est le support de transmission le plus utilisé surtout lorsque les postes se trouvent en des endroits relativement éloignés les uns des autres (quelques dizaines de Kilomètres), mais il n'est plus tellement utilisé pour les petits réseaux car il est lourd, cher et peu flexible.

III.2- Le câble à paire torsadée :

Le câble à paire torsadée ou UTP (Unshielded Twisted Pair) est formé de paires de fils fins enroulés l'un autour de l'autre ne comportant aucun blindage ce qui le rend très sensible aux parasites et aux inductions magnétiques et réduit son utilisation à des connexions entre des postes assez proches les uns des autres (moins de 100 mètres : cas d'un immeuble) ce support est moins cher et plus fin.

III.3- La fibre optique :

La fibre optique est la toute dernière forme de câble de réseau la plus sûre et la plus rapide. Elle est composée d'une âme centrale en verre et elle transmet les signaux par la lumière plutôt que par l'électricité. Totalement insensible aux perturbations électromagnétiques, les réseaux en fibre optique restent cependant peu utilisés, essentiellement, en raison de leur coût relativement élevé et des difficultés de mise en place.

IV- Classification des réseaux :

On peut classer les réseaux en fonction de leurs distances et de leurs modes de fonctionnement.

A. Classification des réseaux en fonction de la distance :

On compte généralement trois catégories de réseaux informatiques, différenciées par la distance maximale entre les deux points les plus éloignés.

➤ **Les réseaux locaux (LAN) :**

Un réseau local est un ensemble d'ordinateurs interconnectés se trouvant dans une zone géographique relativement petite, appartenant à une même entreprise ou établissement, son étendue peut atteindre quelques centaines de mètres.

➤ **Les réseaux métropolitains (MAN) :**

Les réseaux métropolitains effectuent l'interconnexion de plusieurs sites d'une même ville. Ce sont des réseaux ayant une étendue de l'ordre d'une centaine de kilomètres et des débits très importants, ce qui permet ainsi à deux nœuds distants de se comporter comme s'ils faisaient partie du même réseau local.

➤ **Les réseaux étendus (WAN) :**

Les réseaux étendus ou longue distance interconnectent des réseaux de types précédents, sur des zones géographiques considérables, de taille d'un pays, d'un continent ou même toute la planète. C'est le cas, par exemple, du réseau Internet, d'extension planétaire et sur lequel sont connectées des millions de machines. Les WAN fonctionnent grâce à des routeurs qui déterminent le chemin optimal que l'information doit emprunter pour atteindre un autre nœud du réseau.

Remarque : on peut avoir des réseaux couvrant une surface inférieure à celle des réseaux locaux tels un ordinateur et ses périphériques ou un circuit imprimé.

B. Classification des réseaux en fonction du mode de communication :

On y distingue deux classes de réseaux :

- ✓ Ceux en mode de diffusion.
- ✓ Ceux en mode point à point.

Le premier mode de fonctionnement consiste à partager un seul support de transmission. Chaque message envoyé par un équipement sur le réseau est reçu par tous les autres. Un champ adresse placé dans le message permettra à chaque équipement de déterminer si le message lui est adressé ou non. À un moment donné, un seul équipement a le droit d'envoyer un message sur le support, il faut donc qu'il écoute au préalable si la voie est libre; si ce n'est pas le cas il attend, selon un protocole spécifique à chaque architecture.

Les réseaux locaux adoptent pour la plupart, le mode diffusion sur une architecture en bus ou en anneau et les réseaux satellitaires ou radio suivent également ce mode de communication.

Dans une telle configuration la rupture du support provoque l'arrêt du réseau, par contre la panne d'un des éléments ne provoque pas la panne globale du réseau.

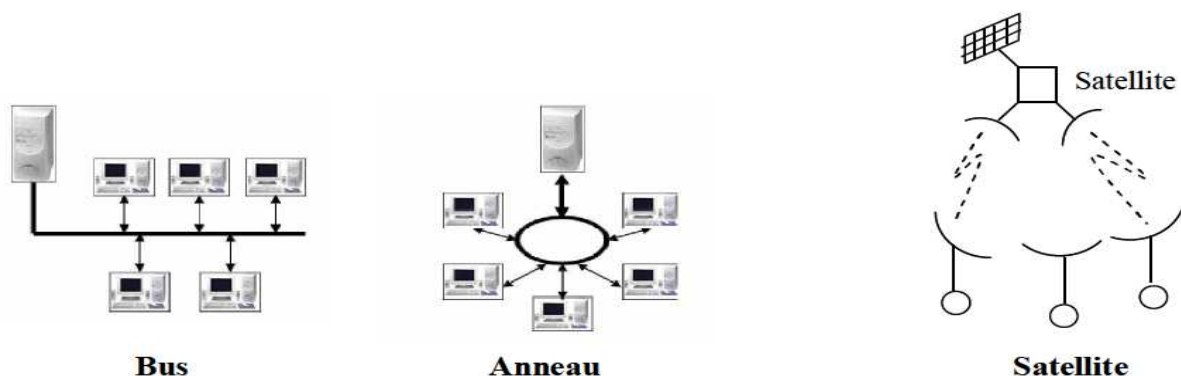


Figure I.1 : Mode de diffusion

Dans le deuxième mode, le mode point à point, chaque paire d'équipement est reliée par une ligne et si deux équipements veulent communiquer sans être reliés directement, ils le font indirectement par l'intermédiaire d'autres nœuds du réseau. Quand un message est envoyé d'une machine à une autre via un ou plusieurs nœuds intermédiaires, le message est reçu et stocké jusqu'à ce que la ligne de sortie requise soit libre. La plus part des réseaux de grandes étendues utilisent ce mode de communication.

La figure suivante montre quelques architectures utilisant le mode point à point :

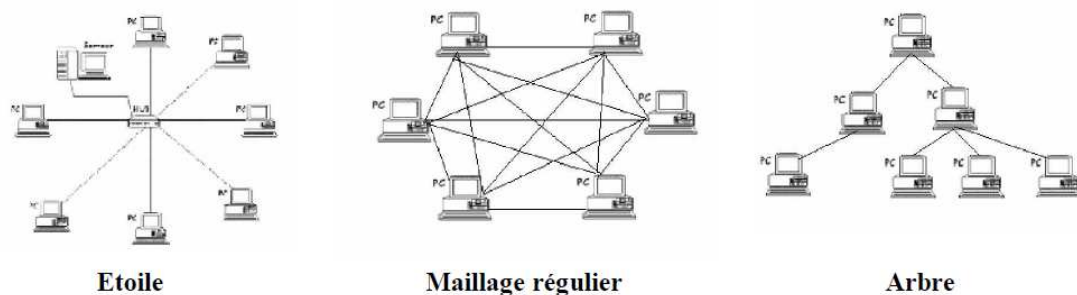


Figure I.2 : Mode point à point

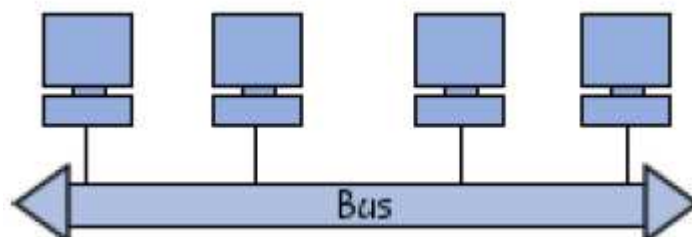
V- Les topologies : [8]

A. La topologie physique :

Un réseau informatique est constitué d'ordinateurs reliés entre eux grâce à des lignes de communication et des éléments matériels (cartes réseaux, ainsi que d'autres équipements permettant d'assurer la bonne circulation des données). L'arrangement physique, c'est-à-dire la configuration spatiale du réseau est appelée topologie physique. On distingue généralement les topologies suivantes :

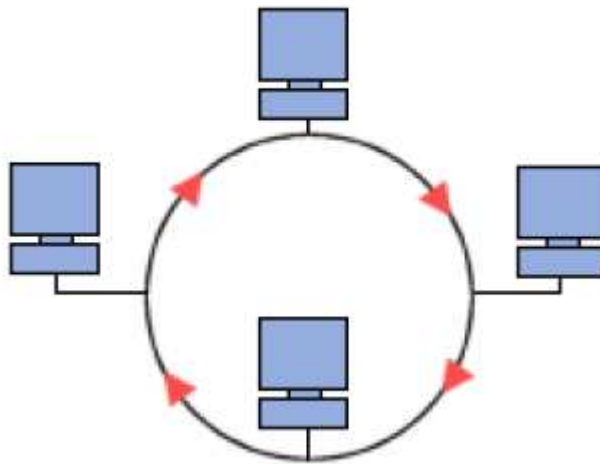
a- La topologie en bus :

Une topologie en bus est l'organisation la plus simple d'un réseau. En effet, dans une topologie

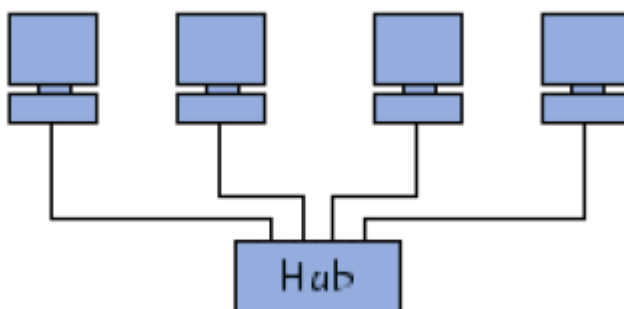


b- La topologie en anneau :

La topologie en anneau consiste à disposer les ordinateurs sous forme d'un anneau. En réalité, dans une topologie en anneau, les ordinateurs ne sont pas reliés en boucle, mais sont reliés à un répartiteur (appelé MAU : Multi-station Access Unit) qui gère la communication entre les ordinateurs. Ces derniers transmettent à tour de rôle par le biais d'un message spécial (jeton) que chaque ordinateur passe au suivant. A un moment donné seule la machine qui possède le jeton peut transmettre ses données. En bus tous les ordinateurs sont reliés par une liaison commune appelée « Bus » (segment central).

***c- la topologie en étoile :***

C'est la topologie la plus utilisée actuellement. Elle comporte un HUB auquel sont reliés les ordinateurs, ce dernier organise les échanges de données qui se font sur autant de lignes qu'il y a de nœuds. Avec cette topologie il est tout à fait possible de transmettre plusieurs messages en même temps sans qu'il y soit le moindre problème.



VI- Architecture des réseaux :

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.

VI.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 coucheinteragit 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 :

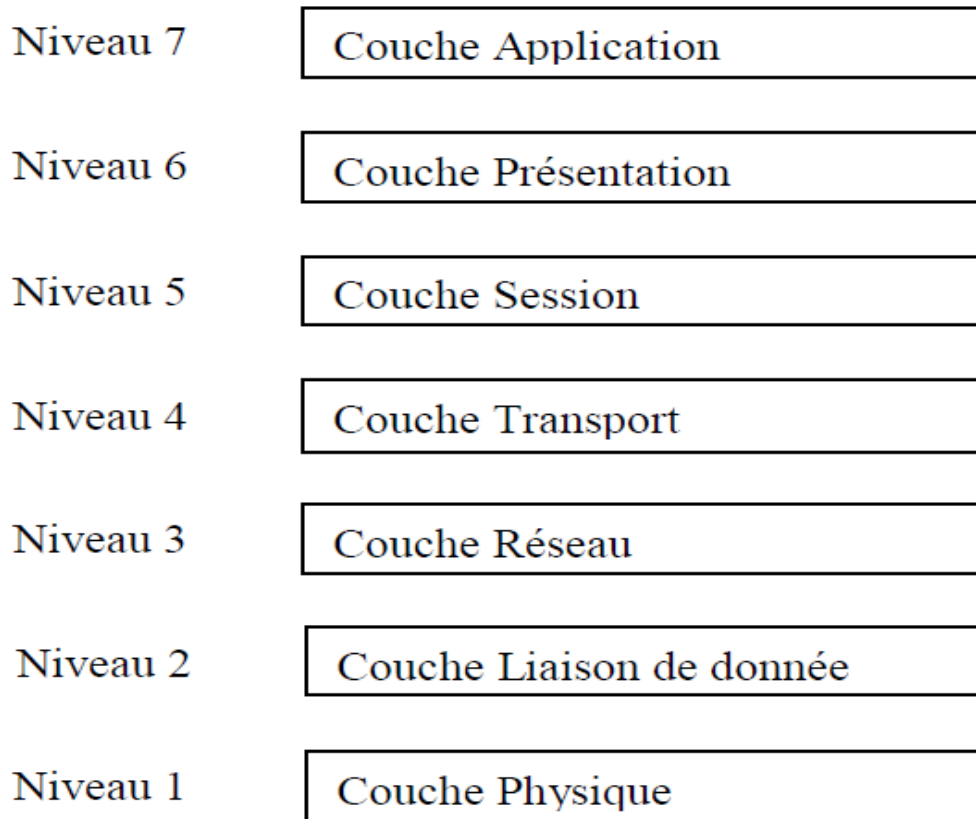


Figure : Le modèle OSI en 7 couches

➤ ***Couche physique :***

Décrit les propriétés physiques, telles que les caractéristiques mécaniques des différents supports et les caractéristiques des signaux qui véhiculent l'information [3], 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 brutes 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.

➤ ***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.

➤ **Couche session :**

Elle permet à des utilisateurs travaillant sur différentes machines d'établir des sessions entre eux.

➤ **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.

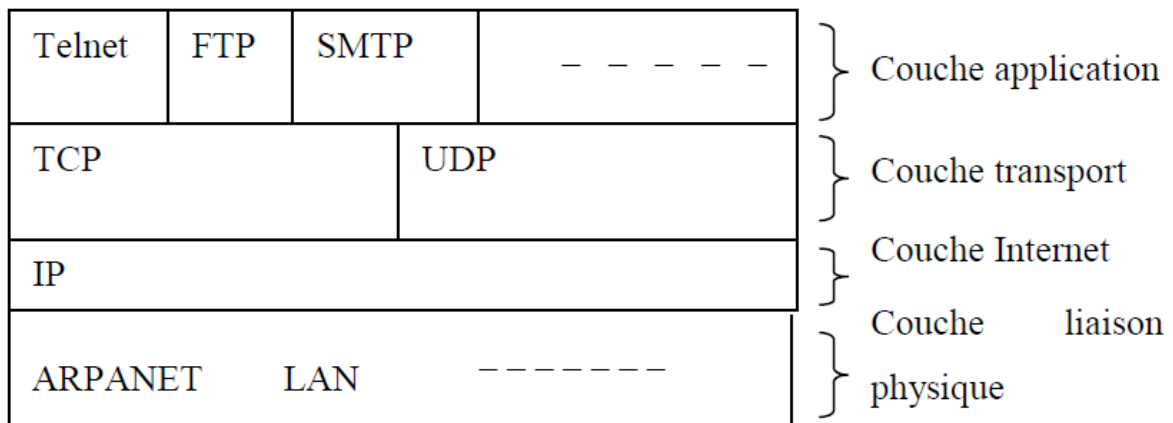
➤ **Couche application :**

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.

VI.2 Architecture TCP / IP :

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 :



IP: Internet Protocol : protocole de niveau réseau assurant un service sans connexion.

TCP : Transmission Control Protocol : niveau transport (niveau 4) qui fournit un service fiable en mode connecté.

UDP : User Datagram Protocol : niveau transport en mode non connecté.

FTP : File transfert Protocol : pour le transport de fichiers.

SMTP : Simple Mail Transfert Protocol : pour le transfert du courrier électronique.

Telnet : Protocol de gestion de Terminal Virtuel (permet d'obtenir les logiciels d'un autre ordinateur grâce au réseau).

1. Aperçu sur UML pour le Web :

UML (Unified Modeling Language) traduire « langage de modélisation unifié » est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, Booch ET OOSE. Issu « du terrain » et fruit d'un travail d'expert reconnus, UML est le résultat d'un large consensus. De très nombreux actes industriels de renom ont adopté UML et participent à son développement. Comme UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente. UML est une sorte de boîte à outils, qui permet d'améliorer progressivement les méthodes de travail, tout en préservant les modes de fonctionnement.

Intégrer UML par étapes dans un processus, de manière pragmatique, et tout à fait possible.

La faculté d'UML se fonde dans le processus courant, tout en véhiculant une démarche méthodologique, facilite son intégration et limite de nombreux risques (rejet des utilisateurs, coût...).

II. Définition de l'UML :

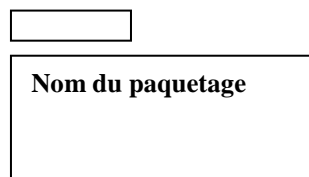
L'UML (Unified Modeling Language) est un langage qui se compose du vocabulaire, de la syntaxe et de la sémantique. De plus, c'est un langage de modélisation qui permet de représenter un système à différents niveaux: conceptuel, physique et se compose du vocabulaire et des règles pour décrire des différents modèles représentant un système.

Quelques définitions :

- **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....



- **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.



III. Les éléments de l'UML :

III.1 Les éléments de structure : Les éléments de structure sont les noms des modèles UML.

➤ Les classes :

Une classe est une description d'une collection d'objets qui partagent les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique. Graphiquement, une classe est représentée par un rectangle.

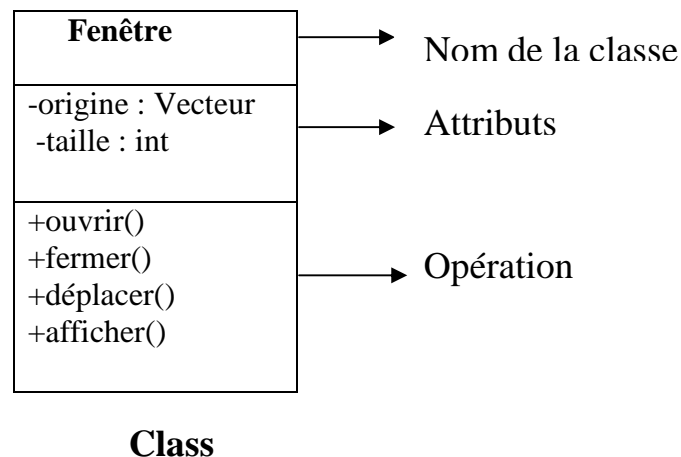


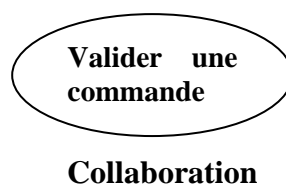
Figure A.1 : Schéma représentatif d'une classe

➤ Les interfaces :

Une interface est une collection de spécifications d'opérations qui définissent un service d'une classe ou d'une composante. Graphiquement, une interface est représentée par un cercle.

➤ Les collaborations :

Une collaboration définit une interaction entre éléments pour fournir un comportement coopératif. Graphiquement, une collaboration est représentée par une ellipse en pointillé.



➤ Les classes actives :

Une classe active est une classe dont les objets possèdent un ou plusieurs processus et peuvent initier des activités de contrôle.

Graphiquement, une classe active est représentée comme une classe, mais avec des lignes plus épaisses.

Exemple :

Tampon
-taille : int
+retirer()

Active Class

Figure A.4 : Schéma représentatif d'une classe active

III.2 Les éléments de dynamique :

➤ Les interactions :

Une interaction est un comportement qui comprend un ensemble de messages échangés entre objets. Elle peut être visualisée selon le point de vue du temps (diagramme de séquences) ou selon le point de vue de l'espace (diagramme de collaboration).

La figure suivante schématise une interaction visualisée selon le point de vue du temps :

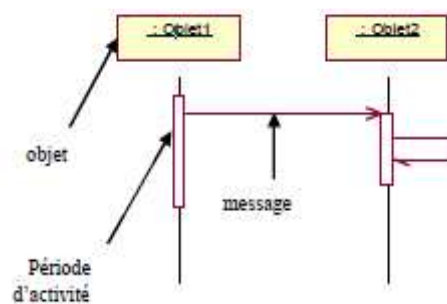


Figure A.6 : schéma représentatif d'une interaction

III.3 Les élément de regroupement : Ils sont les parties organisationnelles des modèles UML, ils comprennent les paquetages.

➤ Les paquetages :

Ils regroupent les trois types d'éléments précédents, ils sont purement conceptuels i-e qu'ils n'existent que lors de la phase de développement, Ils offrent un mécanisme général pour la partition des modèles et le regroupement des éléments.

Un paquetage est représenté en général par un dossier étiquette, et contient seulement son nom, mais parfois son contenu.

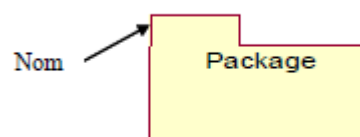


Figure A.7 : Schéma représentatif d'un paquetage.

IV. 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.

IV.1 Description des cas d'utilisation [17]

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.

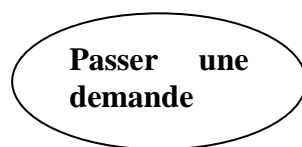
b. Le cas d'utilisation :

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.

Graphiquement, un cas d'utilisation est représenté par une ellipse.

Exemple :



Use Case

Figure A.3 : Schéma représentatif d'un cas d'utilisation.

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 :

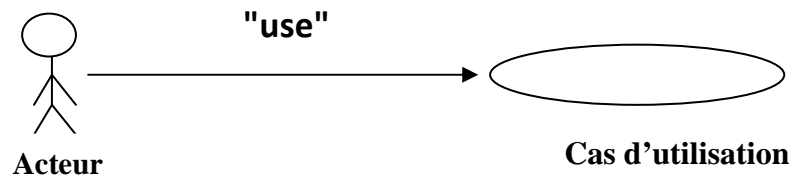
- la relation d'utilisation
- la relation d'extension

- 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.

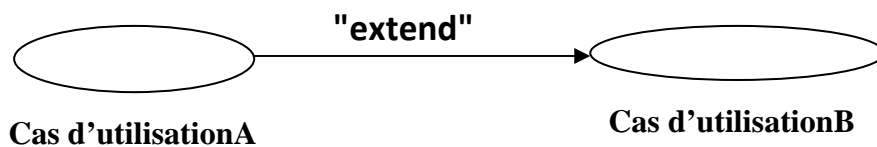
Formalisme :



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).

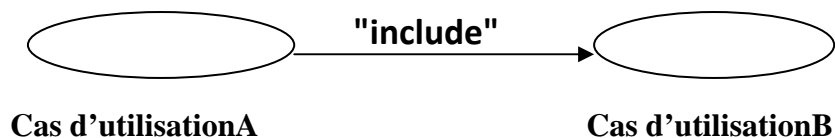
Formalisme :



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.

Formalisme :

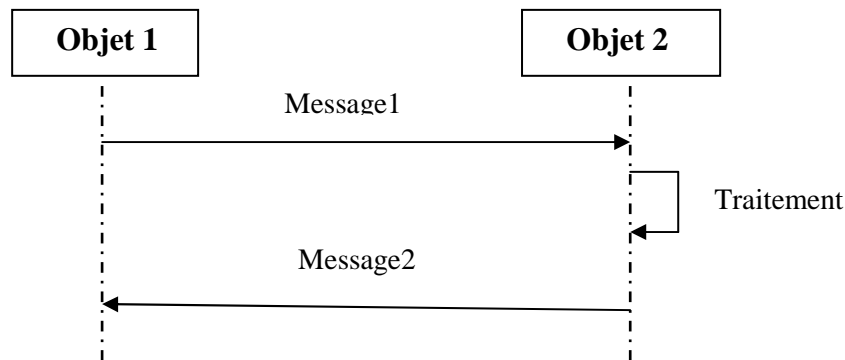


IV.2. Diagramme de séquence [16] :

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

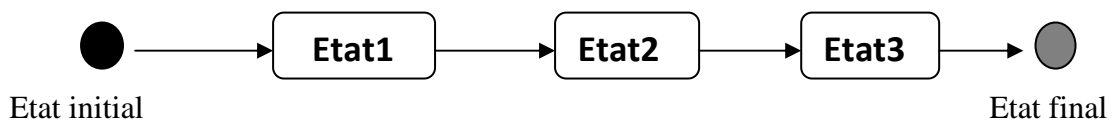
Formalisme :



IV.3 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 séquencement.

Formalisme :



IV.4. Les autres diagrammes UML :

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.

V. Extension d'UML pour le web [10]

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

V.1. Classe :**V.1.1 Page serveur « Server page »** **Icône :**

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

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

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

V.1.2. Page client « client page » **Icône :**

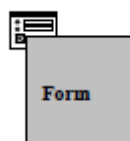
Description : Une instance d'une page client est une page Web formatée en HTML. Les pages clients peuvent contenir des scripts interprétés par les navigateurs lorsque celles-ci sont restituées par ces derniers. Les fonctions des pages clients correspondent aux fonctions des scripts de la page web.

Contrainte : Aucune.

Etiquettes : _ Titre (Title) : Titre de la page tel qu'il est affiché par le navigateur.

_ Base (Base) : URL de base pour déréférencer les URL relatives

Corps (Body) : ensemble des attributs de la balise <body>, qui définit des caractéristiques par défaut du texte et de l'arrière-plan.

V.1.3. Formulaire **Icône :**

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

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

Contraintes : Aucune.

Étiquettes : Méthodes utilisées pour soumettre les données à l'URL de l'attribut

V.2. Association

V.2.1. Lien « link »

Icône : Aucune

Description : Un lien est un pointeur d'une page client vers une autre page. Dans un diagramme de classe, un lien est une association entre une page client et une autre page client ou une page serveur.

Contraintes : Aucune.

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

V.2.2. Soumet « submit »

Icône : Aucune.

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

Contraintes : Aucune.

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

V.2.3. Construit « build »

Icône : Aucune.

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

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

Contraintes : Aucune.

V.2.4. Redirige « redirect »

Icône : Aucune

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

Contraintes : Aucune.

Etiquettes : Délai (delay) : délai que doit observer une page client avant de rediriger vers la page destination.

VI. Les points forts et points faible d’UML [16]

VI.1. Les points forts :

- a. UML est un langage formel et normalisé, gain de précision, gage de stabilité encourage l'utilisation d'outils
- b. UML est un support de communication performant, il cadre l'analyse, il facilite la compréhension de représentations abstraites complexes, son caractère polyvalent et sa souplesse font un langage universel.

VI.2. Les points faibles :

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

La nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue

Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard.

Bibliographie

La bibliographie

[01] : Architectures de Systèmes d'Information
Novembre 2002
Livre Blanc

Se trouve sur le site :

➤ www.dsi.cnrs.fr/methodes/gestion-projet/.../OCTO-architecture-des-SI.pdf

[02] : fr.wikipedia.org/wiki

[03] : George et Olivier Gardarin, « **Le client / serveur** », Edition EYROLLES 2000.

[04] : http://www.informatique-inside.com/index.php?page_client/serveur

[05] : Alin Lefebvre, « **L'architecture client-serveur** », Edition ARMAND COLIN 2006.

[06] : les architectures client/ serveur, interne et intranet des CGI aux ETB.
(A.cloux & Pierre-Yves)

[07] : Le modèle client-serveur Olivier Aubert

Se trouve sur le site :

➤ www.olivieraubert.net/cours/reseaux-iup/archi-client-serveur.pdf

[08] : Douglas Comer, « **Internet : Services et réseaux** », Edition DUNOD 2003.

[09] : [1] [http://www .Wikipédia.com](http://www.Wikipédia.com)

[10] : G. Gardarin, « **Internet/Intranet et les bases de données** », Edition Eyrolles 2000.

[11] document : pfe de J2EE, fichier PDF, qui se trouve sur

➤ fac-sciences-unv-batna.dz/cs/enseignants/guezouli.../projet_03.pdf

[12] document : introduction_a_la_plateforme_J2EE, fichier PDF qui se trouve sur

➤ www.mccours.net/cours/pdf/onfo/introduction_a_la_plateforme_J2EE.pdf

[13] livre : JAVA Autoformation

« Isabelle Valembois, Louis Millecam »
Ellipses 2001

[14] Site web: www.wikipédia.fr

La bibliographie

[15] document : J2EE vs Net Centre d'enseignements de grenoble 2008 /2009 qui se trouve sur

➤ membres-liglab.imag.fr/plumejeaud/NFE107.../J2EE%20vs%20NET.pdf

[16] Pascal Roques, « **UML par la pratique** », Edition EYROLLES 2eme édition 2004

[17] Grady Booch et al, « **Le guide de l'utilisation d'UML** », Edition EYROLLES 2003.

[18] Pierre-allain MULLER, "**Modélisation objet avec UML**", Edition : Eyrolles 1997.

[19] <http://fr.wikipedia.org/wiki/html>

[20] www.ccim.be/ccim328/js

[21] http://en.wikipedia.org/wiki/Adobe_Dreamweaver

[22] <http://www.mysql.com/>

[23] Les Cahiers du Programmeur Java 1.5 *Emmanuel Puybaret*

[24] "Servlets et Java Server page le guide du développeur", P.Y.Saument, A.Mirecourt.

[25]site : aonet.fr/.../IR3/...AcciR%20DDODT%20-%20Partie%20IDE.docx