

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

Université Saad Dahlab, Blida  
USDB.

Faculté des sciences.  
Département informatique.



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

Sujet :

**Conception et réalisation d'un  
portail Internet pour un modèle  
de gestion des enseignements**

**Présenté par :** M.Habbiche younes.  
M.Moumen samir.

**Promoteur :** M.Hadj yahia ouahid.  
**Encadreur :** M.Nehal Djilali.

**Organisme d'accueil :** Université Saad Dahlab, Blida  
USDB.



## Remerciements

Tous d'abord nous remercions *الله عز و جل* pour nous avoir guidé vers le bon chemin de la lumière et de savoir, pour nous avoir donné du courage et de la volonté afin de pouvoir réaliser ce modeste travail.

Nous exprimons nos sincères remerciements à nos parents pour tout ce qu'ils ont pu nous apporter.

Nous tenons à remercier notre promoteur M. Ouahid Hadj Yahia et co-promoteur Nehal Djilali pour leurs aides, leurs patiences, leurs disponibilités, et leurs compréhensibilités.

Nous tenons à remercier les membres du jury pour avoir eu l'obligeance d'accepter et d'apprécier notre travail.

A tous les enseignants de la faculté des sciences exacts de Blida et surtout les enseignants du département de l'informatique.

Enfin, nous remercions, de tout coeur, tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail



## Dédicace



*Ce travail est dédié :*

*A mes très chers parents pour leur soutien durant  
toute ma carrière,  
Pour leur bienveillance, leurs efforts constants dans mes études,  
Pour leur encouragement,*

*A tous mes sœurs Amina et Yasmine,*

*A toute la famille Habbiche et Kalache,*

*A mon binôme madjid et à toute sa famille,*

*A mon oncle et meilleur amis Krimou,*

*A tous mes amis que j'aime et qui m'aiment.*

*Younes*

*Ce travail est dédié :*

*A mes très chers parents pour leur soutien durant  
toute ma carrière,  
Pour leur bienveillance, leurs efforts constants dans mes études,  
Pour leur encouragement,*

*A toute la famille Moumen, Benhelal,*

*A mes frères Mohamed, Riadh et nabil*

*A ma sœur Sihem et ses enfants,*

*A mon binôme Younes et à toute sa famille,*

*A tous mes amis que j'aime et qui m'aiment.*

*Samir*

**Table des matières :**

Introduction générale.....	1
Problématique : .....	3
Présentation du sujet et ses objectifs : .....	3

**Chapitre I: La technologie Web et Wap.**

**Partie Web.**

1. Introduction : .....	5
2. Définition du WEB : .....	5
3. Qu'est-ce qu'un site web ?.....	5
4. Les Types des sites web : .....	5
4.1. Le Site statique : .....	5
4.2. Le site dynamique : .....	6
4.3. Le site semi-dynamique : .....	7
5. Site statique vs site dynamique : .....	7
6. L'architecture Client/Serveur : .....	8
6.1. Présentation de l'architecture d'un système client/serveur : .....	8
6.2. Fonctionnement d'un système client/serveur : .....	9
6.2.1. Architecture à deux niveaux : .....	9
6.2.2. Architecture à trois niveaux : .....	10
6.3. Comparaison des deux types d'architecture : .....	10
6.4. Avantages de l'architecture client/serveur : .....	11
6.5. Inconvénients du modèle client/serveur : .....	11
7. Le protocole HTTP : .....	11
7.1. Présentation : .....	11
7.2. GET et POST : .....	12
8. Les Serveurs Web : .....	12
9. Les navigateurs : .....	12

**Partie Wap.**

10. Présentation de la technologie WAP : .....	13
11. Les couches WAP : .....	14
11.1. La couche WAE (Wireless Application Environment) : .....	14
11.2. La couche WSP (Wireless Session Protocol) : .....	15
11.3. La couche WTP (Wireless Transport Protocol) : .....	15
11.4. La couche WTLS (Wireless Transport Layer Security): .....	15
11.5. La couche WDP (Wireless datagram Protocol) : .....	16
12. Le modèle WAP : .....	16
13. Architecture matérielle du protocole WAP : .....	17
13.1. Le serveur Web : .....	17
13.2. Les passerelles WAP (Geteway) : .....	18
13.3. Le réseau de l'opérateur mobile : .....	19
13.4. Les navigateurs WAP : .....	19
14. Fonctionnement d'un protocole WAP : .....	19
15. Les différents implémentations d'une plate forme WAP : .....	20
15.1- Passerelle WAP hébergée chez l'opérateur de téléphonie mobiles : .....	20
15.2-Passerelle WAP hébergée chez un fournisseur d'accès : .....	21
15.3- passerelle WAP en interne .....	22
16. Les avantages et les inconvénients de la technologie WAP : .....	22

**Chapitre II: les outils de développement.**

1. Les outils de développement web : .....	24
1.1. Langage HTML : .....	24
1.2. Quelques exemples de navigateur : .....	26

## Table des matières

2. Les outils de développement Wap : .....	27
2.1. Le langage WML : .....	27
2.2. La structure d'un document WAP : .....	29
2.3. Les éditeurs WAP : .....	29
2.4. Les convertisseurs d'images : .....	29
2.5. Les émulateurs WAP : .....	30
2.6. Les Kits de développement : .....	30
2.6.1. Nokia Internet mobile toolkit : .....	31
2.6.2. OpenWave SDK : .....	32
3. Base de donnée et SGBD : .....	32
3.1. Les limites à l'utilisation des fichiers : .....	33
3.2. Qu'est-ce qu'une base de données? .....	33
3.3. Utilité d'une base de données : .....	33
3.4. La gestion des bases de données : .....	34
3.5. Les différents modèles des SGBD : .....	36
3.6. Les caractéristiques d'un SGBD : .....	36
3.7. Les principaux SGBD .....	37
3.8. Comparaison entre bases de données relationnelles : .....	38
3.9. Les procédures stockées .....	38
3.10. Les déclencheurs (Trigger).....	39
4. Langage de programmation.....	39
4.1. Introduction à J2EE.....	40
4.2. Les API de J2EE.....	41
4.3. Les servlets java : .....	41
4.3.1. Qu'est-ce qu'une servlet?.....	42
4.3.2. Les différents types de moteur de servlets : .....	43
4.3.3. Interface d'une servlet : .....	43
4.3.4. Le cycle de vie : .....	46
4.3.5. Invoquer la servlet : .....	47
4.4. Java Server Page : .....	47
4.4.1. Qu'est-ce que Java Server Page ? .....	47
4.4.2. Types d'élément script JSP : .....	49
4.5. Les solutions de substitution .....	49
5. L'accès à la base de donnée : .....	49
5.1. ODBC (Open DataBase Connectivity):.....	49
5.1.1. Présentation: .....	50
5.1.2. Inconvénients de la technologie ODBC : .....	50
5.1.3. Qu'est-ce qu'un DSN ? .....	50
5.2. JDBC (Java DataBase Connectivity) : .....	50
5.2.1. Présentation : .....	51
5.2.2. Qu'est ce que JDBC ?.....	51
5.2.3. Drivers et gestionnaire de drivers : .....	52
5.2.4. Types de drivers : .....	53
5.2.5. Structure d'une application JDBC : .....	56
6. Serveur web et serveur d'application.....	56
6.1. Serveur web .....	56
6.1.1. Les besoins de sécurité : .....	57
6.1.2. Authentification des utilisateurs : .....	57
6.1.3. Confidentialité des informations échangées : .....	57
6.1.4. Journalisation des événements : .....	59
6.2. Serveurs d'applications.....	59
6.2.1. Présentation : .....	60
6.2.2. Rôle d'un serveur d'application : .....	61
6.2.3. Serveur web / serveur d'applications.....	61
6.2.4. Quelques serveurs d'application .....	61

6.2.5. Comparaison avec les serveurs propriétaires ..... 62

**Chapitre III: Le conduite d'une application Web.**

1. Introduction ..... 63

2. Le cycle de vie d'une application: ..... 63

3. La phase Etude : ..... 64

    3.1. Le cahier de charge : ..... 64

    3.2. Définir le périmètre fonctionnel en corrélation avec les objectifs poursuivis : ..... 64

    3.3. Définir le champ d'application : ..... 65

4. Conception : ..... 65

    4.1. Conception fonctionnelle : ..... 66

        4.1.1. Approche éditorial : ..... 66

        4.1.2 Créer le dossier de rubriquage : ..... 66

        4.1.3 Approche par processus : ..... 66

    4.2. Conception graphique : ..... 66

    4.3. Spécification fonctionnelles : ..... 67

    4.4. Conception technique : ..... 67

        4.4.1 Architecture : ..... 67

        4.4.2. Modélisation : ..... 68

    4.5. Spécification technique : ..... 70

    4.6. Tests utilisateurs : ..... 70

5. La réalisation ..... 70

    5.1. Production des contenus : ..... 71

        5.1.1. Contenus éditoriaux : ..... 71

        5.1.2. Contenus graphiques : ..... 71

    5.2. Développements : ..... 71

        5.2.1. Définir le plan de lotissement : ..... 71

        5.2.2. Développer les lots : ..... 72

        5.2.3. Réaliser les testes unitaires : ..... 72

        5.2.4. Opérer l'intégration : ..... 72

        5.2.5. Conduire les testes d'intégration : ..... 72

    5.3. Reprise de l'existant : ..... 72

    5.4. Recette de l'application : ..... 72

    5.5. Documentations : ..... 73

        5.5.1. Formaliser le manuel d'exploitation : ..... 73

        5.5.2. Formaliser le manuel utilisateurs : ..... 73

**Chapitre IV: Etude de l'existant.**

1. Flux d'information : ..... 77

2. Etude des postes de travail : ..... 78

3. Etudes des documents : ..... 81

4. Etude des procédures : ..... 84

5. Les règles de gestion: ..... 88

6. Codification existante : ..... 93

7. Critiques et suggestions : ..... 95

8. Solution informatique : ..... 96

9. Conclusion : ..... 98

**Chapitre V: Etude conceptuelle.**

1. Introduction : ..... 99

2. Dictionnaire de donnée : ..... 99

3. Codification proposée : ..... 102

4. Modèle conceptuel de données (M.C.D) : ..... 106

5. Modèle conceptuel des traitements (M.C.T) : ..... 114

## Table des matières

6. Modèle logique de données (M.L.D):.....	116
7. Modèle organisationnel des traitements (M.O.T) : .....	119
8. Le modèle physique de données (M.P.D):.....	122
9. Validation des vues externes: .....	122
9.1. Validation en consultation : .....	123
9.2. Validation en mise à jours : .....	128
10. Conclusion : .....	132
<b>Chapitre VI: Implémentation et réalisation.</b>	
1. Implémentation .....	133
1.1. Choix de l'environnement de développement .....	133
1.1.1. Côte serveur .....	133
1.1.2. Côte clients .....	135
2. Réalisation : .....	135
3. Sécurité de l'applicatif : .....	136
4. Présentation de l'application : .....	136
4.1. Partie I : Wap .....	136
4.1.1. Identification : .....	136
4.1.2. Affectation : .....	137
4.1.3. Notes : .....	138
4.2. Partie II : Web .....	138
L'espace 'Etudiant' : .....	140
L'espace 'Enseignant' : .....	141
L'espace 'Administrateurs' : .....	141
Conclusion générale .....	147
Bibliographie.....	148

# Introduction générale

Le gain de notre étude  
c'est en être devenu  
meilleur et plus sage.

*Michel Eyquem*

La multiplication des outils de production de l'information, qu'elle soit structurée (bases de données) ou semi-structurée (documents bureautiques par exemple) et la multiplication des espaces de stockage associés crée une masse d'informations difficilement exploitables pour l'entreprise. Ces informations qui représentent des sources communes qui doivent être partagées, la nécessité de partager ces dernières le plus efficacement et le plus rapidement possible est un enjeu considérable.

Les sources d'informations apparues progressivement pour répondre à des besoins spécifiques, n'ont pas été conçues dans la perspective d'un accès unique ; la diversité des formats de documents, les difficultés liées à l'interopérabilité des supports et la gestion de la sécurité afférente ont engendré une multitude d'interfaces d'accès à l'information. L'utilisateur ne dispose donc pas d'une interface de travail lui permettant un accès central à l'ensemble des données qui lui sont nécessaires. C'est dans cette optique que la notion de portail est apparue.

La vocation du portail est de proposer un point d'accès unique à un ensemble de contenus et d'applications à caractère analytique ou collaboratif en facilitant l'accès à des données structurées (base de donnée) et non structurées (documents textes) indépendamment de leur format et de leur localisation.

Il est essentiel d'intégrer le fait que la mise à disposition d'un point d'accès unique n'implique pas «bien au contraire» un stockage centralisé de l'information.

Le portail offre un point d'accès unique facilitant l'accès et la navigation dans l'information indépendamment de son origine et de son lieu de stockage.

L'information n'est pas répliquée vers un point central, mais accédée depuis une porte d'entrée unique. C'est à cette condition que l'utilisateur a la garantie d'avoir accès à la véritable information, en temps réel, et non à duplicata plus ou moins fiable construit pour l'occasion. En ce sens, le portail marque une rupture forte par rapport à des projets de ce type qui ont échoué en tentant l'impossible : l'information «par essence» est décentralisée et hormis des agrégations, doit rester proche de ses sources d'émission. Le portail centralise l'accès à l'information et non l'information elle-même.



La notion de portail est étroitement liée aux technologies Internet, car ce sont elles qui ont permis d'offrir une interface unique (le navigateur) pour accéder à l'information indépendamment de son lieu de stockage.

Aujourd'hui l'explosion des technologies nomades (ordinateur portable, téléphones mobiles WAP, etc...) souligne une caractéristique complémentaire du portail : le portail constitue un point d'accès unique mais il doit être accessible par un maximum d'interfaces : Le navigateur web bien sûr, mais aussi tous les types de terminaux couramment utilisés dans l'entreprise « a fortiori » les terminaux à caractère nomade, et ce, indépendamment de la localisation géographique de l'utilisateur (ce qui, nous y reviendrons, ne va pas sans poser des problèmes d'ergonomie et de sécurité).

Le portail doit offrir plusieurs modes d'accès d'un point de vue technique (diversité de ses supports) mais doit également permettre un accès indépendant de la position géographique de l'utilisateur. Il s'agit d'un point-clé : le portail d'entreprise fait voler en éclats la nécessité d'être géographiquement proche des applications et des contenus pour pouvoir y accéder: le portail offre à l'utilisateur un accès universel à son entreprise indépendamment du fait qu'il soit ou non au bureau.

## Problématique :

La gestion des directions des études et en particulier les services de scolarité demeure une priorité pour les responsables en charge des départements, facultés et services centraux au sein de l'université. La mesure de travail et les urgences auxquelles les responsables de ces services ont souvent à faire face rendent nécessaire l'utilisation de l'outil informatique. A cela s'ajoute l'importance d'avoir un accès rapide à une information fiable concernant les étudiants.

Au sein de l'université, Chaque faculté utilise son propre logiciel de gestion (son propre système d'information), si ce n'est pas chaque département dans plusieurs cas. De plus, la plupart du temps cela se résume en une gestion de scolarité et en monoposte. Ceci entraîne forcément une lourdeur dans les traitements, une surcharge de travail pour les personnes qui assurent la saisie et les éditions enfin des risques accrus d'erreurs de saisie. Ajoutons à cela une autre dimension : la mauvaise circulation de l'information (un doyen ou bien un vice-recteur devra contacter le responsable de la scolarité par téléphone et attendre l'édition d'un relevé de note d'un étudiant X s'il voudra consulter les résultats d'un quelconque étudiant surtout si celui-ci a fait plusieurs facultés). Un étudiant transféré d'une faculté à une autre ne conservera pas de traces des antécédents dans sa nouvelle affectation, etc..., pour les enseignants le problème se pose d'une manière plus pointue (plus forte) ...

Cela risque en plus d'empiéter sur la nouvelle stratégie de l'université si l'on considère le nouveau système d'enseignement LMD qui ouvre des passerelles aux étudiants à différents niveaux de la formation.

C'est pourquoi nous nous proposons d'exploiter l'Intranet qui est entrain de se concrétiser au sein de l'université reliant les différents pavillons afin de pallier aux différents handicaps cités.

## Présentation du sujet et ses objectifs :

A travers la problématique, il est clair qu'une solution globale et typique doit être présentée pour répondre aux besoins incessants des cadres et des responsables afin d'aboutir à une satisfaction totale et un progrès évident dans l'université "SAAD DAHLAB", et c'est exactement ce dernier point qui nous a poussé à travailler sur une solution qui s'intitule:

*«Conception et réalisation d'un portail Internet pour un modèle de gestion des enseignements».*

Il paraît après la première lecture de ce titre qu'il s'agit bien sûr de la conception et la réalisation d'un système d'information pour une gestion parfaite de la scolarité et aboutir à une base de donnée via le réseau intranet de l'université et Internet. Cela n'est possible que sous un portail Web qui sera paramétrable en fonction des diverses possibilités de consultation.

Par opposition au système classique, basé sur une technologie propriétaire, notre système sera bâti autour d'une architecture technique ouverte adaptable aux besoins et garantissant la pérennité du système informatique. A ce titre, les choix techniques retenus permettront d'atteindre les objectifs suivants :

- Développer un système information fiable et robuste.
- Disposer d'un environnement technologique ouvert et pérenne, basé sur les standards du marché et indépendant vis-à-vis des fournisseurs de technologies.
- Faciliter l'évolutivité du système à travers une réponse rapide à la croissance.
- Faciliter le développement de passerelles entre les différents systèmes informatiques, notamment pour accroître les échanges de données avec les partenaires de l'Administration.
- Permettre un accès plus large, à partir d'un portail Internet.
- Bénéficier des outils gratuits et Open Source.

Le système cible trois types de clients (**Client interne, Client Web et Client Mobil**) et repose sur une architecture trois tiers qui est un modèle d'architecture d'applications dans lequel on sépare la présentation, les traitements et les données. L'objectif poursuivi est de permettre une évolution de l'un de ces trois niveaux de façon relativement indépendante des deux autres.

### **Les différentes étapes du projet :**

#### **1-développement des applications pour les différents services et facultés de l'université.**

- Pédagogie : Facultés, V/R pédagogie, V/R PGRS
- Gestion des locaux : facultés, V/R planification
- Gestion Bibliothèque : Département, Facultés, Centrale
- Gestion des projets de recherche : PGRS facultés, V/R PGRS
- Gestion de Personnel : facultés, service central
- Gestion de la DFC

#### **2- Mise en Oeuvre des services INTRANET :**

- Serveur Web & Site de l'université
- Serveur de messagerie
- Moteur de recherche

#### **3- Migration vers le système Linux et les applications OPEN SOURCE**

# Chapitre I.

## La technologie Web et Wap

### La technologie Web

#### 1. Introduction :

L'invention du web, bien que récente, fait déjà partie de l'histoire de l'informatique. Ce chapitre n'a pas la vocation de faire un historique d'Internet. Cependant, rappeler les circonstances de la création de ce nouveau mode de communication, très déférente de son contexte d'utilisation aujourd'hui, permet de mieux comprendre ses caractéristique et ses composants. Il s'agit de rappeler quelques points essentiels.

#### 2. Définition du WEB :

Par rapport à la consultation papier, le web permet bien sûr un accès immédiat au document électronique et la consultation d'un même document par plusieurs personnes simultanément, il permet à l'utilisateur de déterminer lui-même son cheminement au travers de l'information.

#### 3. Qu'est-ce qu'un site web ?

Un site web (aussi appelé site internet ou page perso dans le cas d'un site internet à but personnel) est un ensemble de fichiers HTML stockés sur un ordinateur connecté en permanence à internet et hébergeant les pages web (on parle généralement de serveur car cet ordinateur possède un logiciel permettant de servir les pages web, il s'agit bien évidemment d'un serveur web).

#### 4. Les Types des sites web :

##### 4.1. Le Site statique :

C'est un site ou une partie de site non connecté à une base de données. Les pages y sont construites "en dur" c'est-à-dire qu'elles seront affichées sur le poste de l'utilisateur telles qu'elles ont été créées à l'origine et telles qu'elles ont été stockées sur le serveur.

Ces sites peuvent néanmoins contenir des éléments animés pour les rendre plus conviviaux (permutation d'images, images animées, animations Flash, DHTML ...) ainsi que des scripts (Java, Javascript, CGI...) pour accroître leurs fonctionnalités (formulaires notamment pour des demandes de renseignements).

Cette solution convient bien à des sites dont la taille n'excède pas la centaine de pages et dont les informations ne doivent pas être quotidiennement remises à jour. Les limites d'un site statique sont :

- une maintenance difficile due à l'obligation de modifier manuellement chacune des pages (notamment dans le cas où toutes les pages possèdent un même menu)
- l'impossibilité de renvoyer une page personnalisée selon le visiteur
- l'impossibilité de créer une page dynamiquement selon les entrées d'une base de données

C'est pourquoi il a été nécessaire de mettre au point une solution permettant de générer des pages Web du côté du serveur. Ainsi de nombreuses solutions mettant en oeuvre un langage de script sur le serveur ont été élaborées. Elles permettent de créer des pages Web dynamiques.

#### 4.2. Le site dynamique :

Le principe d'une page dynamique est d'être construite à la demande (à la volée) par le serveur (côté serveur), en fonction de critères spécifiques. La présentation et le contenu affichés peuvent ainsi être personnalisés de manière interactive, en fonction du contenu de la base de données, des internautes, des langues, etc.

Alors que les pages statiques font appel au HTML, langage de description de données, les pages dynamiques sont mises en oeuvre grâce à un langage de programmation. Ainsi, les pages HTML sont générées du côté serveur grâce à un script qui pioche les informations nécessaires dans une base de données. Le langage de programmation variera en fonction de la technologie retenue (PHP, ASP, Java, etc.).

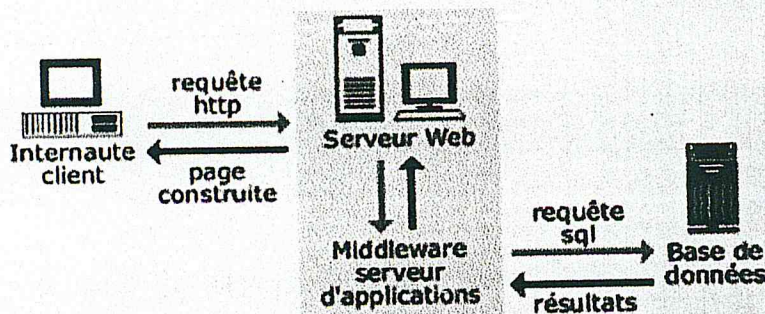


Figure I-1: Fonctionnement d'un site dynamique

Un site dynamique est plus exigeant pour un serveur Web qu'un site statique. En effet, alors que pour un site statique le serveur se contente d'afficher les pages, pour un site dynamique, il doit effectuer différents traitements et accéder à des bases de données. Pour chaque demande de l'utilisateur, il y a une interrogation au serveur qui construit la

page. Ainsi, lorsque qu'un site change peu dans le temps, il n'est pas nécessaire de créer un site dynamique lourd à mettre en œuvre mais un intermédiaire qui est le site semi-dynamique.

### 4.3. Le site semi-dynamique :

Pour un site semi-dynamique, les pages HTML sont générées automatiquement à partir d'une base de données en local. Ensuite, les pages HTML sont mises en ligne comme un site statique classique. Lorsque des modifications sont apportées à la base de données, on relance le programme pour produire une version actualisée des pages HTML.

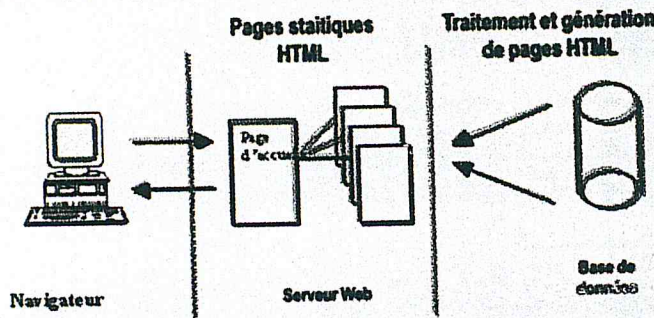


Figure I-2: Principe de génération d'un site semi-dynamique

### 5. Site statique vs site dynamique :

	Site statique	Site dynamique
Définition	Site web constitué de pages HTML prédéfinies, créées une fois pour toutes à l'aide d'un éditeur HTML. Le contenu des pages est fixe.	Site web dont les pages HTML se construisent lors de sa consultation par un internaute, en sollicitant des bases de données filtrées par des outils logiciels de mise en forme.
Evolutivité	Si la ligne éditoriale est amenée à évoluer rapidement, la solution statique est plus simple : il existe moins de séparation entre le fond et la forme. Il est aussi moins coûteux de faire évoluer un site statique vers l'accessibilité, sauf si la volumétrie est très importante.	Si la majorité des paramètres du site sont destinés à être pérennes, la solution dynamique est à privilégier. La solution est également préférée pour des sites à forte volumétrie ou intégrant de flux d'origines diverses.
Mises à jour	Ces mises à jour demandent un minimum de compétences techniques (maîtrise du langage de programmation et du webdesign). Un principe : plus un site statique est volumineux et plus sa mise à jour est fréquente, plus celle-ci sera lourde à gérer. Le site statique implique également un risque de dépendance forte vis-à-vis du webmaster (problème de transmission des compétences et de goulot d'étranglement lié à sa charge de travail).	Elles sont facilitées, tant sur le fond que sur la forme, grâce aux outils d'administration et à une meilleure séparation entre le fond et la forme. Ces mises à jour ne demandent pas de compétences techniques particulières et se font via un outil de gestion de contenus. La formule est notamment beaucoup plus adaptée en cas de contributeurs multiples (boucles de workflow possibles), et de mises à jour très fréquentes.

Fonctionnalités	Limitées en terme d'interactivité. Il est par exemple difficile d'utiliser des fonctionnalités liées à des bases de données.	Plus avancées grâce à l'utilisation de bases de données (moteur de recherche, interactivité, personnalisation...).
Référencement	Toutes les pages étant individualisées, il est plus facile d'obtenir un bon référencement. L'historique du site est également conservé dans sa globalité et en l'état sur le Net, ce qui facilite le travail de référencement sur la durée.	Le référencement des sites dynamiques ne nécessite pas forcément de recourir à des artifices tels que les pages alias ou les pages satellites. Il suffit que le référencement soit pris en compte dès la conception du site. Un bon référencement dépendra de la façon dont le site aura été développé.
Hébergement	Pas de contraintes particulières, les sites statiques représentent la forme élémentaire du Net à laquelle les hébergeurs sont habitués.	Plus de contraintes en théorie, car l'hébergeur doit permettre la mise en place de scripts et de bases de données correspondant aux technologies utilisées.
Prix de revient et coûts de maintenance	Le coût de développement initial est moins élevé, sauf si le site est particulièrement volumineux. Mais la mise à jour revient plus cher à la longue (recours à un prestataire si les compétences ne sont pas présentes en interne, et si elles le sont, le webmaster met plus de temps à réaliser les mises à jour qu'avec un outil de gestion de contenus).	Le coût de développement initial est en général plus élevé, sauf si le site est particulièrement volumineux. Mais ce surcoût est amorti en raison des économies réalisées sur la mise à jour. En revanche, une refonte plus globale que des simples mises à jour sera souvent plus coûteuse : elle nécessite de reparamétrer l'outil de gestion de contenus.

Tableau I.1 : Site statique vs site dynamique

## 6. L'architecture Client/Serveur :

### 6.1. Présentation de l'architecture d'un système client/serveur :

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, ...

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client FTP, client de messagerie, ..., lorsque l'on désigne un programme, tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès du serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client messagerie il s'agit de courrier électronique).[42]

Dans un environnement purement Client/serveur, les ordinateurs du réseau (les clients) ne peuvent voir que le serveur, c'est un des principaux atouts de ce modèle.

### 6.2. Fonctionnement d'un système client/serveur :

Un système client/serveur fonctionne selon le schéma suivant:

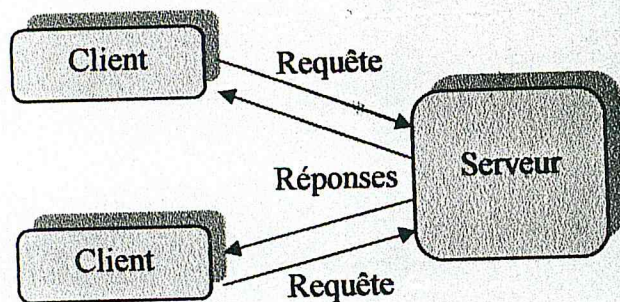


Figure I-3: Le modèle client/serveur

- 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

D'après l'architecture on peut distinguer deux types des applications client-serveur :

#### 6.2.1. Architecture à deux niveaux :

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

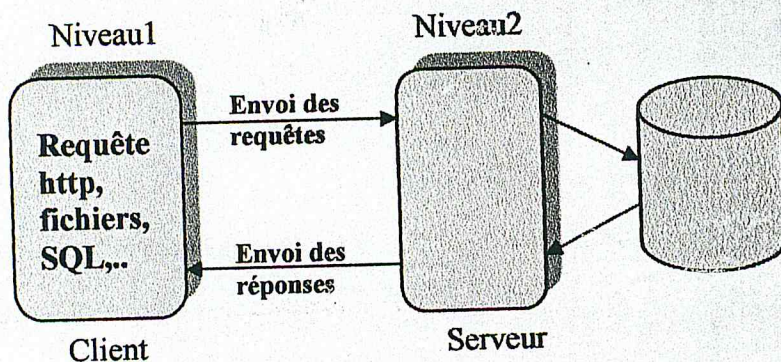


Figure I-4: Architecture client-serveur à deux niveaux

Cette architecture est très limitée, elle nous permet pas d'un traitement parfait des données. Cette solution est parfaite si nos traitements se limitent à une simple validation des données, si le traitement est plus compliqué le client doit être plus complexe qui nous donne



un client lourd ce qui fait que la machine cliente doit être plus puissante ce qui n'est pas toujours possible.[06]

Pour résoudre ces problèmes on propose l'architecture trois-tiers.

### 6.2.2. Architecture à trois niveaux :

Dans l'architecture à 3 niveaux (appelées architecture 3-tier), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre:

1. Le client: le demandeur de ressources
2. Le serveur d'application (appelé aussi middleware): le serveur chargé de fournir la ressource mais faisant appel à un autre serveur.
3. Le serveur secondaire (généralement un serveur de base de données), fournissant un service au premier serveur.

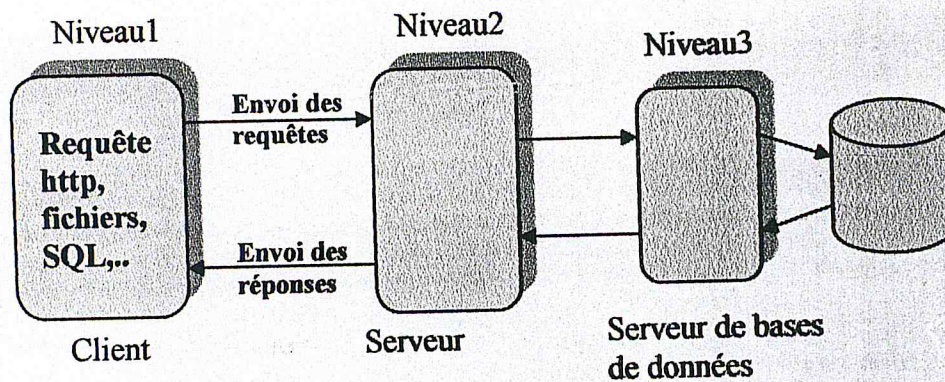


Figure I-5: Architecture des applications à trois niveaux

Cette architecture assure :

- Une connectivité à la base de données isolée : seul le serveur qui accède à la base de données, le client ne communique que avec une application.
- Le traitement métier centralisés : le serveur d'application prend en charge les problèmes de traitement des données qui n'ont place dans les autres tiers.[06]

### 6.3. Comparaison des deux types d'architecture :

L'architecture à deux niveaux est donc une architecture client/serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble des ressources demandées par le client. Dans l'architecture à trois niveaux par contre, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web/serveur de base de données par exemple). Ainsi, l'architecture à trois niveaux permet:

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

#### 6.4. Avantages de l'architecture client/serveur :

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

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

#### 6.5. Inconvénients du modèle client/serveur :

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

- un coût élevé dû à la technicité du serveur.
- une maillon faible: le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui! Heureusement, le serveur a une grande tolérance aux pannes (notamment grâce au système RAID).

### 7. Un protocole HTTP :

Lorsque deux applications communiquent, celles-ci doivent d'abord se mettre d'accord sur un mode de transfert des informations échangées. Le code décrivant la façon dont s'effectue le transport est appelé « protocole ».

#### 7.1. Présentation :

Les protocoles informatiques sont classés en différents niveaux qui se superposent. Un protocole de bas niveau étant proche du transfert physique des données, tandis qu'un protocole de haut niveau est proche de la représentation des données dans les applications qui les utilisent.

Le protocole du Web est dénommé HTTP (*Hyper Text Transfer Protocole*). Ce protocole permet l'encapsulation des données qui transitent entre le navigateur et le serveur web : une requête HTTP est envoyée par le navigateur au serveur, qui renvoie une réponse HTTP. Ce protocole permet au serveur Web de transmettre les informations à afficher par le navigateur. Il véhicule également, lors des requêtes, un certain nombre de renseignements concernant le poste client : les MIME supportés, le système d'exploitation, type de navigateur, ... etc.

HTTP s'appuie sur un autre protocole de plus bas niveau nommé TCP/IP : ce protocole assure la transmission des données après leur découpage en paquets indépendant. Ces paquets ou datagrammes sont réassemblés dans le bon ordre à la réception. Il a également l'avantage d'être disponible pour pratiquement tous les types de réseaux physique (Ethernet, ATM, Token Ring, ... etc), et tous les systèmes d'exploitation. La communication TCP/IP

entre deux applications distantes s'effectue de manière connectée sur un numéro de port (canal de communication virtuel entre deux machines) bien définie : on parle de sockets connectés.

A fin de garantir de bonne performance et une montée en charge facile, HTTP fonctionne en mode déconnecté : après chaque échange d'information, la connexion entre le client et le serveur est rompue. Le serveur Web ainsi un très grand nombre de clients, car ceux-ci ont peu de chance d'être connectés simultanément.

Ce mode de fonctionnement a pourtant ses défauts. L'établissement d'une connexion est un processus coûteux en temps. Or, par exemple, une page Web référencant quatre images nécessite cinq connexions et déconnexions : une pour la page elle-même et une pour chaque image ! Cette caractéristique se traduit par une augmentation de la charge du serveur. La multiplication des séries de requête/réponse accroît aussi la consommation de bande passante.

La version 1.1 de HTTP permet de regrouper le chargement de toutes les ressources d'une même page dans une seule étape de connexion/déconnexion.

HTTP 1.1, par ailleurs totalement compatible avec HTTP 1.0, est aujourd'hui supporté par tous les navigateurs Web du marché. L'optimisation apportée par HTTP 1.1 ne rend pas en cause le principe d'un fonctionnement déconnecté propre au Web. [34]

## 7.2. GET et POST :

Lorsqu'un client se connecte à un serveur et envoie une requête HTTP, cette requête peut être de plusieurs types différents, la référence se réside dans les méthodes appelées. Les méthodes les plus fréquemment utilisées sont GET et POST. Pour simplifier, la méthode GET est conçue pour obtenir des informations (document, graphique, ou résultat d'une requête sur une base de données), tandis que la méthode POST est conçue pour poster des informations (un numéro de carte de crédit, un matricule, ou des données à stocker dans une base de données, ...etc.). [15]

## 8. Les Serveurs Web :

Un serveur est un ordinateur qui propose un certain nombre de services. Un service est un programme qui fonctionne en permanence (démon) dès que l'ordinateur est sous tension. Il est à l'écoute également en permanence du réseau pour capturer la moindre requête le concernant venant de l'extérieur, et dès lors, il va rendre le service demandé. Le service Web est un service comme un autre, mais il est spécialisé par les requêtes propres au protocole http, et de plus, un répertoire lui est dédié pour permettre l'accessibilité depuis l'extérieur. Par contre, ce répertoire est fortement protégé, pour éviter toute malversation venant de l'extérieur. Il est également possible d'utiliser un serveur Web uniquement pour le réseau local sans être accessible depuis l'extérieur. Dans cette configuration, on appelle cela un Intranet. [06]

## 9. Les navigateurs :

Un navigateur, appelé aussi browser en anglais ou bien encore fureteur en québécois est un type de logiciel utilisé pour naviguer sur les réseaux informatiques et dans leurs bases de données, en particulier sur l'Internet.

Le navigateur sert d'interprète entre les machines qui communiquent des fichiers texte entre elles sur le réseau. C'est lui qui transforme les demandes d'un utilisateur en langage compréhensible par un ordinateur afin que la demande parvienne à bon port et soit traitée. C'est encore lui qui transforme l'information obtenue en retour en un format lisible par l'utilisateur.

Le navigateur est l'outil de l'internaute, Il s'agit d'un logiciel possédant une interface graphique. Il ne connaît que les langages html et javascript, si les pages demandées au serveur distant sont rédigées dans un autre langage (c'est le cas pour php,jsp.asp,...), c'est le serveur en question (ici le serveur sur lequel est hébergé notre site) qui se chargera de les traduire et de les envoyer en html ou en javascript à l'ordinateur client.[06]

## La technologie WAP

Depuis quelques années, on a pu assister à une véritable révolution du monde des télécommunications, En effet l'ordinateur personnel n'est plus le seul type de poste de client Internet disponible. L'explosion de la téléphonie mobile commence à bouleverser le marché des terminaux Internet.

Le nouvel eldorado du Web se nomme aujourd'hui WAP (Wirless Application Protocol). Le Wap est né de l'alliance en 1997 de plusieurs grands groupes regroupent les constructeurs des mobiles (Nokia, Ericsson, Motorola, ...). La version 1.0 du protocole WAP a été publié en 30 avril 1998.

L'objectif de WAP est de fournir une alternative au modèle actuel qui requiert une puissance de traitement et un débit réseau inaccessible aux terminaux mobiles.

### 10. Présentation de la technologie WAP :

La technologie WAP a pour but de permettre à des terminaux mobiles (les téléphones portables par exemple) d'accéder à des documents circulant par des réseaux sans fil. Il s'agit donc de permettre à n'importe quel terminal mobile de pouvoir formater des documents. C'est pour cela qu'un protocole universel a été mis en place: le **WAP** (*Wireless Application Protocol*). Il se propose définir la façon de laquelle les terminaux mobiles accèdent à des services Internet, et cela à un niveau au-dessus de la transmission des données, celle-ci étant spécifique à chaque opérateur de téléphonie. Le WAP définit aussi la manière de laquelle doivent être structurés les documents, grâce à un langage dérivant du HTML et nommé pour l'occasion **WML** (*Wireless Markup Language*) et un langage de script baptisé *WMLScript*.

En effet, étant donné l'émergence des réseaux sans fil, il était nécessaire de mettre en place des services Internet "nomades". Toutefois, étant donné les restrictions engendrées par:

- le réseau (bande passante réduite)
- le terminal (écran réduit, mémoire en petite quantité, de faibles capacités en terme de processeur, autonomie restreinte)

Il était nécessaire de mettre au point un protocole spécifique à ces terminaux d'un nouveau type.[06]

## 11. Les couches WAP :

Le protocole WAP est défini selon des couches, c'est-à-dire des niveaux d'abstraction des données (dans l'esprit du modèle OSI) afin de séparer les différents traitements des données nécessaires pour effectuer la transaction. Cela donne la possibilité d'implémenter (c'est-à-dire en quelque sorte de programmer avec un langage spécifique) différemment chacune de ces couches. Ainsi, chacune de ces couches définit une interface vis-à-vis de la couche suivante, ce qui permet une grande modularité des services, donc une évolutivité possible de chacune des implémentations.

Le protocole WAP est scindé en cinq couches [06]:

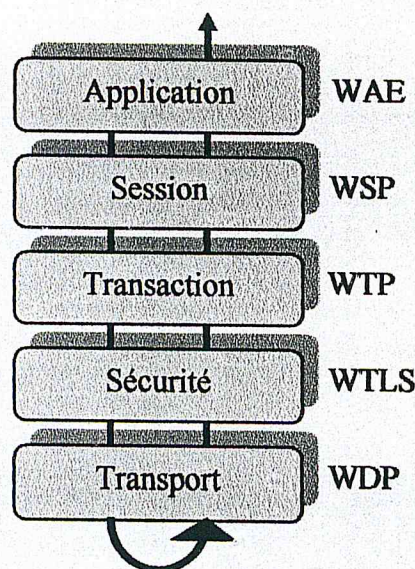


Figure I.6 : Les couches du protocole WAP

- La couche WAE (*Wireless Application Environment*) est la couche application du WAP.
- La couche WSP (*Wireless Session Protocol*) est la couche session du WAP.
- La couche WTP (*Wireless Transaction Protocol*) est la couche transaction du WAP.
- La couche WTLS (*Wireless Transport Layer Security*) est la couche sécurité du WAP.
- La couche WDP (*Wireless Datagram Protocol*) est la couche transport du WAP.

### 11.1. La couche WAE (Wireless Application Environment) :

La couche application du WAP définit l'environnement de développement des applications sur les terminaux mobiles. Elle fournit ainsi des fonctionnalités applicatives telles que:

- Le WML (*Wireless Markup Language*, que l'on peut traduire par *Langage à balises pour le monde sans fil*).
- le WMLScript: un langage de script interprété par le terminal mobile, dérivé de l'ECMAScript à l'image du Javascript (il n'est toutefois pas compatible), il permet d'exécuter de petits scripts contenant des instructions simples telles que des boucles, des structures conditionnelles, des fonctions, ...

- le WTA (*Wireless Telephony Applications*) un ensemble d'interfaces prédéfinies servant à créer des applications téléphoniques [07]

### 11.2. La couche WSP (Wireless Session Protocol) :

La couche session est constituée de deux protocoles:

- un protocole orienté connexion agissant au-dessus de la couche transaction
- un protocole non orienté connexion agissant au-dessus de la couche transport

La présence de ces deux protocoles permet de bénéficier soit de longues sessions sans acquittement, dans laquelle la communication peut être suspendue puis reprise, ou bien de sessions initiées par le serveur (technologie *PUSH*). [06]

### 11.3. La couche WTP (Wireless Transport Protocol) :

La couche de transaction gère le déroulement de la transaction, elle définit donc la fiabilité du service. La communication peut se faire de trois façons, c'est-à-dire:

- à sens unique avec acquittement
- à sens unique sans acquittement
- en full duplex avec acquittement

Elle permet en outre d'effectuer des transactions synchrones et de retarder les acquittements afin de les gérer par paquets.

### 11.4. La couche WTLS (Wireless Transport Layer Security):

Etant donné que les données circulent entre le terminal mobile et la passerelle grâce à des réseaux sans fil, il est nécessaire que les transactions soient sécurisées, c'est ce que se propose de faire la couche sécurité. Celle-ci est basée sur le standard SSL (*Secure Socket Layer*) et permet:

- de crypter les échanges de données
- de garantir l'intégrité des données (vérifier que celles-ci n'ont pas été modifiées)
- d'authentifier les acteurs de l'échange

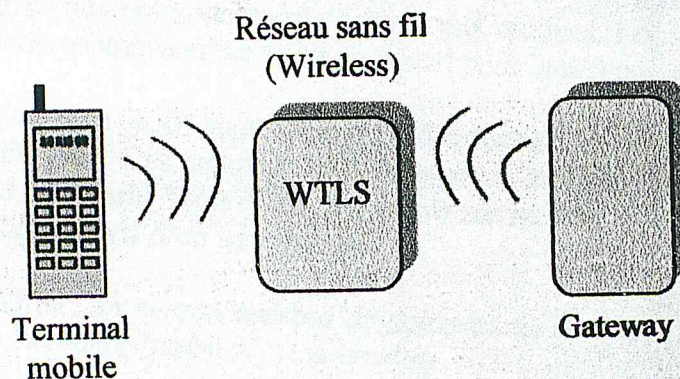


Figure I.7 : La couche WTLS

### 13. Architecture matérielle du protocole WAP :

Les architectures WAP reposent sur quatre briques technologiques, chacune étant nécessaire pour garantir l'acheminement d'un message par le protocole WAP :

- La mise en place d'un serveur Web et (ou) applicatif disposant de contenu au format WAP (ou plus précisément WML).
- L'intégration d'une passerelle ou d'un serveur WAP.
- Le réseau de l'opérateur.
- L'utilisation par le client d'un terminal WAP c'est à dire qui héberge un navigateur WAP.

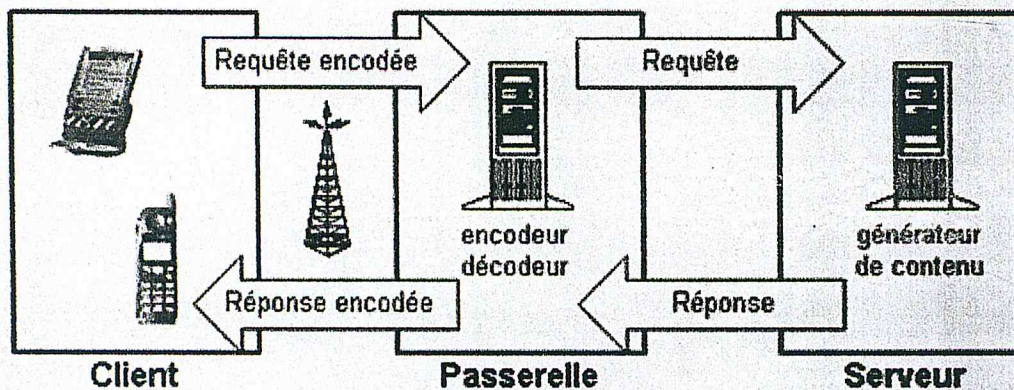


Figure I.8 : L'architecture WAP

#### 13.1. Le serveur Web :

Celui-ci a pour but de fournir des documents WML. Les technologies mises en œuvre sont identiques aux techniques Internet. C'est donc un serveur HTTP qui diffuse un contenu statique ou dynamique. La seule différence avec Internet est les formats de diffusion des pages en WML, et non plus en HTML.

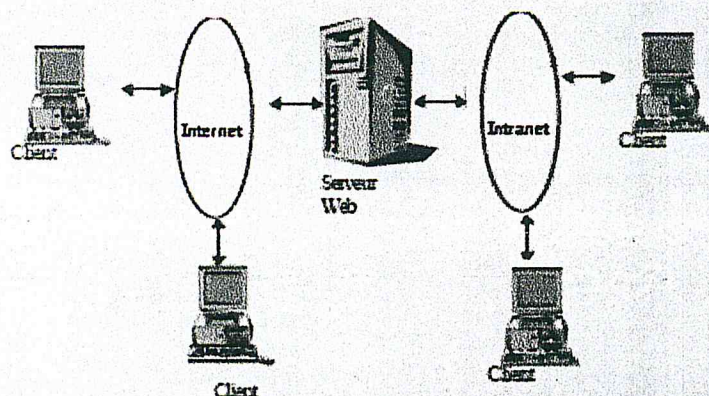


Figure L.9 : Le Server Web

Puisque les réseaux sans fil n'ont au début du millénaire qu'une bande passante réduite, la couche sécurité a été prévue pour optimiser la taille des données sécurisées.

### 11.5. La couche WDP (Wireless datagram Protocol) :

La couche WDP est à la base de la pile de protocoles WAP, c'est elle qui est chargé de l'interface avec les protocoles de transmission de données utilisés par les opérateurs de télécoms:

- GSM data
- HSCSD
- GPRS
- UMTS

## 12. Le modèle WAP :

Le modèle WAP est similaire à celui du World Wide Web. Ceci apporte un grand nombre d'avantage aux développeurs d'applications, en leurs permettant de réutiliser leurs outils et leurs modèles de programmation sur une architecture connue et stable. Des extensions, comme les langages WML et WMLScript, permettent de prendre en charge les caractéristiques spécifiques aux environnements des réseaux mobiles. Les standards existants ont été autant que possibles utilisés comme point de départ pour la technologie WAP.

Les applications et le contenu WAP sont écrits dans des formats similaires aux formats en usage sur le World Wide Web. Ces contenus sont transportés en utilisant les protocoles de communication standard du World Wide Web. Un micro navigateur ans le terminal mobile gère l'interface utilisateur vis-à-vis d'un navigateur Web.

L'architecture WAP définit un certain nombre de composants standard qui permettent la communication entre les terminaux mobiles et les serveurs Web :

- Un modèle de la nommage standard : utilisant les URLs du World Wide Web pour localiser les contenus et les applications WAP.
- Un type spécifique a été attribué à tous les types de contenus WAP : Ceci permet aux navigateurs WAP de gérer correctement les contenus reçus en se basant sur leur types.
- Un jeu de protocole de communication standard : permet aux périphérique WAP de communiquer avec un serveur Web de réseau.
- Ainsi le WAP met en œuvre une passerelle permettent d'interconnecter le mode de réseau mobile avec celui du World Wide Web. Cette passerelle comprend généralement les fonctionnalités suivantes :
- Une passerelle de protocoles traduit les requêtes en provenance de la pile de Protocole WAP (WSP, WTP, WTLS, et WDP) vers la pile de protocoles World Wide Web (http et TCP/IP).
- Un encodeur/décodeur de contenus traduit le contenu WAP en un format codé compact afin de réduire la taille de datas au travers de réseau mobile.

Cette infrastructure assure aux utilisateurs des terminaux mobiles de pouvoir utiliser une grande variété d'application et de contenus.[03]



### 13.2. Les passerelles WAP (Gateway) :

La passerelle a dans un premier temps un rôle d'interface entre le mobile fonctionnant sur un réseau sans fil, et le réseau IP, fonctionnant sur un support quelconque. Toutefois le rôle de la gateway ne s'arrête pas là. En effet, celle-ci permet de transformer les réponses en provenance du serveur applicatif en données binaires compactées, donc beaucoup plus adaptées à transiter sur le réseau sans fil de bande passante plus faible. Lorsque le terminal mobile reçoit ces données compressées, il les décode à l'aide d'un circuit prévu à cet effet.[03]. Il existe plusieurs passerelle WAP dans le marché permis eux :

#### Audicode WAP Server 2.0

Créé par Audicode

Site Web [www.audicode.com/eng/dow.html](http://www.audicode.com/eng/dow.html)

Pour Microsoft Windows 95/98/NT4/2000.

#### Ericsson's WAP Gateway/Proxy

Créé par Ericsson

Site Web [www.ericsson.com/WAP/products/infrastructure.shtml](http://www.ericsson.com/WAP/products/infrastructure.shtml)

#### Kannel Open Source WAP and SMS Gateway

Créé par Kannel

Site Web [www.kannel.org](http://www.kannel.org)

Kannel est une passerelle WAP en open source WAP gateway.

#### Motorola's WAP Server

Créé par Motorola

Site Web [www.motorola.com/MIMS/ISG/wap/sales.htm](http://www.motorola.com/MIMS/ISG/wap/sales.htm)

#### Nokia WAP Server 1.0

Créé par Nokia

Site Web [www.nokia.com/corporate/wap](http://www.nokia.com/corporate/wap)

Nokia WAP Gateway pour les opérateurs mobile.

#### UP.Link Server Suite

Créé par Apion/Phone.com

Site Web [www.phone.com/products/uplink.html](http://www.phone.com/products/uplink.html)

Phone.com Gateway pour les opérateurs mobile.

### 13.3. Le réseau de l'opérateur mobile :

L'opérateur fournit les services de transmission radio nécessaire au fonctionnement des mobiles. Les normes de transmission des données dans les réseaux de téléphonie mobile ont connu déjà plusieurs générations :

**1<sup>ère</sup> génération : système analogiques**

- AMPS (Advanced Mobile Phone system) 1983.
- TACS (Total Access Communication System) 1985.
- NMT (Nordic Mobile Telephone).

**2<sup>ème</sup> génération :**

- GSM (Global System for Mobile Communication) 1991 → 9.5 Kbit/s.
- CDMA (Code Division Multiple Access).

**2<sup>ème</sup> génération vers 3<sup>ème</sup> génération :**

- GPRS (General Packet Radio Service) 2000 → 171.2 Kbit/s.
- EDGE (Enhanced Data for GSM Evolution) 2001 → 473.6 Kbit/s.

**3<sup>ème</sup> génération :**

- UMTS (Universal Mobile Telecommunication System) 2002 → jusqu'à 2Mb/s

**13.4. Les navigateurs WAP :**

Le navigateur se charge de décoder les informations transmises par le passerelle WAP afin d'afficher les pages pour l'utilisateur, il joue le même rôle qu'un navigateur Web telle qu'Internet Explorer ou Netscape navigateur.

Il existe plusieurs navigateurs de téléphones mobile sur le marché on cite parmi eux :

- Up.Browser de phone.com.
- Navigateur 7110 du Nokia.
- Microsoft mobile explorer.
- Wap man.
- Navigateur Ericsson pour Plam pilot.

**14. Fonctionnement d'un protocole WAP :**

La figure suivante explicite la façon dont va avoir lieu un échange d'informations à travers une passerelle lors d'une requête.[09]

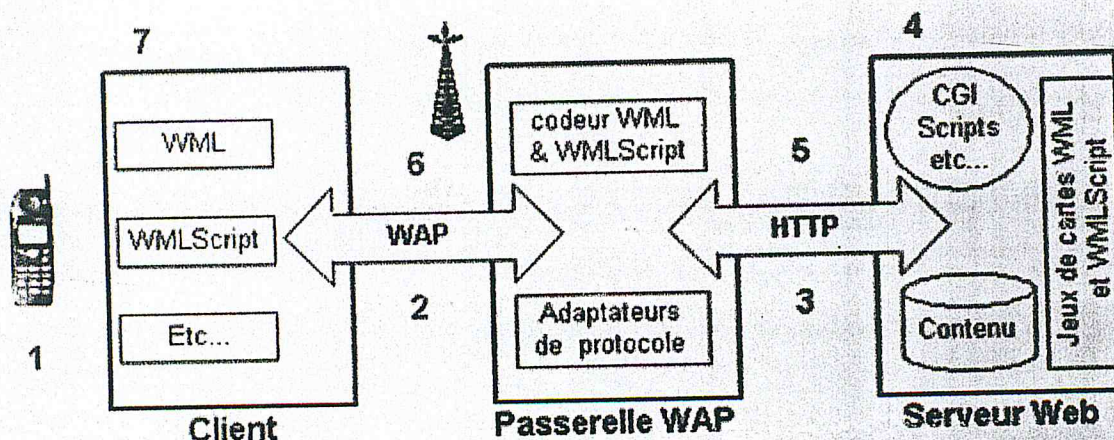


Figure I.10: Echanges d'information avec le protocole WAP

1. L'utilisateur appuie sur une touche de son téléphone à laquelle correspond une URL
2. Une requête est envoyée à la passerelle configurée par défaut, en utilisant le protocole WAP
3. La passerelle WAP crée une requête HTTP conventionnelle pour l'URL demandée et la transmet au serveur Web
4. Cette requête HTTP est analysée par le serveur Web. Si l'URL correspond à un fichier statique (html), le serveur Web va chercher ce fichier et lui ajoute un entête HTTP. Si l'URL correspond à un script CGI ou autres, dans ce cas, le serveur lance l'application correspondante.
5. Le serveur Web retourne le jeu de cartes WML avec l'entête HTTP ajouté, ou directement les données WML issues du script.
6. La passerelle WAP vérifie l'entête HTTP et le contenu WML, et code le tout dans une forme binaire. La passerelle crée alors une réponse au format WAP qui est transmise au navigateur.
7. Le navigateur reçoit une réponse. Il interprète le contenu WML et affiche la première carte du jeu de carte (voire description du langage WML).

### 15. Les différents implémentations d'une plate forme WAP :

Les architectures suivantes présentent les différentes possibilités d'intégration des briques technologiques qui composent une plate forme WAP.[09]

#### 15.1- Passerelle WAP hébergée chez l'opérateur de téléphonie mobiles :

Cette architecture est la solution par défaut proposée par les opérateurs de téléphonie mobile pour accéder au service WAP, L'opérateur héberge dans ce cas le service d'accès distant (accès réseau IP) et la passerelle.

Les services WAP référencés sur les portails des opérateurs utilisent ce type de solution dont la mise en œuvre est simple.

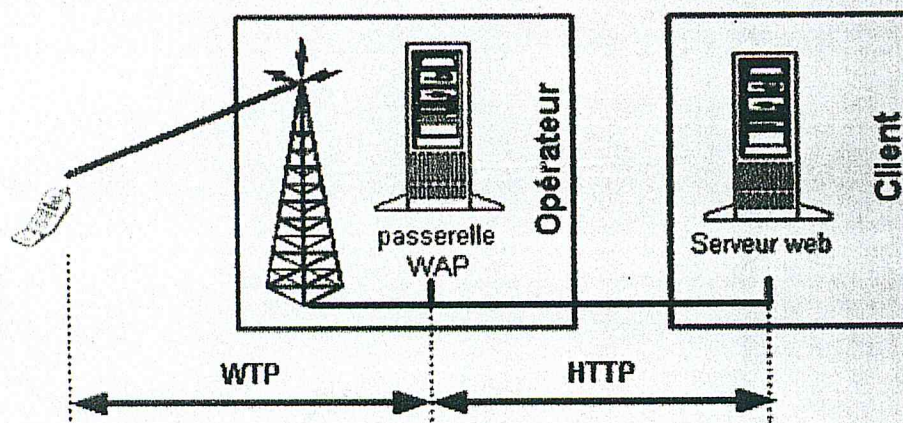


Figure I.11 : Passerelle WAP chez l'opérateur

15.2-Passerelle WAP hébergée chez un fournisseur d'accès :

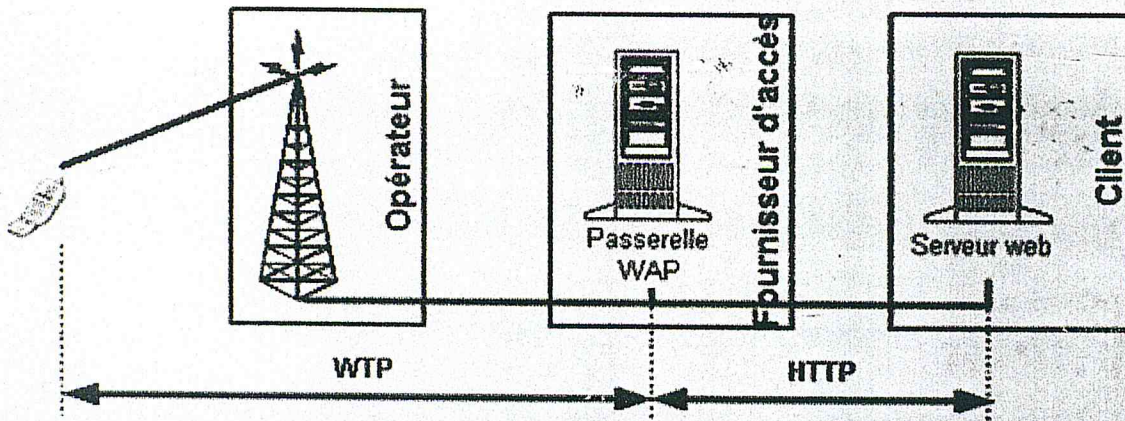


Figure I.12 : Passerelle chez un fournisseur sans hébergement de sites web

Cette solution présente l'avantage, pour les fournisseurs d'accès, de ne pas être dépendants des opérateurs téléphoniques et de pouvoir ainsi administrer leur propre passerelle WAP. Dans ce cas, seule la connectivité GSM/RTC des opérateurs est utilisée. L'utilisateur compose ainsi directement le numéro du fournisseur qui devient donc un opérateur téléphonique alternatif spécialisé dans la connexion Internet via le WAP.

Si le fournisseur a également la vocation d'héberger des sites Web, on se retrouve dans la configuration ci dessous :

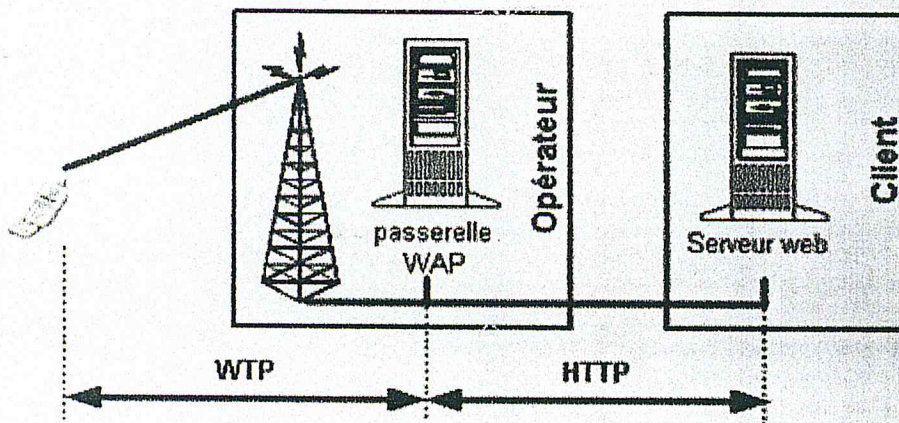


Figure I.13 : passerelle chez un fournisseur avec hébergement de sites web

Dans ce cas, on ne parle plus de passerelle WAP mais bien de serveur WAP, puisque la machine qui l'héberge intègre à la fois la passerelle WAP mais aussi un serveur applicatif WAP. Le fournisseur devient donc un opérateur téléphonique alternatif spécialisé dans la connexion Internet via le WAP mais aussi dans la diffusion de contenus.

### 15.3- passerelle WAP en interne

Si le client (propriétaire de l'applicatif) veut garder la maîtrise complète des connexions WAP, il peut décider d'avoir sa propre passerelle WAP, on se retrouve alors dans la situation ci-dessous

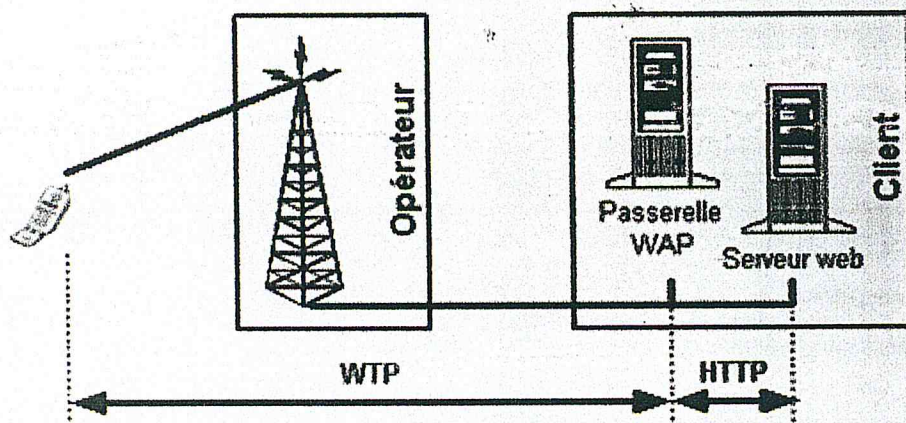


Figure I.14 : passerelle Wap chez le client

Cette solution peut être envisageable dans le cas d'applications WAP dédiées à une entreprise et exploitant des données sur l'Intranet.

## 16. Les avantages et les inconvénients de la technologie WAP :

- **Les avantages :**

Les téléphonies mobiles intégrant la technologie WAP permettent tout d'abord d'accéder à des services interactifs tels que la consultation de la météo, de l'évolution des embouteillages dans les grandes villes, e consulte son compte e banque, comme ils permettent également de consulter des pages WML. On peut donc, n'importe où, profiter de l'immense regroupement de connaissance se trouvant sur le Web. De plus, avec l'évolution du GSM vers le GPRS, et donc l'évolution des taux d'accès sur le réseau (le GPRS étant annoncé comme offrant un débit de communication dix fois supérieurs à celui du GSM actuel), le protocole WAP permettra d'offrir à ces clients des services d'une grande interactivité (en utilisant par exemple le système de localisation du porteur du téléphone en temps réel).

- **Les inconvénients :**

Malgré tout les efforts développés pour établir une compatibilité entre les sites Web et l'affichage sur téléphone mobile, la surface de l'écran de ces téléphones ne dépassera pas quelques lignes. De plus, avec les taux de transfert actuel (9,6 Kbps), l'affichage d'image reste possible mais guère performante.

# Chapitre II.

## Les outils de développement

### 1. Les outils de développement web :

#### 1.1. Langage HTML :

Le langage HTML (Hyper Text Markup Langage) est un langage de formatage de documents hypertextes. Autrement dit, il sert à définir la structure et le contenu des pages Web.

Le langage en lui-même fut développé en 1990 par Tim Berners-Lee pendant sa thèse au CERN (Genève). Il fut rapidement popularisé par Mosaic (le premier navigateur Web). Dès la version 2.0, les spécifications de cet langage ont été confiées à un groupe international indépendant, le 'World Wide Web Consortium' (W3C). Ce consortium rassemble des universités, des entreprises et des organismes publics des différents pays (DARPA, MIT, INRIA, ...).

L'indépendance du W3C a permis de mettre sur un pied d'égalité toutes les entreprises et les utilisateurs du standard HTML. Cette structure permet d'éviter toute prise de monopole par l'un des acteurs du marché (comme c'est souvent le cas en informatique). Ainsi, les sociétés rivalisent pour essayer de fournir le navigateur le plus compatible possible avec la norme HTML en préservant le standard commun.

Le langage HTML lui-même est spécifié en SGML (Standard Generalized Markup Langage). Ce langage permet de décrire d'autre langage du type HTML.

SGML est compliqué mais il a l'avantage de pouvoir être traduit facilement en plusieurs formats (HTML, Postscript, ASCII, RTF). Cela permet de n'écrire qu'une seule fois le documents et ne pas avoir à le retoucher suivant le format désiré (page Web, document papier, lecture sur ordinateur de poche, ...)

Cependant, au fil du temps, GML s'est avéré très limité sur certains points (complexité d'utilisation, liens trop étroits entre le contenu d'un document et sa représentation, faiblesse du pouvoir d'expression, manque de flexibilité, ...). [06]

HTML décrit donc la présentation u contenu d,une page Web : les éléments à afficher sont encadrés par des balises qui précisent leur mise en forme. Ainsi, différentes balise permettent de changer la taille es caractères, alignement, etc. :

**<b> avec cette balise, cette phrase sera afficher en gras. </b>**

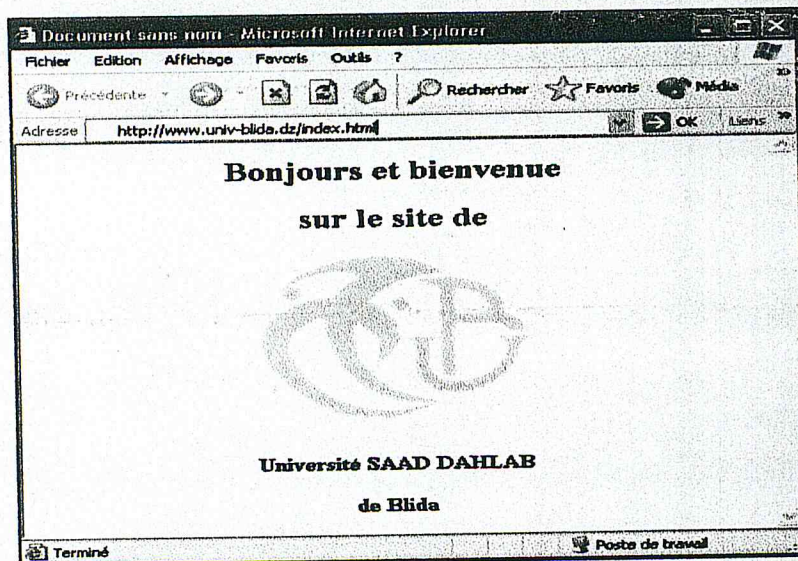
La liste des balises utilisables est définie pour chaque version du langage HTML. Certaines balises contiennent des références sur d'autres objets, comme des images par exemple, qui sont affichées au milieu du texte :

**<IMG SRC= "logo :gif" ALIGN = CENTER >**

D'autre balise particulière contiennent des liens hypertextes, sur lesquels l'utilisateur peut cliquer pour remplacer la page active par une autre :

**<A HERF="/cours/java.html" >**

Bien sûr ; il ne s'agit ici que d'un court aperçu des possibilités du langage HTML ; de nombreux ouvrages sont consacrés à ce sujet et il serait hors de propos de le détailler ici. D'ailleurs, de nombreux outils à interface WYSIWYG permettent aujourd'hui de générer du HTML de façon totalement transparente pour le développeur de sites Web.



```
<html>
<head>
  <title>Exemple</title>
</head>
<body>
  <div align="center" ><h2>Bonjours et bienvenue<br>
    sur le site de </h2>
    
    <p><h3>Universit&eacute; SAAD
      DAHLAB</strong><br>
      de Blida </h3>
    </p>
  </div>
</body>
</html>
```

FigureII-1 Une page HTML simple et le code correspondant.

## 1.2. Quelques exemples de navigateur :

Les navigateurs présentés ci-dessous sont tous téléchargeables gratuitement. La liste n'est pas exhaustive mais présente les navigateurs les plus rencontrés.

### Internet Explorer

Il s'agit du navigateur qui est systématiquement fourni avec le système d'exploitation Windows. C'est la raison pour laquelle il est le navigateur le plus utilisé, ce qui fait aussi de lui la cible rêvée des concepteurs de virus qui veulent atteindre le plus haut degré de propagation. IE n'est utilisable que sous Windows. Il existe actuellement une version d'Internet Explorer pour Macintosh qui est très différente de son homonyme sous Windows.

### Mozilla

Fruit du mouvement Open-Source Mozilla.org (initié par Netscape), Mozilla est disponible sur Windows, Macintosh, Linux et bien d'autres systèmes d'exploitation (Solaris, OS/2, HP/UX, OpenVMS). Il dispose de fonctionnalités qui séduisent autant les utilisateurs avancés (blocage anti-pop-up, blocage anti-spam) que les développeurs web (Débugueur JavaScript, inspecteur de DOM) et les débutants (avec un éditeur de pages web très convivial). Originellement en anglais, il est possible de télécharger une version française.

Outre un navigateur Internet, Mozilla propose aussi un client mail et un éditeur de pages web.

### Mozilla FireFox

Anciennement connu sous les noms Phoenix et Firebird, FireFox est le petit frère de Mozilla. Il utilise une version optimisée du cœur de son aîné et s'améliore au fil du temps. Objectif de FireFox : être le plus léger et le plus rapide possible, et remplacer à terme Mozilla. De nombreuses extensions sont disponibles pour étendre ses fonctionnalités. Bien entendu, il existe une version française de ce navigateur.

Contrairement à Mozilla, Firefox ne propose qu'un navigateur Internet qui est bien plus sûr que Internet Explorer et plus épuré que Mozilla.

Firefox est également utilisable par les différents systèmes d'exploitation (Windows, Linux, Macintosh, ...).

Mozilla et FireFox sont non seulement gratuits mais également open-source, ce qui signifie que leur code source est consultable et modifiable par les utilisateurs désireux d'adapter ces navigateurs à leurs besoins.

### Netscape navigator

Frère presque jumeau de Mozilla, Netscape est la version grand public de ce dernier. Netscape est gratuit mais pas Open-Source. Il est disponible pour Windows (de 98 à XP), Mac OS 8, 9 et X, Linux.

**Safari** Il s'agit du navigateur par défaut du système d'exploitation Macintosh.



## 2. Les outils de développement Wap :

### 2.1. Le langage WML :

Le langage WML (*Wireless Markup Language*) est un langage à balises basé sur XML permettant à des terminaux mobiles (téléphones mobiles, pagers ou bien des assistants digitaux personnels) d'afficher des documents reliés par des liens hypertextes. Le langage WML est très proche du HTML, mais il est toutefois beaucoup moins "souple" que ce dernier, c'est-à-dire qu'il doit être conforme à un document appelé **DTD** (*Document Type Definition*), indiquant la syntaxe à suivre pour créer un document WML. En effet, étant donné que les documents WML sont prévus pour être utilisés sur des réseaux de bande passante réduite, et des terminaux restreints (en terme d'affichage, et de ressources systèmes), le WML se doit d'être compact. Le langage WML permet ainsi l'affichage de textes, d'images, de tableaux sur l'écran du terminal, ainsi qu'une navigation entre les documents.[04]

Le langage WML n'est donc pas un langage de programmation, c'est un simple fichier texte contenant des balises permettant de mettre en forme le texte, les images ... Une balise est une commande (un nom) encadrée par le caractère inférieur (<), aussi appelé *chevron ouvrant*, et le caractère supérieur (>), appelé *chevron fermant*, par exemple "<p>".

### 2.2. La structure d'un document WAP :

La conception d'une "page" Wap est assez différente d'une page Web classique. En effet, la page Web qui est affichée correspond au contenu d'un fichier Html complet. Du fait de la taille d'écran réduite des mobiles, on ne pourra afficher qu'un petit fragment d'information. Par ailleurs, il n'est pas concevable d'entreprendre une connexion téléphonique pour chacun de ces petits fragments.

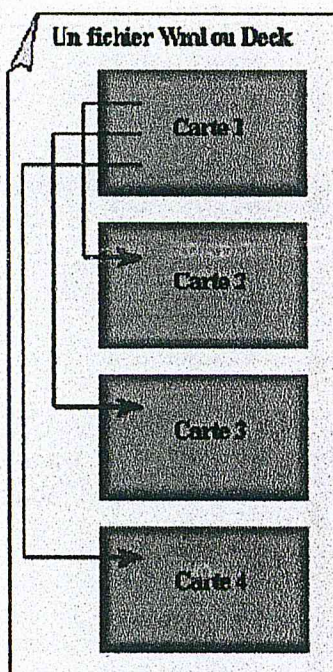


Figure II.2 Un Fichier WML ou DECK

Ainsi, la procédure adoptée par le Wap sera de charger un ensemble, un paquet de fragments qui sera stocké dans la mémoire du mobile. Les différents fragments seront appelés au cours de la consultation.

L'ensemble des fragments correspond à un fichier à l'extension .wml et est appelé un document WML, par assimilation une page WML et par les anglo-saxons, un "deck" ou un paquet de cartes.

Les différents fragments sont appelés des "cards" ou simplement des cartes.

Un fichier ou une page WML est donc un "paquet de cartes". Ce fichier ou ce paquet est composé d'une série de "cartes". [10]

### 2.3. Exemple1 :

```
<?xml version="1.0"?>
<!DOCTYPE wml ...>
<wml>
<card id="ct1" title="Carte 1">
<p>
Carte 1<br/>
<a href="#ct2">Vers carte 2</a><br/>
<a href="#ct3">Vers carte 3</a>
</p>
</card>
<card id="ct2" title="Carte 2">
<p>
Carte 2<br/>
<a href="#ct1">Retour carte 1</a>
</p>
</card>
<card id="ct3" title="Carte 3">
<p>
Carte 3<br/>
<a href="#ct1">Retour carte 1</a>
</p>
</card>
</wml>
```

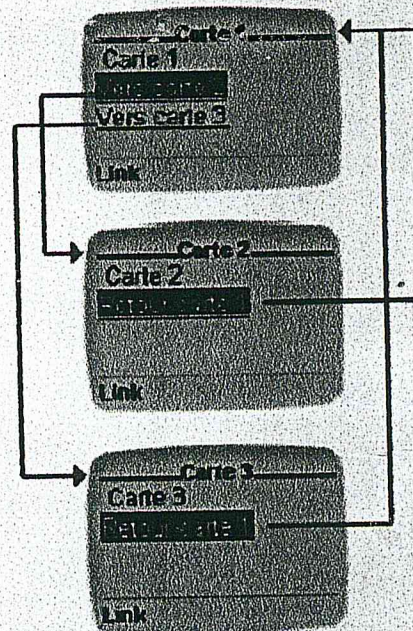


Figure II.3 Les Résultats de l'exemple1

### Exemple2 :

```
<?xml version="1.0"?>
<!DOCTYPE wml ...>
<wml>
<card id="ct1" title="La balise p">
<p align="left">
à gauche
</p>
<p align="center">
centré
</p>
<p align="right">
```

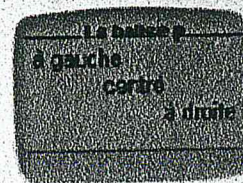


Figure II.4 Les Résultats de l'exemple2

à droite

</p>

</card>

</wml>

**Principalement pour Développez avec le Wap on est besoin des outils suivantes :**

- Un éditeur WML.
- Un convertisseur d'image.
- Un émulateur WAP

### **2.3. Les editeurs WAP :**

Il existe sur le marché un nombre important d'éditeurs, et voila quelques un, qui sont les plus utilisé.

- Easypad WAPtor : Simple d'usage et permettant de visualiser directement le résultat du code WML.
- DotWap : Sa principe qualité est de pouvoir visualiser directement le résultat code WML sur une simulation d'un mobile Ericsson.

### **2.4. Les convertisseurs d'images :**

La plupart des téléphones portables ne supportent que les images de format Wbmp ou Png, Pour cela il faut utiliser un convertisseur des images de tout formats vers une format compatible avec les terminaux WAP. Mais il existe actuellement dans le marché des téléphones portables qui supportent autre format d'image couleurs comme bmp, jpeg ... .

Il existe plusieurs logiciels convertisseur vers Wbmp comme le **Pic2wbmp**, **wbmp converter...**, et comme **paint shop** pour les images de format Png.

### **2.5. Les émulateurs WAP :**

Les émulateurs WAP sont des outils permettant au développeurs de tester leurs applications WAP sont être besoin d'un véritable téléphone portable, comme il peut consulter un site WAP depuis leurs disques durs et non a partir d'une adresse HTTP sur le Web.

Il existe plusieurs émulateurs sur le marché comme :

- Emulateur Nokia 7110.
- Ericsson.
- Siemens S45.
- Openwave.
- M3gate.
- Win Wap... etc.



Openwave



Siemens



Nokia



M3gate

Figure II.5 : Les différents émulateurs Wap

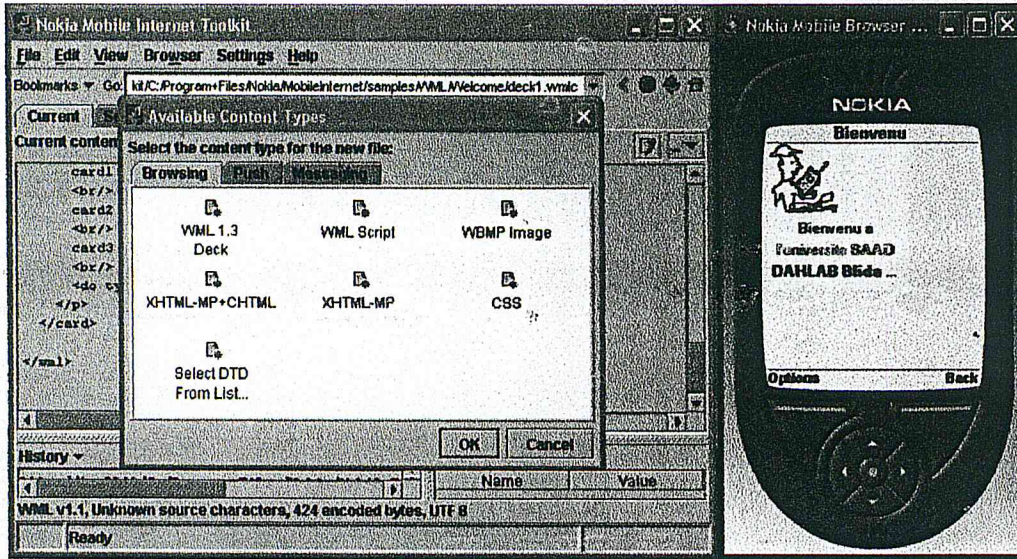
## 2.6. Les Kits de développement :

Les Kits de développement proposés par les constructeurs de téléphones portables sont la meilleure solution pour les WAP –développeurs. Vu les problèmes de compatibilité du WAP entre les marques et les types de mobiles, c'est une solution idéale pour tester un site. Bien étendu leur prise en main et moins intuitive que les éditeurs précédentes. Les trois acteurs majeurs de marché de téléphonie mobile que sont NOKIA, Motorola et Ericsson proposent chacun une plate forme de développement et de test adapté à leur produits.

### 2.6.1. Nokia Internet mobile toolkit :

Nokia dispose un kit de développement qui s'appelle « Nokia mobile Internet toolkit », ce kit propose une série de fonctions qui sont :[08]

- Simulation des mobiles 7110,6110 et 6150.
- Visualisation de requete des erreurs.
- Editeur des images WBmp.
- Visualisation des variables locales.
- Connexion à une passerelle WAP.
- Connexion directe a un serveur http.
- Edition, analyse syntaxique.
- Documentation complete sur WAP, WML et WMLScript

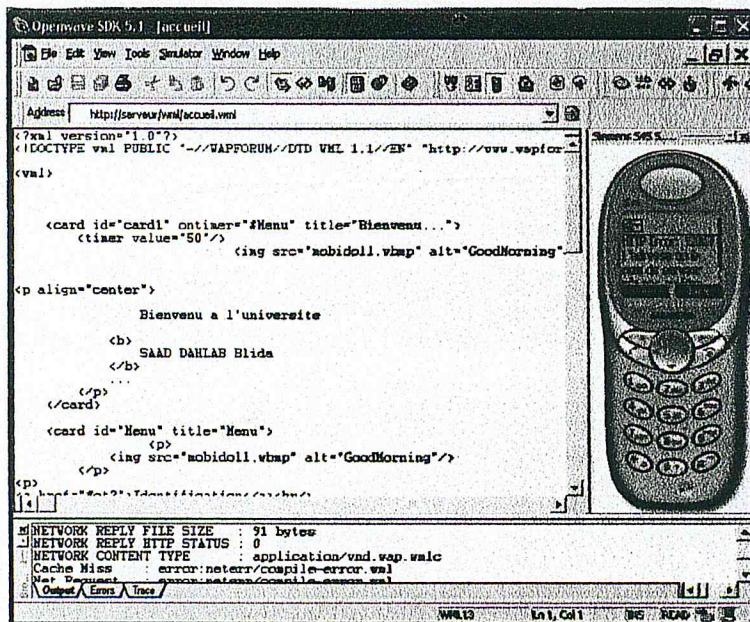


FigureII.6 : Nokia Internet mobile toolkit

2.6.2. OpenWave SDK :

Tout comme le Nokia mobile Internet toolkit, le kit de développement OpenWave SDK assure une série de fonction tels que :[49]

- Simulation des mobiles Siemens S45 et OpenWave.
- Visualisation de requête des erreurs.
- Visualisation des variables locales.
- Connexion à une passerelle WAP.
- Connexion directe a un serveur http.
- Edition, analyse syntaxique des documents WML,XHTML, HTML,...
- Exemples du code WML.



FigureII.7 :OpenWave SDK toolkit

### 3. Base de donnée et SGBD :

Les bases de données sont actuellement au coeur du système d'information des entreprises. Les systèmes de gestion de bases de données, initialement disponibles uniquement sur des « mainframes », peuvent maintenant être installés sur tous les types d'ordinateurs y compris les ordinateurs personnels.

Mais, souvent on désigne, par abus de langage, sous le nom "bases de données" des ensembles de données qui n'en sont pas.

Qu'est-ce donc qu'une base de données? Que peut-on attendre d'un système de gestion de bases de données? C'est à ces questions, entre autres, que cette partie essaie d'apporter des réponses.

Dans un premier temps, et de façon informelle, on peut considérer une Base de Données (BD) comme une grande quantité de données, centralisées ou non, servant pour les besoins d'une ou plusieurs applications, interrogeables et modifiables par un groupe d'utilisateurs travaillant en parallèle. Quant au Système de Gestion de Bases de Données (SGBD), il peut être vu comme le logiciel qui prend en charge la structuration, le stockage, la mise à jour et la maintenance des données ; c'est, en fait, l'interface entre la base de données et les utilisateurs ou leurs programmes.

#### 3.1. Les limites à l'utilisation des fichiers :

L'utilisation de fichiers impose d'une part, à l'utilisateur de connaître l'organisation (séquentielle, indexée, ...) des fichiers qu'il utilise afin de pouvoir accéder aux informations dont il a besoin et, d'autre part, d'écrire des programmes pour pouvoir effectivement manipuler ces informations. Pour des applications nouvelles, l'utilisateur devra obligatoirement écrire de nouveaux programmes et il pourra être amené à créer de nouveaux fichiers qui contiendront peut-être des informations déjà présentes dans d'autres fichiers.

De telles applications sont :

- rigides,
- contraignantes,
- longues et coûteuses à mettre en oeuvre.

Les données associées sont :

- mal définies et mal désignées,
- redondantes,
- peu accessibles de manière ponctuelle,
- peu fiables.

La prise de décision est une part importante de la vie d'une société. Mais elle nécessite d'être bien informé sur la situation et donc d'avoir des informations à jour et disponibles immédiatement.

Les utilisateurs, quant à eux, ne veulent plus de systèmes d'information constitués d'un ensemble de programmes inflexibles et de données inaccessibles à tout non spécialiste ; ils souhaitent des systèmes d'informations globaux, cohérents, directement accessibles (sans

qu'ils aient besoin soit d'écrire des programmes soit de demander à un programmeur de les écrire pour eux) et des réponses immédiates aux questions qu'ils posent. On a donc recherché des solutions tenant compte à la fois des désirs des utilisateurs et des progrès techniques. Cette recherche a abouti au concept de base de données.

### 3.2. Qu'est-ce qu'une base de données?

Une base de données (son abbréviation est BD, en anglais DB, *database*) est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de base. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données. [06]

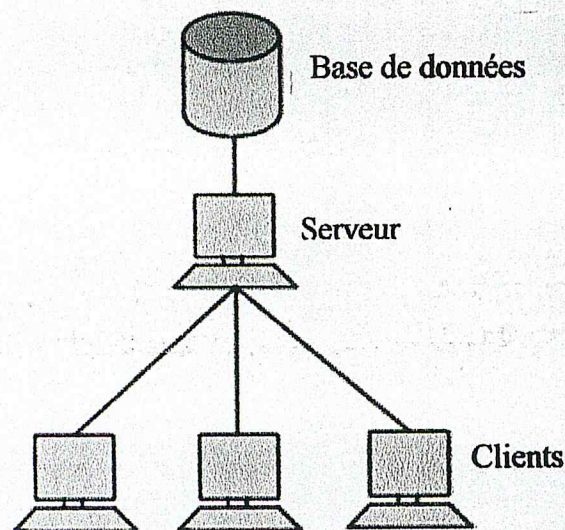


Figure II-8: les bases de données dans un réseau

### 3.3. Utilité d'une base de données :

Une base de données permet de regrouper des données au sein d'un même enregistrement. Cela est d'autant plus utile que les données informatiques sont de plus en plus nombreuses.

Une base de données peut être locale, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien répartie, c'est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau.

L'avantage majeur de l'utilisation de bases de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément.

### 3.4. La gestion des bases de données :

Afin de pouvoir contrôler les données ainsi que les utilisateurs, le besoin d'un système de gestion s'est vite fait ressentir. La gestion de la base de données se fait grâce à un système appelé SGBD (système de gestion de bases de données) ou en anglais DBMS (Database management system). Le SGBD est un ensemble de services (applications logicielles) permettant de gérer les bases de données, c'est-à-dire: [22]

- permettre l'accès aux données de façon simple
- autoriser un accès aux informations à de multiples utilisateurs
- manipuler les données présentes dans la base de données (insertion, suppression, modification)

Le SGBD peut se décomposer en trois sous-systèmes ou Niveau selon le standard ANSI/SPARC:

- le système de gestion de fichiers: il permet le stockage des informations sur un support physique
- le SGBD interne (ou conceptuel): il gère l'ordonnancement des informations
- le SGBD externe: il représente l'interface avec l'utilisateur

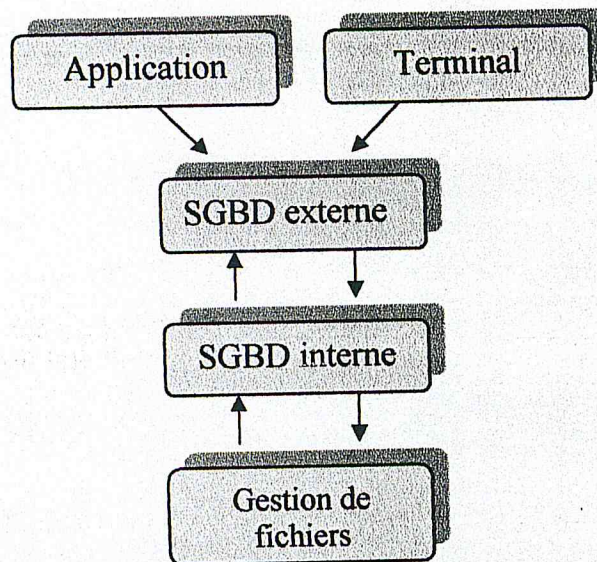


Figure II-9: Les niveaux d'un SGBD selon le standard ANSI/SPARC

### 3.5. Les différents modèles des SGBD :

Les bases de données sont apparues la fin des années 60, à une époque où la nécessité d'un système de gestion de l'information souple se faisait ressentir. Il existe cinq modèles de SGBD, différenciés selon la façon de laquelle les données sont représentées:

1. **le modèle hiérarchique:** les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements. Il s'agit du premier modèle de SGBD

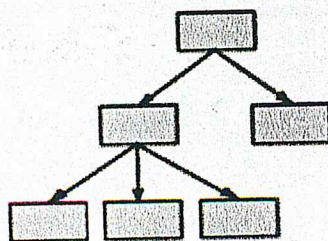


Figure II-10: le modèle hiérarchique



2. **le modèle réseau:** Comme le modèle hiérarchique ce modèle utilise des pointeurs vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant

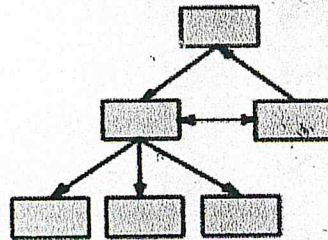


Figure II-11:le modèle réseau

3. **le modèle relationnel (SGBDR, Système de gestion de bases de données relationnelles):** les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes). La manipulation de ces données se fait selon la théorie mathématique des relations

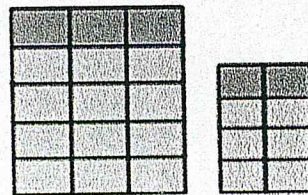


Figure II-12:le modèle relationnel

4. **le modèle déductif:** les données sont représentées sous forme de table, mais leur manipulation se fait par calcul de prédicats
5. **le modèle objet (SGBDO, Système de gestion de bases de données objet):** les données sont stockées sous forme d'objets, c'est-à-dire de structures appelées classes présentant des données membres. Les champs sont des instances de ces classes

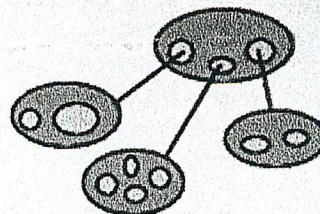


Figure II-13:le modèle objet

### 3.6. Les caractéristiques d'un SGBD :

L'architecture à trois niveaux définie par le standard ANSI/SPARC permet d'avoir une indépendance entre les données et les traitements. D'une manière générale un SGBD doit avoir les caractéristiques suivantes:[22]

- **Indépendance physique:** Le niveau physique peut être modifié indépendamment du niveau conceptuel. Cela signifie que tous les aspects matériels de la base de données n'apparaissent pas pour l'utilisateur, il s'agit simplement d'une structure transparente de représentation des informations
- **Indépendance logique:** le niveau conceptuel doit pouvoir être modifié sans remettre en cause le niveau physique, c'est-à-dire que l'administrateur de la base doit pouvoir la faire évoluer sans que cela gêne les utilisateurs
- **Manipulabilité:** des personnes ne connaissant pas la base de données doivent être capables de décrire leurs requêtes sans faire référence à des éléments techniques de la base de données
- **Rapidité des accès:** le système doit pouvoir fournir les réponses aux requêtes le plus rapidement possible, cela implique des algorithmes de recherche rapides
- **Administration centralisée:** le SGBD doit permettre à l'administrateur de pouvoir manipuler les données, insérer des éléments, vérifier son intégrité de façon centralisée
- **Limitation de la redondance:** le SGBD doit pouvoir éviter dans la mesure du possible des informations redondantes, afin d'éviter d'une part un gaspillage d'espace mémoire mais aussi des erreurs
- **Vérification de l'intégrité:** les données doivent être cohérentes entre elles, de plus lorsque des éléments font références à d'autres, ces derniers doivent être présents
- **Partageabilité des données:** le SGBD doit permettre l'accès simultané à la base de données par plusieurs utilisateurs
- **Sécurité des données:** Le SGBD doit présenter des mécanismes permettant de gérer les droits d'accès aux données selon les utilisateurs

### 3.7. Les principaux SGBD

Il existe maintenant plusieurs bases de données distribuées comme logiciels libres. Le lien suivant [www.linas.org/linux/db.html](http://www.linas.org/linux/db.html) dresse une liste non exhaustive des bases de données disponibles en *open source*. Les bases de données les plus utilisées sont MySQL et PostgreSQL. Deux autres bases de données qui méritent une certaine attention sont SAP DB et Firebird. Une avenue supplémentaire se profile aussi avec l'apparition récente de bases de données XML.[45]

#### MySQL

MySQL est un système de gestion des bases de données relationnelles, multiutilisateurs et multiprocesseur. Cette dernière particularité fait en sorte qu'il peut être utilisé sur un serveur multiprocesseur.

Les principaux objectifs misés par l'équipe de développement sont la facilité d'utilisation, la robustesse ainsi que la performance. Cette base de données est disponible sous licence BSD. Des licences commerciales sont aussi disponibles. À l'écriture de ce rapport, la version 4.0 était disponible. Le site Web de MySQL([www.mysql.com](http://www.mysql.com)) permet le téléchargement de toutes les versions disponibles.

## PostgreSQL

Le développement de cette base de données de type object-relational database management system (ORDBMS) a débuté en 1986 à l'Université de Berkeley en Californie. Elle est distribuée sous licence BSD. La dernière version peut être téléchargée sur le site officiel, soit [www.postgresql.org](http://www.postgresql.org). À l'écriture de ce rapport, la version 7.3.3 était disponible.

## FIREBIRD

Ce projet a été initié en août 2000 lorsque la compagnie Inprise a rendu publique la version bêta d'Interbase 6.0. Cette base de données relationnelle est indépendante de la plate-forme et supporte le standard SQL. La dernière version stable (Firebird 1.0.3) peut être téléchargée via [www.firebird.sourceforge.net](http://www.firebird.sourceforge.net).

## InterBase

InterBase est un système de gestion de base de données relationnelle (SGBDR) développé par Borland permettant un traitement rapide des transactions et un partage de données en environnement mono ou multi-utilisateur. En fait, InterBase est un serveur offrant un support transparent à travers des réseaux hétérogènes. InterBase fonctionne sous Windows 95, Windows NT, Novell NetWare et de nombreuses implantations des systèmes d'exploitation Unix. La version 5.0 est disponible sous une licence libre (gratuite) via : <http://www.borland.com/devsupport/Interbase/opensource/>.

### 3.8. Comparaison entre bases de données relationnelles :

Le tableau suivant, inspiré de [www.geocities.com/mailsoftware42/db/dbs.html](http://www.geocities.com/mailsoftware42/db/dbs.html), présente une comparaison de ces quatre produits. Le symbole (+) utilisé n'est utilisé que pour des fins de comparaisons entre les différents produits.

Caracteristiques	Mysql	PostgreSQL	InterBase	Firebrid
Rapidité <sup>1</sup>	++	+		
Stabilité	++			
Respect des propriétés ACID <sup>2</sup>	Non <sup>3</sup>	Oui		Oui
Multiplate-forme	Oui	Oui (émulation Cygwin)		Oui
Contrôle de privilèges <sup>4</sup>	+++	++	++++	-
Support SSL	Oui	Oui	Oui	Non
Vues	Non	Oui	Oui	Oui
Schémas	Non	Oui	Oui	Non
Réplication	Oui	Non <sup>5</sup>	Oui <sup>6</sup>	Non
API disponible <sup>7</sup>	Oui	Oui	Oui	Oui
Transaction	Non <sup>3</sup>	Oui	Oui	Oui
Procédure stockées	Non	Oui	Oui	Oui
Requete complexe (sub select)	Non	Oui	Oui	Oui
Répartition de la charge	Non	Non	Non	Non

Caracteristiques	Mysql	PostgreSQL	InterBase	Firebird
Contraintes d'intégrité	Non <sup>3</sup>	Oui	Oui	Oui
Trigger	Non	Oui	Oui	Oui
Unicode	Oui	Oui	Oui	Oui
Héritage	Non	Oui	Non	Non

**Tableau II.1 : Comparaison entre bases de données relationnelles**

1 La performance supérieure de MySQL par rapport à PostgreSQL provient du fait que MySQL soit multiprocesseur; les connexions sont donc prises en charge plus rapidement. PostgreSQL doit initier un nouveau processus pour chaque nouvelle connexion. Il est possible d'utiliser des procédures stockées afin d'améliorer les performances de PostgreSQL.

2 ACID: Atomicity, Consistency, Isolation, Durability. Il s'agit des propriétés de base auxquelles une base de données doit se conformer lorsqu'elle supporte les transactions.

3 Le gestionnaire de table transactionnelle InnoDB, disponible sous licence commerciale, offre ce support.

4 MySQL offre un contrôle d'accès (GRANT, REVOKE) pour toutes les commandes. Ce contrôle peut être basé sur un nom d'utilisateur, une table ou client externe.

5 Un mécanisme de réplication asynchrone est possible avec Enterprise Replication Server ([gborg.postgresql.org/project/pgreplication/projdisplay.php](http://gborg.postgresql.org/project/pgreplication/projdisplay.php)).

6 SAP DB n'offre qu'un mécanisme de réplication d'une table en entier et non à partir d'un fichier de journalisation.

7 Toutes les bases de données offrent une API pour ODBC, JDBC, C/C++ et PHP. Une API pour Perl et Python est également disponible pour toutes sauf Firebird. Une API pour TCL est disponible pour MySQL et PostgreSQL. Une API pour Delphi est disponible pour MySQL et Firebird. Il faut cependant noter que certaines API ne sont pas implémentées à 100 %. Par exemple, la fonction getSchemaName n'est pas implémentée pour MySQL et Firebird étant donné que les schémas ne sont pas supportés.

### 3.9. Les procédures stockées

Une procédure stockée est un programme qui comprend des instructions SQL pré-compilées et qui est enregistré dans la base de données

Le plus souvent le programme est écrit dans un langage spécial qui contient à la fois des instructions procédurales et des ordres SQL. Ces instructions ajoutent les possibilités habituelles des langages dits de troisième génération comme le langage C ou le Pascal (boucles, tests, fonctions et procédures,...).

Les procédures stockées offrent des gros avantages pour les applications client/serveur, surtout au niveau des performances [36]:

- le trafic sur le réseau est réduit car les clients SQL ont seulement à envoyer l'identification de la procédure et ses paramètres au serveur sur lequel elle est stockée.
- les procédures sont pré-compilées une seule fois quand elles sont enregistrées. L'optimisation a lieu à ce moment et leurs exécutions ultérieures n'ont plus à passer par cette étape et sont donc plus rapides.
- la gestion et la maintenance des procédures sont facilitées car elles sont enregistrées sur le serveur de base de données et ne sont pas dispersées sur les postes clients.

### 3.10. Les déclencheurs (Trigger)

Un déclencheur est une routine indépendante associée à une table ou une vue, qui exécute automatiquement une action lorsqu'une ligne est insérée, mise à jour ou supprimée dans cette table ou vue.

Un déclencheur n'est jamais appelé directement. Par contre lorsqu'une application ou un utilisateur tente d'insérer, de mettre à jour ou de supprimer (INSERT, UPDATE, DELETE)

une ligne dans une table, Les déclencheurs associés à cette table et à cette opération sont automatiquement exécutés, ou actifs.

Les déclencheurs peuvent faire appel à des exceptions, qui sont des messages nommés utilisés pour la gestion d'erreur. Lorsqu'une exception est activée par un déclencheur, elle envoie un message d'erreur, met fin au déclencheur et annule toutes les modifications qu'il avait effectuées, sauf si elle est interceptée par une instruction `WHEN` au sein du déclencheur.[47]

L'utilisation des déclencheurs offre les avantages suivants :

- Elle rend automatiquement la mise en application des restrictions relatives aux données, ce qui permet de s'assurer que les utilisateurs n'inséreront dans les colonnes que des valeurs autorisées.
- Elle réduit la maintenance de l'application, étant donné que les modifications effectuées sur un déclencheur sont automatiquement répercutées dans toutes les applications qui utilisent la table associée, sans qu'il soit nécessaire de les recompiler ou d'effectuer à nouveau leur édition de liens.
- Elle permet de consigner automatiquement les modifications effectuées sur les tables. Une application peut tenir un journal d'exécution des modifications, grâce à un déclencheur activé chaque fois qu'une table est modifiée.
- Elle signale automatiquement les modifications effectuées dans la base de données, à l'aide de signaux d'événements situés dans les déclencheurs.

## 4. Langage de programmation

### 4.1. Introduction à 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 «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 : [06]

- 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.

L'un des avantages évidents de ce langage est une bibliothèque d'exécution qui se veut indépendante de la plate-forme : en théorie, il vous est possible d'utiliser le même code pour Windows 95/98/NT, Solaris, UNIX, Macintosh, etc. Cette propriété est indispensable pour une programmation sur Internet (cependant, par rapport à la disponibilité sur Windows et Solaris, les implémentations sur d'autres plates-formes ont toujours un léger décalage). [41]

#### 4.2. Les API de J2EE

Les API de J2EE peuvent se répartir en trois grandes catégories :

- Les composants. On distingue habituellement deux familles de composants :
  - Les composants web : Servlets et JSP (Java Server Pages). Il s'agit de la partie chargée de l'interface avec l'utilisateur (on parle de *logique de présentation*).
  - Les composants métier : EJB (Enterprise Java Beans). Il s'agit de composants spécifiques chargés des traitements des données propres à un secteur d'activité (on parle de *logique métier* ou de *logique applicative*) et de l'interfaçage avec les bases de données.
- Les services, pouvant être classés par catégories :
  - Les services d'infrastructures : il en existe un grand nombre, définis ci-dessous :
    - **JDBC** (*Java DataBase Connectivity*) est une API d'accès aux bases de données relationnelles.
    - **JNDI** (*Java Naming and Directory Interface*) est une API d'accès aux services de nommage et aux annuaires d'entreprises tels que DNS, NIS, LDAP, etc.
    - **JTA/JTS** (*Java Transaction API/Java Transaction Services*) est un API définissant des interfaces standard avec un gestionnaire de transactions.
    - **JCA** (*J2EE Connector Architecture*) est une API de connexion au système d'information de l'entreprise, notamment aux systèmes dits «Legacy» tels que les ERP.
    - **JMX** (*Java Management Extension*) fournit des extensions permettant de développer des applications web de supervision d'applications.
  - Les services de communication :
    - **JAAS** (*Java Authentication and Authorization Service*) est une API de gestion de l'authentification et des droits d'accès.
    - **JavaMail** est une API permettant l'envoi de courrier électronique.
    - **JMS** (*Java Message Service*) fournit des fonctionnalités de communication asynchrone (appelées *MOM* pour *Middleware Object Message*) entre applications.
    - **RMI-IIOP** est une API permettant la communication synchrone entre objets.

L'architecture J2EE permet ainsi de séparer la couche présentation, correspondant à l'interface homme-machine (IHM), la couche métier contenant l'essentiel des traitements de données en se basant dans la mesure du possible sur des API existantes, et enfin la couche

de données correspondant aux informations de l'entreprise stockées dans des fichiers, dans des bases de données relationnelles ou XML, dans des annuaires d'entreprise ou encore dans des systèmes d'information complexes.[06]

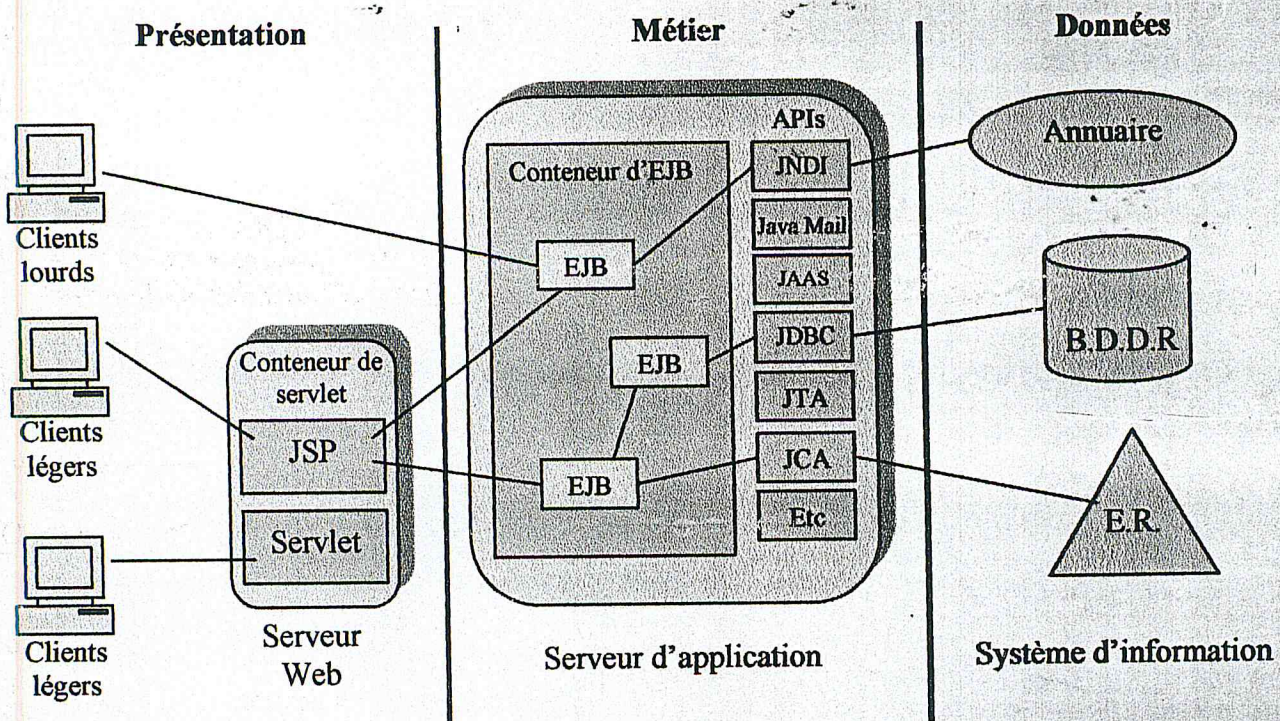


Figure II-14: L'architecture J2EE

### 4.3. Les servlets java :

Les servlets sont apparus en 1997, Depuis, leur utilisation n'a cessé de croître. Parce qu'ils sont indépendants de la plate-forme, qu'ils supportent la montée en charge, et qu'ils permettent d'atteindre des performances bien supérieures à celles des script CGI (Common Gateway Interface), Les servlets sont un composant important de J2EE (Java 2 Entreprise Edition) et suscitent l'intérêt de nombreux programmes.

#### 4.3.1. Qu'est-ce qu'une servlet ?

Une servlet (Server side applet) est une classe Java qui étend de manière dynamique les fonctions d'un serveur Web. Parce qu'il est écrit en Java, il a pleinement accès aux fonctionnalités avancées de Java-connectivité aux bases de données, gestion du réseau, orienté objet, et support natif pour les processus à plusieurs threads. Ces fonctionnalités peuvent être mises en œuvre pour réaliser des applications sophistiquées auxquelles les clients Web peuvent accéder sans avoir besoin d'une configuration particulière. Parce qu'ils utilisent le protocole HTTP pour interface, les servlets fonctionnent toujours de la même manière, quelque soit l'environnement de navigateur.

Une servlet s'exécute dans le contexte d'une machine virtuelle Java (JVM) gérée par le moteur de servlet. Tout comme un script CGI, une servlet est invoquée pour traiter une requête formulée via un client Web ; mais contrairement à un script CGI, qui requiert qu'un nouveau processus soit créé pour traiter chaque requête, une servlet reste chargée dans la machine virtuelle, pour traiter d'autres requêtes. Chaque nouvelle requête utilise le même

copie du servlet en mémoire, mais traitée par un thread particulier pour optimiser les performances de l'exécution.

La figure II.15 présente les principaux composants de l'environnement des servlets. Un moteur de servlets, qui est généralement un programme externe, est connecté par un moyen quelconque à un serveur Web spécifique. Le moteur de servlets intercepte les requêtes HTTP qu'il identifie comme étant des requêtes à des servlets. Les autres requêtes sont traitées par le serveur Web de la manière la plus habituelle. Le moteur de servlets charge la servlet appropriée s'il n'est pas déjà en cours d'exécution, puis affecte un thread disponible au traitement de la requête, et renvoie les données en sortie par la servlet au client Web à l'origine de la requête. [30]

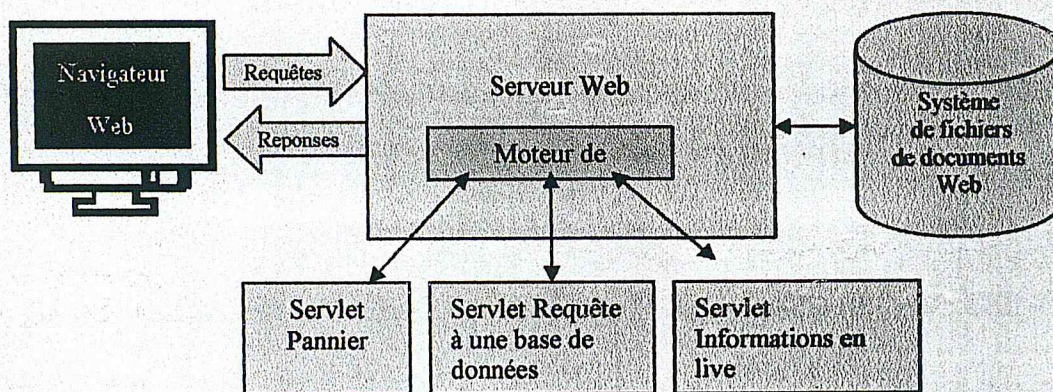


Figure II.15 : les servlets gères par le moteur de servlets.

#### 4.3.2. Les différents types de moteur de servlets :

Il existe trois types différents de moteurs de Servlets :

- Les moteurs de Servlets indépendants.
- Les moteurs de Servlets embarqués.
- Les moteurs de Servlets externes.

- 1- Les moteurs de Servlets indépendants constituent une partie intégrante du serveur Web. Pour qu'une telle chose soit possible, il faut en principe que le serveur Web soit développé en Java. Un exemple de ce type de serveurs est le **Java Web Server**, ou encore **Tomcat** (dont nous analyserons la configuration en détail ultérieurement).
- 2- Les moteurs de Servlets embarqués sont une combinaison d'un ajout à un serveur Web et d'une implémentation de l'API Servlet. L'ajout (le plugin) au serveur Web permet d'aiguiller les requêtes venant des clients et qui concernent des Servlets vers une machine virtuelle contenue dans le processus du serveur Web. Cette machine virtuelle est exécutée dans un thread séparé ce qui implique le fonctionnement du serveur Web en multi-threads, ce qui n'est pas le cas de tous les serveurs Web (par exemple Apache sous Unix). Cette configuration apporte de bonnes performances, car les changements de contexte sont moins coûteux, mais est limitée en possibilité d'extensions (un serveur supportant un très grand nombre de requêtes au même moment aura du mal à pouvoir répondre à ces dernières).



3- Les moteurs de Servlets externes sont une combinaison entre un plugin "greffé" au serveur et une machine virtuelle tournant à l'extérieur de celui-ci. Afin de communiquer entre eux, le plugin et le processus associé à la machine virtuelle utilisent un mécanisme de communication inter-processus tel que les sockets TCP/IP. Si une requête passée au serveur Web concerne les Servlets, celui-ci passe la requête au moteur de Servlets en utilisant le mécanisme sus-cité. Un des inconvénients de cette méthode est la diminution des performances. Par contre, ce type de serveur est en général plus stable (un arrêt du serveur Web n'agit pas sur le moteur de Servlets et vice-versa) et plus extensible. De plus, étant donné que la machine virtuelle est extérieure au serveur Web, elle peut tourner sur n'importe quelle machine, ce qui vous permet de faire tourner le serveur Web sur une machine différente de celle sur laquelle est lancé le moteur de Servlets. Vous pouvez également mettre en place plusieurs machines virtuelles pour un même serveur Web2.2. En bref, la souplesse que l'on gagne peut être intéressante à de nombreux points.

#### 4.3.3. Interface d'une servlet :

Pour créer une servlet il est indispensable de mettre en oeuvre l'interface `javax.servlet.Servlet` permettant au conteneur d'assurer le cycle de vie de la servlet.[01]

La mise en place de l'interface (généralement réalisée en étendant `javax.servlet.GenericServlet` ou `javax.servlet.HttpServlet`) fait appel à cinq méthodes :

- la méthode `init()`
- la méthode `service()`
- la méthode `getServletConfig()`
- la méthode `getServletInfo()`
- la méthode `destroy()`

#### 4.3.4. Le cycle de vie :

De la compilation du code à l'arrêt du moteur de Servlets en passant par le traitement des requêtes provenant du client, les Servlets entament un chemin bien précis qui peut se décomposer (de manière très générale) en trois étapes :

- Le chargement.
- Le traitement des requêtes, que l'on peut appeler exécution.
- La destruction.

##### A- Le chargement :

Afin qu'une Servlet puisse accueillir les requêtes venant des clients, nous avons vu dans la section précédente qu'une instance de cette Servlet doit être présente. Le processus de chargement de la Servlet est effectué par le moteur de Servlet. Ce chargement peut être effectué au démarrage du moteur de Servlets ou bien juste au moment où le moteur détermine qu'il a besoin de la Servlet en question.

Tout d'abord, le moteur de Servlets recherche une classe du type de la Servlet à charger à l'endroit où se trouvent les classes des Servlets pour chaque contexte (dont il est détaillé dans le chapitre suivant). Si la Servlet n'est pas déjà chargée, le moteur de Servlets charge la

Servlet en utilisant un chargeur de classe normal à partir du système de fichier local, ou de toutes autre ressource distante (cela dépend des contextes). Une fois que la Servlet est chargée, elle est instanciée et un objet du type de la Servlet chargée est donc présent en mémoire. Une Servlet peut être chargée une fois ou plus par le moteur de Servlets, cela dépend du modèle d'exécution choisi : si la Servlet implémente l'interface `SingleThreadModel` plusieurs instances peuvent être créées en mesure du nombre de requêtes faites par l'utilisateur, si rien n'est précisé par le programmeur (la classe de la servlet n'implémente pas `SingleThreadModel`) une seule instance de la Servlet peut être présente. Plusieurs instances d'une même Servlet peuvent être créées également si plusieurs Servlets identiques possèdent des paramètres d'initialisation différents.

L'initialisation est la dernière étape du chargement d'une Servlet. Une Servlet doit être initialisée avant de pouvoir répondre aux requêtes provenant du client. Le moteur de servlet appelle la méthode `init()` d'une servlet immédiatement après qu'il a construit l'instance de la servlet et avant qu'elle ne traite une requête.

En fonction du serveur et de la configuration de l'application web, la méthode `init()` peut être appelée à l'un des instants suivants :

- lorsque le serveur démarre.
- Lors de la première requête à la servlet, juste avant l'invocation de la méthode `service()`.
- Sur demande de l'administrateur du serveur.

Dans tout les cas, il est garanti que `init()` est appelée avant que la servlet ne traite sa première requête.

La méthode `init()` offre la possibilité qu'une servlet

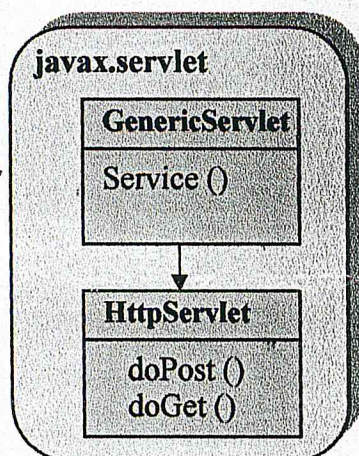
- Lire ses paramètres d'initialisation,
- Spécifier les paramètre d'initialisation (La valeur à la quelle un compteur doit commencer par exemple).

## B- L'exécution :

L'exécution consiste, pour une Servlet, à accueillir les connexions initiées par les clients et à traiter leur demande.

En ce qui concerne les Servlets, il est possible de programmer des applications destinées à servir exclusivement les requêtes utilisant le protocole HTTP, ou bien tous types de requêtes. Dans le premier cas, la Servlet doit hériter de la classe `HttpServlet`, dans l'autre, elle hérite de la classe `GenericServlet`, toutes deux appartenant au package `javax.servlet`.

Pour répondre à une requête générique, le moteur de Servlets appelle la méthode `service()` préalablement redéfinie dans le code de votre Servlet. Il est possible d'effectuer tous les traitements que vous désirez à partir de l'appel de cette méthode et de renvoyer la réponse. A cette fin, un objet de type `ServletRequest` possédant toutes les informations sur la requête et un objet de type `ServletResponse` permettant de transmettre les données produites au cours des traitements effectués par la Servlet au client sont passés en paramètres à cette méthode `service`.

Figure II.16 : héritage de la classe *javax.servlet*

Cependant, la classe la plus utilisable est une classe héritant de la classe *HttpServlet* pour programmer vos Servlets.

Lorsqu'une requête est reçue par le moteur de Servlet, celui-ci examine d'abord de quel type de requête (on parle de méthode) il s'agit. Il en existe peu en ce qui concerne le protocole HTTP. Ce sont :

- GET : afin de demander au serveur de nous transmettre un flux d'information. Cela peut être un fichier statique ou des informations créées dynamiquement (c'est le cas des servlets ou tout autre solution de programmation web côté serveur).
- POST : même utilité que GET mais les paramètres de la requête sont passés directement au serveur au lieu d'être concaténés à l'URL qui identifie la ressource à laquelle on désire accéder.
- HEAD : afin de s'informer sur l'entête renvoyée par le serveur lors d'une requête donnée.
- OPTIONS : permet de connaître les services disponibles sur un serveur.
- PUT : permet de déposer un fichier dans l'arborescence du système de fichier localisé sur la machine faisant tourner le serveur Web.
- DELETE : permet de supprimer un fichier de l'arborescence du système de fichier localisé sur la machine faisant tourner le serveur Web.
- TRACE : permet de "pister" une requête afin de voir ce que voit le serveur.

Une fois que la méthode de la requête est identifiée, le moteur de Servlet appelle la méthode adéquate pour la Servlet à laquelle la requête tente d'accéder. Pour cela il appelle d'abord la méthode *service* de la classe qui se charge alors d'appeler la méthode adéquate. Voici la table de correspondance entre la méthode HTTP et la méthode appelée par le moteur de Servlets :

Méthode HTTP	Méthode appelée par le moteur de servlets
GET	doGet()
POST	doPost()
HEAD	doHead()
OPTION	doOption()
PUT	doPut

Méthode HTTP	Méthode appelée par le moteur de servlets
DELETE	doDelete
TRACE	doTrace()

Tableaux II.2 : table de correspondance entre la méthode HTTP et la méthode appelée par le moteur de Servlets

Deux objets sont passés à ces méthodes lorsqu'elles sont appelées :

- un objet de type *HttpServletRequest* : il possède tous les renseignements sur les paramètres passés à la requête.
- un objet de type *HttpServletResponse* : il permet d'obtenir un flux de sortie pour communiquer la réponse au client.

### C- La destruction :

Pour certaines raisons, le moteur de servlets peut décider s'il doit conserver ou non l'instance (ou les instances) de la Servlet au sein de la machine virtuelle. L'instance d'une Servlet peut être détruite quelques millisecondes après sa création ou après l'arrêt du moteur de Servlets. Rien ne peut contraindre le moteur de Servlets à ne pas détruire l'instance d'une Servlet.

Une fois que le moteur de Servlets a déterminé qu'il n'est plus nécessaire de conserver une instance de la Servlet (par exemple quand le serveur doit libérer de l'espace mémoire, ou lorsque l'exécution du moteur de Servlets est arrêtée), il doit permettre à la Servlet de libérer les ressources qu'elle a acquises au cours de son exécution (que ce soient les buffers utilisés pour l'écriture en sortie, les fichiers lus et écrits ou encore les objets instanciés) et de sauvegarder ce qui doit être persistant. C'est pour cela que le moteur de Servlets appelle la méthode `destroy()` de l'interface Servlet, implémentée par toutes vos Servlets.

Avant cela, tous les threads qui ont été créés au sein de la méthode `service()` de la Servlet doivent soit être terminés, soit dépasser un temps défini par le moteur de Servlets.

Une fois la méthode `destroy()` appelée, le moteur de Servlets ne doit plus router les requêtes provenant des utilisateurs vers la Servlet concernée. Dans le cas où le moteur de servlets pense que l'instance de la servlet puisse être utile à nouveau, il doit créer une nouvelle instance de celle-ci et poursuivre la destruction de la précédente. Une fois que la fin du corps de la méthode `destroy()` est atteinte, le moteur de servlets doit détruire l'instance de la servlet et rendre l'objet éligible par le ramasse-miettes de la machine virtuelle.

#### 4.3.5. Invoquer la servlet :

Les servlets sont enregistrés auprès du moteur de servlets et peuvent en suite être invoqués via une URL depuis un navigateur Web. La manière dont l'enregistrement doit être décrite effectué dépend de l'éditeur du moteur de servlets, et doit être décrite dans la documentation qui accompagne son produit. Généralement, l'URL via laquelle une servlet est invoqué contient la chaîne */servlet* suivie du nom sous lequel la servlet est enregistrée.

Ex : `http://localhost:8080/servlet/nomdu servlet` .

## 4. Java Server Page :

La puissance des servlets est telle qu'ils peuvent être quelquefois encombrants lorsqu'ils génèrent du HTML complexe. On prend par exemple cette ligne de code HTML qui génère le début d'un tableau HTML :

```
out.println("<table border=\"1\" cellpadding=\"4\" cellspacing  
=\"2\" width=\"75%>");
```

cela fonctionne, mais pouvez-vous dire un coup d'œil si tous les guillemets sont fermés? Comparez cette ligne avec la ligne HTML suivante, plus claire, que vous pouvez écrire manuellement :

```
<table border="1" cellpadding="4" cellspacing="2" width="75%">
```

La plupart des servlets contiennent un petit code qui traite les applications logiques, et un plus grand qui traite le formatage de sortie. Il est alors difficile de séparer et de réutiliser les parties du code lorsqu'un format de sortie est requis (par exemple du XML). Pour ces raisons et bien d'autres, les développeurs se sont tournés vers Java Server Page.

### 4.1. Qu'est-ce que Java Server Page ?

Les Java Server Page connues sous le nom *JSP*, constituent une technologie créée par Sun Microsystems et très étroitement liée aux servlets. Le JSP est un environnement de scripts côté serveur qui vous permet de combiner des instructions Java avec du HTML pour produire un contenu dynamique. [38]

Une page JSP se reconnaît avant tout à son extension **.jsp**.

### 4.2. Types d'éléments script JSP :

Afin de d'écrire le code à intégrer dans la page, les JSP offrent des éléments de script qui permettent de préciser les actions Java à effectuer. Il y a 4 types d'éléments :

- les directives qui permettent d'envoyer des messages à l'analyseur de pages,
- les déclarations qui permettent de définir des attributs et méthodes de la Servlet,
- les scriptlets qui sont des fragments de code Java à exécuter,
- les expressions qui sont évaluées et remplacées par une chaîne de caractère émise dans le flux de sortie.

La syntaxe à utiliser est différente pour chacun de ces éléments [38]

```
<%@ pour les directives %>  
<%= pour les expressions %>  
<% pour les scriptlets %>  
<%! pour les déclarations %>
```

- **Les directives JSP** : sont des messages à destination de l'analyseur de la page. Elles s'écrivent avec la syntaxe suivante : `<%@ directive... %>` Les différentes directives disponibles sont : `page` (infos sur la page), `include` (bibliothèques à ajouter pour l'analyse de la page), `taglib` (pour nommer de nouveaux tags).

- **Les JSP expressions** : Contrairement au code Java habituel, l'expression ne doit pas être terminée par un point-virgule. Les expressions sont évaluées et remplacées par leur résultat. Ce sont en général des méthodes qui renvoient un type String. Voici un exemple très simple de JSP qui fait appel à la date et heure du moment de l'exécution :

```
<!-- date.jsp -->
<HTML>
<HEAD>
<TITLE>Affichage de la date avec une JSP</TITLE>
<%@ import="java.util.Date" %>
</HEAD>
<BODY>
<H1>Accès à la date à l'aide de JSP</H1>
La date du jour est <%= new Date() %>.
</BODY>
</HTML>
```

Il ne faut néanmoins pas confondre écriture et exécution. Si les JSP semblent différents des Servlets il n'en est rien. Une JSP est tout d'abord prétraitée par le serveur qui construit une Servlet équivalente si elle n'existait pas déjà. Cette Servlet est ensuite compilée puis exécutée. Contrairement à ce que l'on pourrait parfois croire, une JSP est une Servlet à l'exécution !

- **Les JSP scriptlets**

Les scriptlets permettent de générer du code Java dans la Servlet. Dès qu'une structure de contrôle (If, For, While) est nécessaire, un scriptlet doit être écrit dans la JSP entre les tags `<%et %>`. Les scriptlets ont accès aux variables suivantes :

- **request** pour l'objet `javax.servlet.http.HttpServletRequest` de la Servlet.
- **response** pour l'objet `javax.servlet.http.HttpServletResponse` de la Servlet.
- **out** pour l'objet `java.io.PrintWriter` de la Servlet.
- **in** pour l'objet `java.io.BufferedReader` de la Servlet.

On pourra écrire par exemple :

```
<%
String sexe = request.getParameter( "Sexe" );
String nom = request.getParameter( "Nom" );
if ( sexe.equals("M"))
out.println( "Bonjour Monsieur " + nom );
else
out.println( "Bonjour Madame " + nom );
%>
```

Il est possible d'effectuer des déclarations de variables dans un scriptlet. Dans ce cas, la variable est locale à la méthode de service de la Servlet. Elle sera donc réallouée à chaque invocation de cette méthode (par exemple si le client appuie sur le bouton Reload de son Browser).

- **Les JSP déclarations**

Les déclarations permettent de définir les attributs et méthodes de la Servlet qui sera générée. Les objets déclarés sont partagés par tous les utilisateurs.

Contrairement à une déclaration dans une scriptlet, une variable déclarée ici est globale à la Servlet. Elle n'est donc allouée qu'une seule fois, à la création de cette Servlet.

#### 4.5. Les solutions de substitution

D'autres solutions que les Servlets permettent de réaliser des applications similaires. En effet, les développeurs n'ont pas attendu l'arrivée de ces dernières pour automatiser le traitement de données côté serveur en vue de les communiquer à des utilisateurs. Voici une liste des principales techniques pouvant se substituer aux Servlets :

- CGI (pour Common Gateway Interface) : c'est un ensemble de règles permettant de faire communiquer un serveur Web avec des applications externes. Cette possibilité a donc été rapidement utilisée pour proposer un contenu dynamique. Les scripts CGI peuvent être écrits dans n'importe quel langage (comme le C, le Perl, le Python, etc.).
- PHP (pour Pretty Hypertext Processor) : c'est un langage permettant d'écrire des scripts qui seront ensuite interprétés par un logiciel (on appelle PHP la réunion des deux). C'est un logiciel libre qui en est aujourd'hui à la version 4 et qui est un des plus employés. C'est une technologie très portable.
- ASP (pour Active Server Pages) : la solution proposée par Microsoft. C'est une norme qui permet le développement d'applications Web côté serveur sur la plate-forme Microsoft (système d'exploitation Windows et serveur Web IIS). Comme la norme CGI, ASP permet de développer des scripts en utilisant le langage de programmation de son choix. Cela n'est que théorique, et malgré le développement anecdotiques de support ASP pour le langage Perl en utilisant Apache par exemple, l'indépendance vis à vis du langage, du serveur Web et du système d'exploitation n'est pas une réalité. ASP dans sa version 4 (nommée ASP+) est intégrée à la plateforme .NET.
- Les solutions propriétaires comme Cold Fusion ou WebDev (de PC Soft) qui comprennent tous les outils de la chaîne de développement d'applications Web côté serveur sont assez utilisées mais possèdent les inconvénients de toute architecture propriétaire. Cela dit ces solutions proposent également leur lot d'avantages, comme la gestion de flux de texte très évoluée pour Cold Fusion et le développement rapide pour WebDev.[23]

#### 5. L'accès à la base de donnée :

##### 5.1. ODBC (*Open DataBase Connectivity*):

###### 5.1.1 Présentation:

ODBC signifie *Open DataBase Connectivity*. Il s'agit d'un format défini par Microsoft permettant la communication entre des clients bases de données fonctionnant sous Windows et les SGBD du marché.

Le gestionnaire ODBC est présent sur les systèmes Windows. Il existe toutefois des implémentations sur d'autres plates-formes, notamment des plates-formes UNIX/Linux. Sous Windows le gestionnaire ODBC est disponible dans le panneau de configuration sous l'icône suivant:



La technologie ODBC permet d'interfacer de façon standard une application à n'importe quel serveur de bases de données, pour peu que celui-ci possède un driver ODBC (la quasi-totalité des SGBD possèdent un tel pilote, dont tous les principaux SGBD du marché).[06]

### 5.1.2. Inconvénients de la technologie ODBC :

Bien que ODBC permette un interfaçage avec des bases de données indépendamment du SGBD, cette technologie reste une solution propriétaire de Microsoft.

Cela se traduit par une dépendance de la plateforme (ODBC ne fonctionne que sur les plateformes Microsoft Windows). D'autre part, ODBC est fortement lié au langage C (utilisation de pointeurs), et ODBC utilise des paramètres non standards, ce qui le rend difficile à mettre en oeuvre directement dans les programmes.

### 5.1.3. Qu'est-ce qu'un DSN ?

ODBC permet de relier un client à une base de données en déclarant une source de données (correspondant généralement à une base de données) dans le gestionnaire ODBC (communément appelé *administrateur de source de données ODBC*). La source de données peut être aussi bien une base de données qu'un fichier Access, Excel ou bien même un fichier.

On appelle donc *DSN (Data Source Name)* la déclaration de la source de données qui sera accessible par l'intermédiaire de ODBC.

## 5.2. JDBC (Java DataBase Connectivity) :

### 5.2.1. Présentation :

Le langage Java contient depuis la version 1.1 une API adaptée à la connexion avec les bases de données nommée JDBC. L'avantage de cette API est qu'elle a été construite indépendamment de toute base de données offrant ainsi une portabilité sans équivalent : Le langage Java qui est interprété peut s'exécuter sur toute architecture et l'API JDBC étant indépendante du SGBD s'utilise de la même manière pour toute base de données.

Si ODBC offre une couche d'abstraction universelle indépendante de la base de données, il n'est, par contre, pas indépendant de la plate-forme (MS Windows). JDBC par contre offre au développeur l'abstraction dans les deux directions : base de données et plate-forme système.

Ecrire une application Java de connexion à une base de données nécessite le JDK (Java Development Kit) développé par le constructeur SUN pour permettre la compilation des



programmes Java, un serveur de bases de données pour répondre aux requêtes et un Driver particulier propre à chaque SGBD.

### 5.2.2. Qu'est ce que JDBC ?

JDBC est une API fournie avec Java permettant l'accès à n'importe quelle base de données à travers un réseau. Le terme JDBC signifie *Java DataBase Connectivity* et permet d'écrire facilement des applications Java qui interagissent avec une base de données en respectant le principe de Java : « un seul code + une seule compilation = parfaite portabilité ». JDBC permet notamment de résoudre le problème lié à tous les Intranet actuels : comment relier la base de données d'entreprise à des pages WEB et ainsi permettre aux clients de visualiser les informations qui les concernent à travers le réseau ? Les avantages de JDBC par rapport aux autres approches (ODBC, appels systèmes etc...) sont avant tout des avantages liés à Java : portabilité sur de nombreux systèmes d'exploitation et de nombreuses bases de données, uniformité du langage de description des applications, des Applets et des accès bases de données et surtout liberté totale vis à vis des différents constructeurs.

Pour permettre l'accès aux bases de données, le JDK fournit le paquetage *java.SQL* qui est mis à la disposition des développeurs pour formuler et gérer les requêtes aux bases de données.[35]

### 5.2.3. Drivers et gestionnaire de drivers :

Pour travailler avec un SGBD particulier il faut disposer de classes qui implémentent les interfaces de JDBC. Ces ensembles de classes sont désignés sous le nom de drivers.

Une de ces interfaces les plus importantes s'appelle Driver. Les classes qui l'implémentent doivent fournir une méthode `connect()` qui renvoie une instance d'une classe qui implémente l'interface *Connection*. Cette instance permettra ensuite de lancer des requêtes vers le SGBD en créant des instances de (classes qui implémentent) l'interface *Statement*.

Le gestionnaire de drivers gère la liste des drivers disponibles. En chargeant plusieurs drivers on peut travailler en même temps avec plusieurs SGBD de types différents.

Quand une classe « Driver » est chargée en mémoire, un bloc *static* de la classe crée une instance de la classe et enregistre le driver auprès du gestionnaire de drivers, instance de la classe *DriverManager*, fournie par JDBC.

Lorsque l'on veut ouvrir une connexion avec une base de données dans un programme, le gestionnaire de drivers essaie d'effectuer la connexion avec chacun des drivers qui se sont enregistrés et utilise le premier qui fonctionne pour la base de données. Les drivers sont testés dans l'ordre de leur enregistrement par le gestionnaire de drivers. On peut changer cet ordre en donnant les noms des drivers dans une "propriété" Java `jdbc.drivers` (noms de drivers séparés par des « : »).

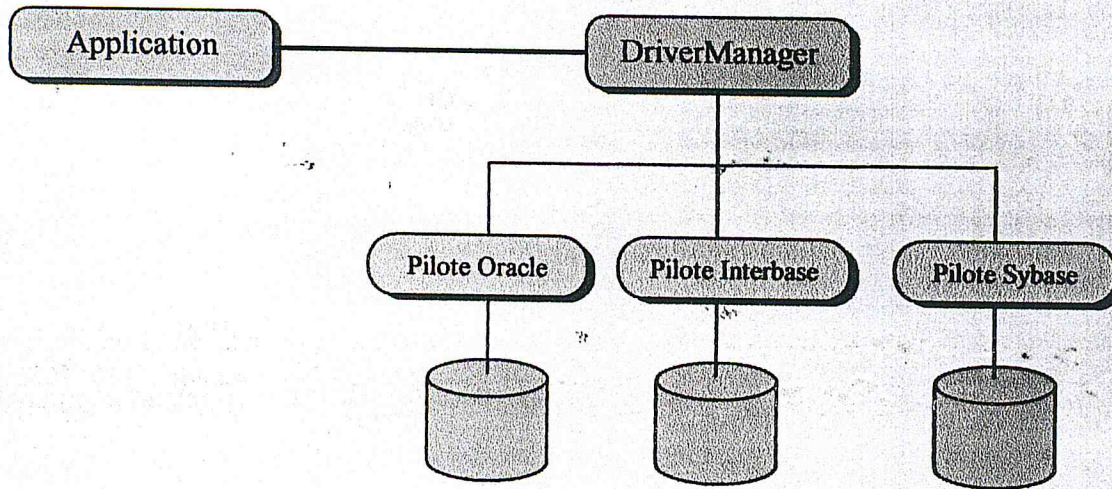


Figure II.17: JDBC cache à l'application les spécificités d'implémentation de chaque base de données

5.2.4. Types de drivers :[40]

Il existe plusieurs types de drivers habituellement numérotés de 1 à 4:

1. **Le pont JDBC-ODBC** permet l'accès aux bases de données en passant par les drivers ODBC (standard Microsoft). Les appels JDBC sont traduits en appels ODBC. Presque tous les SGBD peuvent être accédés par un driver ODBC.
2. **Driver qui fait appel à des fonctions natives non Java** qui permettent de travailler avec le SGBD que l'on veut utiliser (le plus souvent fournies avec le SGBD).
3. **Driver qui permet l'utilisation d'un protocole pour travailler avec un service dit "middleware"** d'accès à plusieurs SGBD ou autres sources de données. Ce protocole est indépendant des SGBD. Le serveur "middleware" accèdera par un moyen quelconque aux différents SGBD (par exemple, JDBC s'il est écrit en Java).
4. **Driver qui utilise le protocole réseau du SGBD.** Il utilise les protocoles réseau intégrés au moteur de bases de données, les pilotes de type 4 conversent directement avec la base de données via des sockets Java. C'est la solution purement Java la plus directe. Ce type proviendra le plus souvent uniquement du fournisseur de base de données, Il est écrit entièrement en Java.

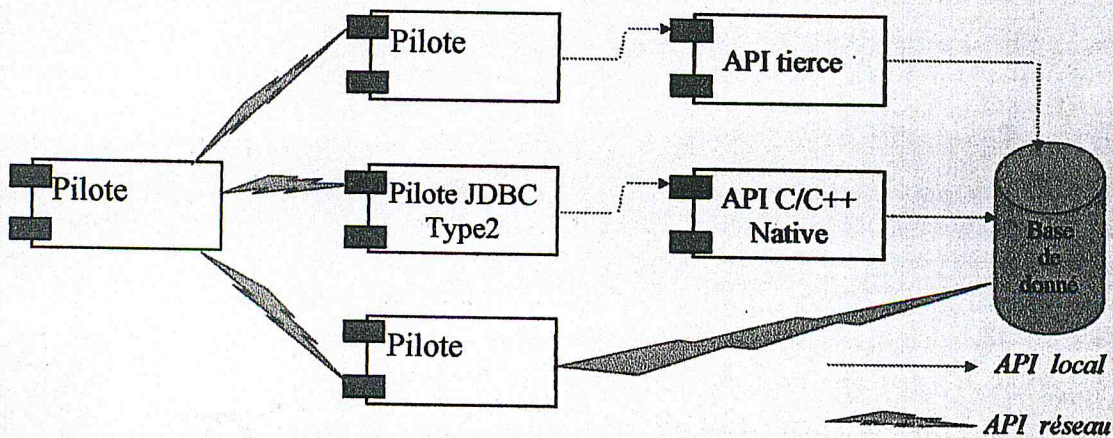


Figure II.18 : Classification des pilotes du JDBC.

### 5.2.5. Structure d'une application JDBC :

Chaque programme Java souhaitant utiliser l'API JDBC doit tout d'abord inclure le paquetage `java.sql` : `import java.sql.*`; . Ensuite, comme pour toutes les entrées-sorties Java, des exceptions sont générées en cas d'erreur. Tous les ordres SQL doivent donc capter l'exception `SQLException` qui est appelée dès qu'un ordre SQL ne se passe pas correctement. Cette classe contient notamment la méthode `getMessage()` qui renvoie le message en clair de l'erreur.

L'étape de développement d'une application JDBC est alors la suivante :

#### Enregistrer le pilote JDBC :

Chaque base de donnée utilise un pilote (driver) qui lui est propre et qui permet de convertir les requêtes JDBC dans le langage natif du SGBD. L'enregistrement de ce pilote doit être effectué dans le `DriverManager Java` au début de chaque application.

Quand une classe de Driver est chargée, elle doit créer une instance d'elle même et s'enregistrer auprès du Driver Manager. Ceci se fait par l'Applet à la méthode suivante :

```
Class.forName("nom du driver");
```

Par exemple, le nom du pilote fourni par le constructeur SUN pour effectuer un pont JDBC-ODBC est `sun.jdbc.odbc.JdbcOdbcDriver`. L'ordre d'enregistrement aura donc la forme :

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

On notera que certains compilateurs refusent cette notation et demandent l'invocation explicite de la méthode `newInstance()` de la manière suivante :

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver").newInstance();
```

#### Etablir la connexion avec la base de données :

Une fois le pilote enregistré, il est alors possible d'établir la connexion avec la base de données.

Ceci se fait par la méthode `getConnection()` du Driver Manager qui renvoie un objet de type `Connection`. Il est nécessaire de passer 3 arguments à cette méthode : l'URL de la base de données, le nom de l'utilisateur de la base et enfin son mot de passe. Ces trois arguments doivent être fournis sous forme de chaînes de caractères.

Par exemple, dans le cas du pont JDBC-ODBC, le nom de DSN à passer doit être de la forme `jdbc:odbc:monDSN`. Si votre DSN se nomme « base » accessible par l'utilisateur « admin » et le mot de passe « univblida » la connexion se fera par l'ordre :

```
java.sql.Connection con =  
DriverManager.getConnection("jdbc:odbc:base","admin","univblida");
```

Dans le cas d'une connexion interbase avec une base de donnée nommé « base » situé dans un répertoire « PFE » accessible par un utilisateur « admin » avec un mot de passe « univblida » la connexion sera :

```
Connection con = DriverManager.getConnection  
("jdbc:InterBase://localhost/PFE/base.gdb","admin","univblida");
```

### Créer une zone de description de requête :

Pour définir une requête, il est ensuite nécessaire de créer un objet de type *Statement*. Cet objet offre non seulement une zone de description de la requête SQL par elle-même, mais aussi les ordres d'exécution de cette requête.

La création de l'objet *Statement* se fait par la méthode `createStatement()` de l'objet *Connection* précédemment créé :

```
java.sql.Statement stmt = con.createStatement();
```

### Exécuter la requête :

Une fois la zone de description, créée, il est alors possible de la remplir avec une requête SQL puis d'exécuter cette requête.

Deux méthodes de l'objet *Statement* permettent respectivement de définir et exécuter la requête. Il s'agit de la méthode `executeQuery()` qui permet d'exécuter une requête du type `SELECT` et qui renvoie un objet de type *ResultSet* contenant les tuples résultant, et de la méthode `executeUpdate()` qui permet d'exécuter les requêtes du type `CREATE`, `INSERT`, `UPDATE` et `DELETE` et qui renvoie un entier (`int`) indiquant le nombre de tuples traités.

Ces deux méthodes nécessitent un argument de type chaîne qui indique la requête SQL à exécuter. Par exemple,

```
java.sql.ResultSet rs = stmt.executeQuery("SELECT * FROM clients");  
int nb = stmt.executeUpdate("INSERT into clients values('Djilali', 'nehal' " );
```

### Traiter les données retournées :

L'objet *ResultSet* retourné après l'exécution de la méthode `executeQuery()` contient plusieurs méthodes permettant de manipuler l'ensemble résultant de la requête :

Pour parcourir l'ensemble résultat, il n'existe qu'une seule méthode : la méthode `next()`.

Initialement le pointeur est positionné avant le premier tuple, chaque appel à la méthode `next()` fait avancer le pointeur sur le tuple suivant. `next()` renvoie à chaque invocation un booléen permettant de savoir s'il y a encore des tuples disponibles.

```
while (rs.next())  
{  
  // Traitement de chaque tuple  
  ...  
}
```

Il n'est donc pas possible de revenir aux tuples précédents ou de parcourir l'ensemble résultat dans un ordre aléatoire; on commence par le premier, on passe au suivant et ainsi de suite jusqu'au dernier, sans possibilité de retour en arrière. Si un retour arrière est souhaité il est donc nécessaire de refaire un `ExecuteQuery()`.

Pour accéder aux éléments d'un tuple, l'objet *ResultSet* fournit toute une série de méthodes de la forme `getXXX()` permettant de lire le type de données `XXX` dans chaque colonne du tuple courant.

On utilisera par exemple les méthodes `getInt()`, `getFloat()`, `getDate()`, `getLong()`, `getString()` ...etc, la colonne est identifiée soit par son nom que l'on passe en

paramètre sous forme de chaîne de caractère soit par son numéro relatif dans l'ordre des colonnes.

```
int pds = rs.getInt(3); // accède à la 3ème colonne
int prod= rs.getString("PRODUIT"); // accède à la colonne 'PRODUIT'
```

Il faut cependant noter que Java ne fournit pas directement la possibilité de savoir si une colonne SQL est nulle ou pas. JDBC fournit pour cela la méthode `wasNull()` de la classe `ResultSet` qui permet de savoir a posteriori si une colonne était nulle ou pas.

```
int poids = rs.getInt("poids");
if (!rs.wasNull()) System.out.println("le poids est connu");
```

### Instruction SQL paramétrée :

Chaque instruction SQL envoyée au SGBD est analysée par le SGBD pour trouver la meilleure façon de l'exécuter. Avec la plupart des SGBD (dont Interbase) on peut s'arranger pour que le SGBD n'analyse qu'une seule fois une requête si elle est exécutée un grand nombre de fois avec des valeurs différentes pour certaines valeurs considérées comme des paramètres.

JDBC a prévu de profiter de ce type de fonctionnalité par l'utilisation de requêtes paramétrées ou de procédures stockées.

Les instructions paramétrées peuvent avoir des paramètres qui peuvent recevoir des valeurs données par le programme Java. Elles n'ont d'intérêt que si elles sont exécutées plusieurs fois dans le programme avec des paramètres différents. La stratégie d'exécution d'une requête ne dépendant pas de ces paramètres, la plupart des SGBD ne choisiront cette stratégie qu'une fois et exécuteront donc plus vite les multiples exécutions de la requête.

Les procédures paramétrées sont associées aux instances des classes qui implémentent l'interface `PreparedStatement` qui hérite de l'interface `Statement`.

Les valeurs des paramètres sont données par les méthodes `setXXX(n, valeur)` où `XXX` désigne le type Java de la valeur. On n'a pas la même flexibilité que pour les méthodes `getXXX()` pour les correspondances entre types Java et SQL. Le driver JDBC traduit les types Java dans les types JDBC donnés par le tableau II.2 avant d'envoyer les données au SGBD. Par exemple, `setShort(3, 28)` passera un `SMALLINT` à la base de données.

Type Java	Type JDBC/SQL
String	<i>VARCHAR (ou LONGVARCHAR)</i>
byte[]	<i>VARBINARY(ou LONGBINARY)</i>
Boolean	<i>BIT</i>
byte	<i>TINYINT</i>
short	<i>SAMLLINT</i>
int	<i>INTEGER</i>
long	<i>BIGINT</i>
float	<i>REAL</i>
double	<i>DOUBLE</i>
java.math.BigDecimal	<i>NUMERIC</i>
java.sql.Date	<i>DATE</i>
java.sql.Time	<i>TIME</i>
java.sql.Timestamp	<i>TIMESSTAMP</i>

Tableau II.3 : conversion Java vers JDBC

### Commit et Rollback :

Par défaut, tout objet *Connection* fonctionne en mode **auto-commit**. Ce qui veut dire, qu'un Commit est effectué automatiquement après chaque ordre SQL. Il est néanmoins possible de repasser en mode manuel par l'appel à la méthode **setAutoCommit()** de l'objet *Connection*. Cette méthode admet un seul argument booléen indiquant si oui ou non le mode est automatique.

```
con.setAutoCommit(false)
```

Une fois mis en mode manuel, c'est à l'application de solliciter le **commit** ou le **rollback** par les méthodes **commit()** et **rollback()** de l'objet *Connection*. Ceci est notamment très intéressant lorsqu'il est nécessaire de valider tout un groupe d'instructions à la fois.

### Fermer les différents espaces :

Tout programme bien construit se doit de fermer les espaces ouverts durant son travail. La méthode **close()** est prévue à cet effet. Elle est définie pour les objets *ResultSet*, *Statement* et *Connection*.

```
rs.close();
st.close();
con.close();
```

## 6. Serveur web et serveur d'application

### 6.1. Serveur web

Un serveur web est un logiciel permettant à des clients d'accéder à des pages web, c'est-à-dire en réalité des fichiers au format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateur distant.

Un serveur web est donc un "simple" logiciel capable d'interpréter les requêtes HTTP arrivant sur le port associé au protocole HTTP (par défaut le port 80), et de fournir une réponse avec ce même protocole.[46]

De plus un serveur web permet d'assurer un certain nombre de fonction comme :

#### 6.1.1. Les besoins de sécurité :

La grande force du Web est sa capacité à rendre toute information disponible immédiatement à des millions d'utilisateur anonyme dans le monde entier.

Cette capacité repose sur des standards ouverts, notamment les protocoles de communication qui s'appuient tous sur TCP/IP.

Ceci peut engendrer des lacunes au niveau de la sécurité des données a deux niveau :

- L'ouverture du réseau interne à l'extérieur permet aux utilisateurs de consulter les pages Web, mais aussi s'introduire de l'extérieur (via TCP/IP).

- Le serveur Web communique au monde extérieur certaines informations, il faut s'assurer qu'il ne permet pas d'avoir accès à des informations confidentielles.

Il est essentiel de protéger l'information contre les menaces. Les serveurs Web possèdent mécanismes de sécurité propre à eux, et surtout support des mécanismes indépendants. Entre autre :

#### 6.1.2. Authentification des utilisateurs :

En contexte Internet, on ne fait pas de distinctions entre les utilisateurs. Tous sont généralement autorisés à accéder aux pages du serveur Web. En revanche, intranets ou extranets imposent la mise en œuvre de notions de profil utilisateur et de listes de contrôle d'accès (ACL ou Access Control List). Les utilisateurs peuvent être identifiés de trois manières différentes :

- Par leur adresse IP : dans ce cas, c'est en réalité la machine de l'utilisateur qui est identifiée, et non l'utilisateur lui-même (ceci à la condition que le réseau n'utilise pas de système d'adressage dynamique comme DHCP) ;
- Par un nom utilisateur et un mot de passe associé : une liste d'utilisateurs habilités est maintenue au niveau du serveur Web ;
- Par leur certificat qui peut, par exemple, être stocké sur une carte à puce.

La deuxième et la troisième solutions sont les plus sûres, à condition que les utilisateurs soient responsabilisés sur les choix de leur mot de passe, et la nécessité de garder confidentiels leurs moyens d'authentification.

#### 6.1.3. Confidentialité des informations échangées :

TCP/IP n'est pas par essence un protocole très sécurisé. En effet, toutes les trames peuvent être interceptées de manière totalement transparente au niveau de n'importe quel ordinateur connecté au réseau.

Pour garantir une sécurité efficace, il est essentiel de crypter les informations échangées (au moins les mots de passe). Les mécanismes de cryptage sont basés sur un système d'échange de certificats entre le serveur Web et le navigateur Web (poste client), leur intégration dans le serveur Web et le navigateur (à partir des versions 3 de Netscape et d'Internet Explorer), il ne demande pas d'effort supplémentaires de développement. Il faut noter, que les temps de cryptage et décryptage des informations pénalisent quelque peu les performances.

#### 6.1.4. Journalisation des événements :

Tous les serveurs Web génèrent des fichiers *log*, qui permettent de garder une trace des différents événements survenus.

#### 6.1.5 Les différents serveurs Web sur marché :

Le choix disponible en matière de serveur Web est très large. On constate néanmoins que trois serveurs représentent à eux seuls près de 90 % de la base installée.[45]

Serveur	Part des serveurs en activité sur Internet (%)
Apache	63,34
IIS (Microsoft)	26,62
IPlanet	2,83
Zeus	1,27

Tableau II.4 : Statistiques de Netcraft ([www.netcraft.com](http://www.netcraft.com)) sur les serveurs Web.

### Le projet Apache :

Apache est depuis plusieurs années le serveur Web le plus utilisé sur Internet. Le projet Apache est issu d'un effort de développement collaboratif destiné à créer un serveur HTTP robuste, fiable et gratuit. Sous licence GNU, Apache peut être téléchargé gratuitement sur le site <http://www.apache.org>.

Ce serveur est disponible sur Unix et Windows. Sa fiabilité n'est plus à démontrer, mais son administration nécessite une bonne connaissance de la structure des fichiers de paramètres. Elle reste difficilement réalisable par un néophyte, d'autant qu'il n'existe pas de console graphique de configuration à distance.

Néanmoins, les configurations associant Linux et Apache gagnent chaque de nouvelles parts de marché. Il faut dire que les attraits de cette association sont nombreux :

- Faible coût à l'achat,
- Fiabilité et performances inégalées,
- Large spectre d'interfaces de programmation

### L'offre de Microsoft :

Dans la catégorie des serveurs commerciaux, IIS (Internet Information Serveur) est leader avec plus de 25 % de parts de marché. Cette offre est bien sûr exclusivement basée sur le système d'exploitation Windows. Les serveurs Windows ont l'avantage d'être simple à administrer mais ont la réputation d'être nettement moins fiables que les serveurs Unix.

Dans le cas des sites à forte fréquentation ou pour lesquels un niveau de sécurité élevé est requis, Unix reste le choix de référence. Néanmoins, de nombreux progrès ont été accomplis par les équipes de développement de Microsoft et IIS offre aujourd'hui de bons rapports performances/prix.

IIS est aujourd'hui intégré au système d'exploitation professionnel de Microsoft. Simple à administrer, il offre par ailleurs toutes les fonctions attendues d'un serveur Web. Il intègre en standard une interface de programmation par script (VBScript ou JavaScript) : Active Server Page.



### L'offre d'iPlanet :

La société iPlanet est issue du rapprochement de Sun et de Netscape. Dans le domaine des serveurs Web professionnel, iPlanet Web Server (PSW) occupe la deuxième place derrière IIS de Microsoft.

Le principal intérêt de la gamme Web d'iPlanet dispose d'une interface d'administration graphique de bonne qualité, accessible via un navigateur Web. On notera cependant que cette interface est rarement utilisée par les administrateurs de grandes infrastructures Web qui lui préfèrent la série de scripts en ligne de commande livrée avec le produit.

## 6.2 Serveurs d'applications

### 6.2.1. Présentation :

Le serveur d'application permet de générer une réponse HTML à une requête après avoir effectué un certain nombre d'opérations (connexion à une base de données, à un annuaire LDAP...). Pour le client (un navigateur web en général), il n'y a pas de différence avec une page web statique : il reçoit toujours du HTML, seul langage qu'il comprend. Seule la manière dont la réponse est formée côté serveur change.[29]

Les requêtes, pour le client, ne diffèrent pas non plus. Qu'il souhaite accéder à une ressource statique ou à une application web, il utilise toujours une URL au même format (standard HTTP). C'est donc côté serveur que la distinction doit s'opérer. Le schéma suivant montre le déroulement classique d'une requête vers un serveur d'applications :

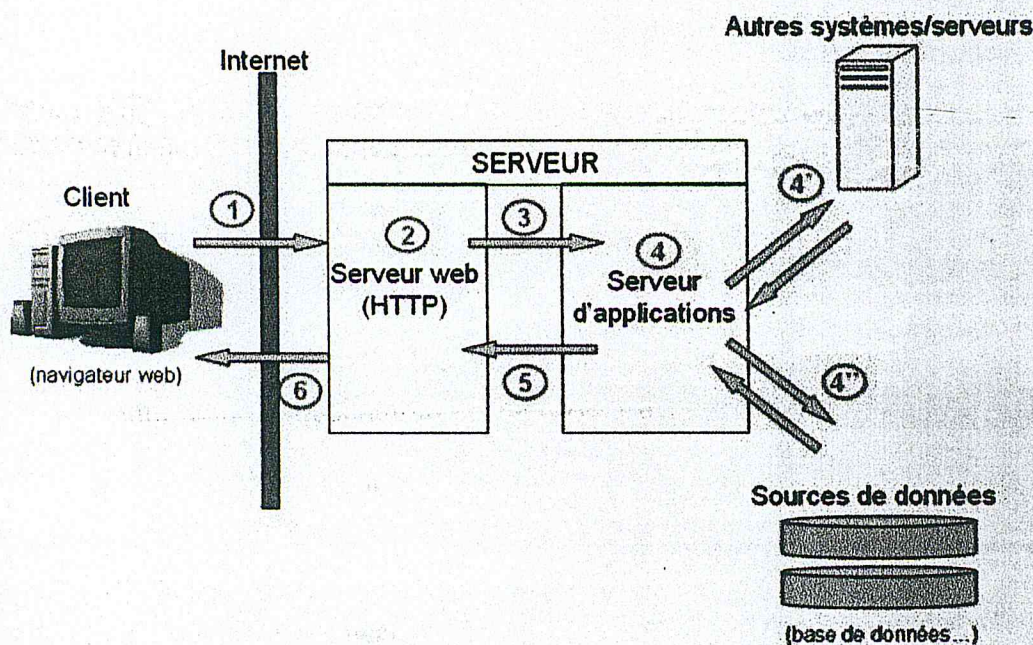


Figure II.19 : le déroulement classique d'une requête vers un serveur d'applications

- 1) Le client émet une requête (i.e. appelle une URL) pour demander une ressource au serveur. Exemple : `http://www.univ_blida.dz/welcome`. Il ne sait pas ici si la réponse qui va lui parvenir est statique (page HTML simple) ou dynamique (générée par une

application web). Dans notre cas, il s'agit d'une application répondant à l'adresse **welcome** sur le serveur *www.Univ\_blida.dz*.

- 2) Côté serveur, c'est le serveur web (exemple : Apache) qui traite les requêtes HTTP entrantes. Il traite donc toutes les requêtes, qu'elles demandent une ressource statique ou dynamique. Seulement, un serveur HTTP ne sait répondre qu'aux requêtes visant des ressources statiques. Il ne peut que renvoyer des pages HTML, des images, ... existantes.
- 3) Ainsi, si le serveur HTTP s'aperçoit que la requête reçue est destinée au serveur d'applications, il la lui transmet. Les deux serveurs sont reliés par un canal, nommé connecteur.
- 4) Le serveur d'applications (exemple : Tomcat !) reçoit la requête à son tour. Il est, lui, en mesure de la traiter. Il exécute donc le partie d'application (la servlet, JSP... etc) auquel est destinée la requête, en fonction de l'URL. Cette opération est effectuée à partir de la configuration du serveur. La servlet est donc invoquée, et le serveur lui fournit notamment deux objets Java (Tomcat est un serveur d'applications Java) exploitables : un représentant la requête, l'autre représentant la réponse. La servlet peut maintenant travailler, et générer la réponse à la demande. Cela peut passer par la consultation de sources de données, comme des bases de données (4" sur le schéma). Ou bien par l'interrogation d'autres serveurs ou systèmes (4' sur le schéma), l'environnement Java web permettant de se connecter à de nombreux systèmes.
- 5) Une fois sa réponse générée, le serveur d'applications la renvoie, par le connecteur, au serveur web. Celui-ci la récupère comme s'il était lui-même allé chercher une ressource statique. Il a simplement délégué la récupération de la réponse, et celle-ci a été générée.
- 6) La réponse est dorénavant du simple code HTML, compréhensible par un navigateur. Le serveur HTTP peut donc retourner la réponse au client.

### 6.2.2. Rôle d'un serveur d'application :

Le serveur d'application est l'environnement d'exécution des applications côté serveur. Il prend en charge l'ensemble des fonctionnalités qui permettent à N clients d'utiliser une même application :

- **Gestion de la session utilisateur** : N clients utilisant une même instance d'application sur le serveur, il est nécessaire que le serveur d'application puisse conserver des contextes propres à chaque utilisateur (par exemple, un panier de commandes). La plupart des serveurs d'application génèrent un identifiant unique pour chaque nouveau client et transmettent cet identifiant lors de chaque échange HTTP par URL longs, variables cachées ou cookies.
- **Gestion des montées en charge et reprise sur incident** : Afin de gérer toujours plus d'utilisateurs, le serveur d'application doit pouvoir se déployer sur plusieurs machines et éventuellement offrir des possibilités de reprise sur incident (même si dans la grande majorité des cas, on se contente d'une gestion des montées en charge au niveau réseau - boîtier de répartition, DNS *round-robin*, *reverse proxy*...).
- **Ouverture sur de multiples sources de données** : C'est le serveur d'application qui rend accessible les données des applications du système d'information. Il doit donc

pouvoir accéder à de nombreuses sources de données. On s'attend également à ce qu'il fournisse des mécanismes performants tels que le *pooling* de connexion base de données.

Le serveur d'application est donc indispensable si l'on souhaite éviter de re-développer l'ensemble de ces fonctionnalités. Les moteurs JSP/Servlets, Cold Fusion, PHP... sont à ce titre des serveurs d'application.

### 6.2.3. Serveur web / serveur d'applications

Dans le schéma en haut de la page, nous avons séparé le serveur web et le serveur d'applications. Ces deux composants sont en effet nécessaires côté serveur, puisqu'ils se complètent : le serveur d'applications ne sait pas traiter une requête HTTP, le serveur web ne sait pas exécuter d'applications !

### 6.2.4. Quelques serveurs d'application

#### Tomcat

Tomcat est la référence d'implémentation des spécifications Java Servlet et JavaServer Pages développées par SUN. Tomcat peut être utilisé à la fois comme serveur HTTP et comme conteneur d'exécution des servlets et des JSP. Ce serveur d'applications Web comporte les avantages suivants : [45]

- il est compatible J2EE ce qui signifie que l'ensemble des développements réalisés pour Tomcat pourront être portés sur des serveurs d'applications commerciaux comme WebSphere, WebLogic, Oracle Application Server.
- les développements se font en Java ce qui assure un niveau élevé de portabilité : il sera possible de déployer le code qui tourne sur Tomcat sur la plupart des systèmes d'exploitation, (Linux, AS400, Solaris, HP-UX, AIX, Windows, Netware, Mac OS X, OS390, etc.). Ce haut niveau de portabilité favorise la réutilisation, atout des technologies objets.
- Tomcat est facile à intégrer avec les serveurs Web les plus connus comme Apache et IIS. Le serveur Web est alors en charge de toute la partie statique du site Web, alors que Tomcat gère la partie dynamique (requêtes sur les servlets et les JSP).

#### Jetty

Jetty est un serveur d'applications Web 100 % Java. Comme Tomcat, il s'agit d'un conteneur d'exécution des servlets et des JSP. Il peut jouer le rôle d'un serveur HTTP, et il peut s'intégrer avec les serveurs Web comme Apache. Il est disponible pour toutes les plates-formes qui supportent Java. Jetty est devenu le conteneur par défaut pour les serveurs d'applications d'entreprise Jboss.

#### PHP

PHP, acronyme de « PHP Hypertext Preprocessor », est un serveur d'applications Web open source destiné à la réalisation de sites Web dynamiques. Ses qualités techniques sont aujourd'hui reconnues par tous. Depuis la sortie majeure de PHP4 en juin 2000, un nombre croissant d'entreprises ont découvert les bénéfices qu'elles pouvaient tirer de l'utilisation de cette plate-forme.

La croissance rapide de PHP est explicable par ses liens avec le logiciel libre et son emploi à l'intérieur d'un quatuor de logiciels connu sous l'acronyme LAMP (Linux/Apache/MySQL/PHP). En plus d'être gratuit et disponible sur la majorité des systèmes d'exploitation, PHP possède quelques qualités qui le distinguent des outils concurrents :

- Créé pour le Web : PHP a été créé pour répondre aux besoins des sites à fort trafic et pour simplifier la vie des développeurs.
- Facile d'utilisation : le gain de productivité des équipes de développement est important.

### 6.2.5. Comparaison avec les serveurs propriétaires

Plusieurs solutions commerciales pour les serveurs d'applications Web existent, les plus connus sont IIS avec la technologie ASP de Microsoft et Coldfusion de Macromedia. Ces technologies sont faciles à utiliser. L'inconvénient majeur de la solution Microsoft est d'être intimement liée à l'environnement Windows. Les serveurs d'applications Web supportant les technologies JSP/Servlet adhèrent à la philosophie de l'architecture Java qui prône le « Write Once, Run Anywhere ». Au lieu d'être restreinte à une seule plateforme, ces serveurs peuvent évoluer sur n'importe quelle plate-forme et être supportés par une grande variété d'outils de divers producteurs, d'où un grand intérêt et une large implication de la communauté de l'open source.

On peut consulter [www.webperformanceinc.com/library/ServletReport](http://www.webperformanceinc.com/library/ServletReport) pour obtenir des données supplémentaires de comparaison.

# Chapitre III.

## Conduite d'une application Web

### 1. Introduction

Les applications de l'informatique deviennent de plus en plus diverses et s'infiltrent dans presque tous les domaines de la vie quotidienne. Il est donc évident que la caractéristique dynamique la plus importante de la plupart des applications est leur fiabilité.

La fiabilité d'un système quelconque (pas seulement d'une application) dépend de la qualité de la conception, de la qualité de la réalisation et de la fiabilité des composants qui constituent le système. Pour obtenir un bon degré de fiabilité, on devra comprendre l'environnement dans lequel l'application opère et préparer des spécifications qui définissent, dans la mesure du possible, le rôle de l'application dans cet environnement. [46]

### 2. Le cycle de vie d'une application:

Il faut un temps considérable pour développer une grande application qui d'ailleurs, sera utilisée pendant très longtemps. On identifie donc un certain nombre d'étapes distinctes dans cette période de développement et d'utilisation ; ces étapes constituent le cycle de vie de l'application.

C'est sans doute Royce (1970) qui le premier a proposé un modèle du cycle de vie. Depuis, de nombreuses améliorations et modifications y ont été apportées. L'esprit de ces divers modèles peut cependant être résumé de la façon suivante [17] :

1. *Analyse et définition des besoins* : Les fonctionnalités du système et les contraintes sont établis en consultant les utilisateurs. Elles doivent être définies de façon compréhensible à la fois par les utilisateurs et par l'équipe de développement.
2. *Conception du système et du logiciel* : En partant de l'analyse des besoins, on décompose le système en deux parties : matériel et logiciel. Ce processus est appelé conception du système. C'est le processus qui consiste à représenter les diverses fonctions du système d'une manière qui permettra d'obtenir rapidement un ou plusieurs programmes réalisant ces fonctions.
3. *Réalisation et tests unitaires* : Lors de cette étape, on réalise un ensemble d'unités de programme écrites dans un langage de programmation exécutable. Les tests unitaires permettent de vérifier que ces unités répondent à leurs spécifications.

4. *Tests du système* : On intègre les unités de programme et on réalise des tests globaux pour être sûr que les besoins logiciels ont été satisfaits. Le système alors est livré au client.
5. *Utilisation et maintenance* : Normalement (mais pas nécessairement) ceci est la plus longue étape de cycle de vie. L'activité de maintenance consiste à corriger les erreurs qui n'ont pas été découvertes lors des étapes antérieures du cycle de vie, à améliorer la réalisation des unités du système et à augmenter ses fonctionnalités au future et à mesure que de nouveaux besoins apparaissent.

### 3. La phase Etude :

Les problèmes qui sont soumis aux ingénieurs développeurs sont souvent d'une complexité extrême. Il peut être très difficile de comprendre la nature exacte d'un problème, particulièrement si le système est nouveau et qu'aucun système non automatique n'existe pour servir de modèle au logiciel à développer. Le processus visant à établir quelles fonctionnalités le système doit fournir et les contraintes auxquelles il sera soumis s'appelle l'analyse et la définition des besoins. C'est, en principe, la première grande étape du cycle de vie du logiciel.[43]

#### 3.1. Le cahier de charge :

La description précise des besoins concernant un système logiciel est appelée cahier de charge. Ce cahier est défini par yeh et zave (1980) de la façon suivante :

Un ensemble de propriété ou de contraintes, décrites de façon précise, qu'un système logiciel doit satisfaire.

Un cahier de charge permet de validé la conception c'est-à-dire déterminer si les contraintes et les propriétés spécifiques sont satisfaites et si la conception est une solution acceptable au problème donné.[17]

#### 3.2. Définir le périmètre fonctionnel en corrélation avec les objectifs poursuivis :

Dés la première phase du projet, il est indispensable de définir quel est le périmètre fonctionnel des système conçus, ceux-ci doivent être en relation avec les objectifs décrit dans le cahier de charge (objectifs organisationnels, objectifs managériaux, objectif économique).

Le périmètre fonctionnel correspond aux entités interne et/ou externe à l'organisme qu'il conviendra de traiter avec le projet. Il est ensuite associé à un ensemble de processus et d'activités, eux-mêmes liés à des ressources matérielles, immatérielles et humaines.

La définition d'un tel périmètre doit être validé par la direction fonctionnelle responsable, ce périmètre fonctionnel doit être intégré dans les documents associés à la description du projet. Il forme à ce titre un chapitre du Plan d'Assurance Qualité (PAQ).[43]

**3.3. Définir le champ d'application :**

Dans la même logique, il est essentiel de positionner le champ d'application du projet. Plusieurs question doivent trouver des réponses dès cette phase : recherche-t-on des solutions de capitalisation d'information, d'échanges, de travail en groupe ou encore de modélisation et d'application de processus ?

Recherche-t-on encore des solutions intégrées devant harmoniser des processus avec des partenaires sous-traitants ?

Ce travail préalable permet de chef de projet de sélectionner les membres des différents entité de travail : groupe de utilisateur, groupe technique,... etc.[46]

**4. Conception :**

- Définition de l'architecture applicative
- Définition de l'architecture logicielle
- Définition de politique de sécurité
- Création des MCD, MCT, MLD, MOT, MOPT, MPD

- Organisation des tests
- Passation
- Analyse
- Rapport



- Rassemblement de la matière
- Formalisation des spécifications

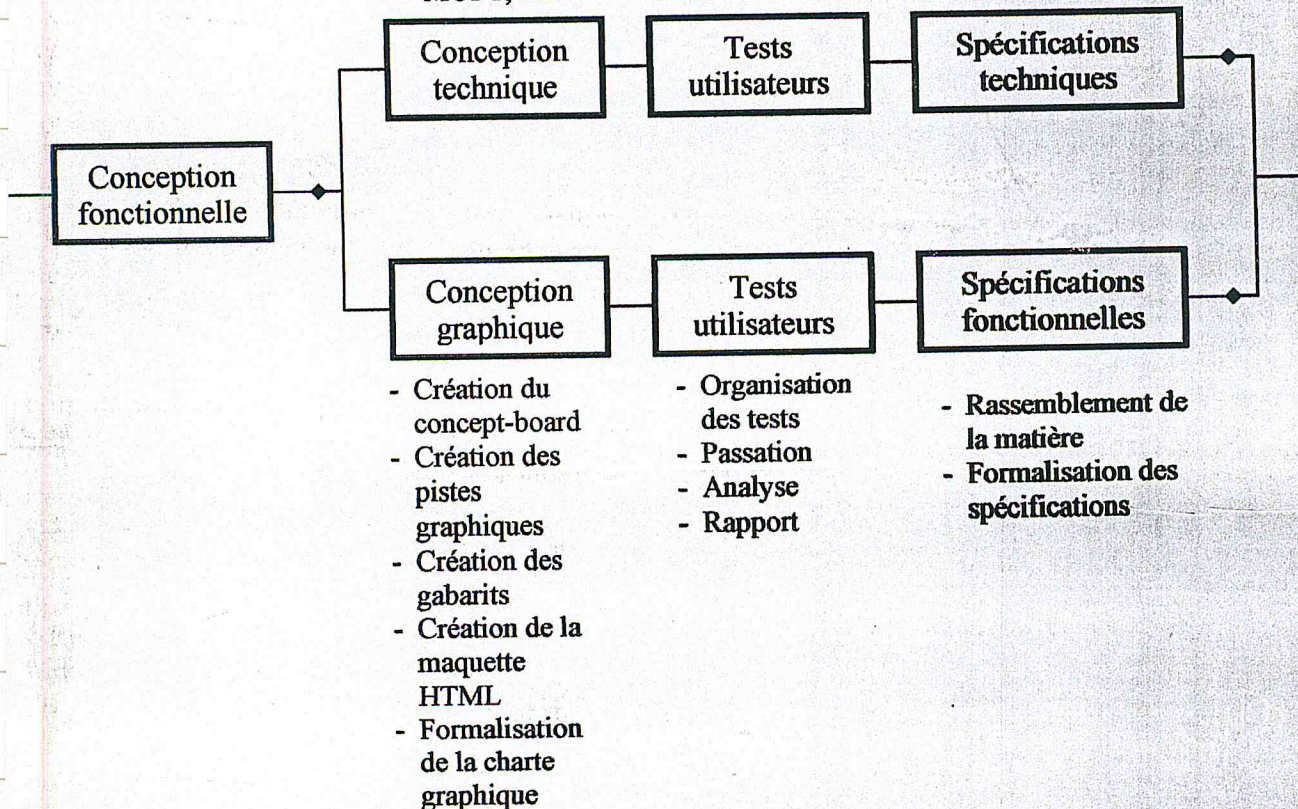


Figure III.1 : Les différentes étapes de la conception.

#### 4.1. Conception fonctionnelle :

Il y a quelques années, l'espace de conception fonctionnel consistait essentiellement à créer le rubriquage du site.

Aujourd'hui, la description des processus métier devient le cœur de la démarche, participant à la lente intégration du projet web au système d'information. Il n'en reste pas moins que les approches éditorial et par processus sont complémentaires et répondant à un seul objectif : la satisfaction de l'utilisateur.

##### 4.1.1. Approche éditorial :

Le dossier de rubriquage et la charte éditoriale sont souvent les premières pierres des projets web. Ils permettent de clarifier les idées du groupe de projet sur le contenu et la forme du site.

##### 4.1.2 Créer le dossier de rubriquage :

Cette tâche consiste à créer, dans un premier temps, l'arborescence générale et le premier niveau de rubriquage (rubrique et sous-rubrique) puis, dans un deuxième temps, à définir les contenus et les fonctionnalités du site. Le niveau de détail exigé est important puisque le but est de décomposer chaque page ou composant en titre, intertitre, texte courant ... et pour chacun d'entre eux de déterminer un calibrage ainsi que des règles de gestion associées.

##### 4.1.3 Approche par processus :

La plus part des projets web ont désormais pour objectif de réaliser des gains de productivité en automatisant certains processus stratégiques.

Pour y arriver, l'organisme doit apprendre à se connaître en analysant les processus constitutifs de son métier, puis les optimiser et, en fin de compte, les rendre disponibles en ligne.

Pour la plus part des projets web, l'approche par processus est complémentaire avec l'approche éditoriale. [46]

#### 4.2. Conception graphique :

Les premiers sites Internet étaient très proches des plaquettes commerciales. Il s'agissait souvent d'une accumulation logique de pages statiques dont l'objectif était d'impressionner l'internaute par la qualité de son graphisme. Puis les premières études sérieuses sur le comportement des utilisateurs ont prouvé que le web était différent des autres médias et qu'il possédait ses contraintes propres.

Commença alors une guerre de tranchées entre aficionados des beaux sites en Flash et les défenseurs des sites utiles construits en trois colonnes avec des liens en bleus.

Aujourd'hui, pour concilier le meilleur des deux camps, une solution efficace consiste à confier la conception graphique à une équipe spécialisée en infographie.



### 4.3. Spécification fonctionnelles :

Les spécifications fonctionnelles ont pour objectif de décrire le plus précisément possible la future application. Le but est de donner les moyens au prestataire qui réalise les développements d'appréhender rapidement le fonctionnement général et de comprendre chaque détail de chaque fonctionnalité. Ainsi, il peut développer l'application en toute autonomie.

### 4.4. Conception technique :

La conception technique est l'une des étapes fondamentales du projet. Si elle est menée avec intelligence, elle permettra de réaliser, tout au long de la vie du site, des économies très substantielle.

En effet, contrairement aux premiers sites dont la maintenance s'avérait plus coûteuse sur quelques années que leur re-développement, les projets web actuels ont une obligation de rentabilité et de pérennité. Cet objectif peut être atteint grâce à une conception technique modulaire.

Celle-ci se déroule en trois étapes : la formalisation de l'architecture, la modélisation puis la rédaction des spécification techniques détaillées. [46]

#### 4.4.1 Architecture :

L'architecture d'une solution web est décrite à travers son architecture applicative, technique et matérielle. Ces différentes vues de la plus conceptuelle au plus concrète.

##### • L'Architecture applicative :

L'architecture applicative répond à un double enjeu : aider les concepteurs d'un logiciel à en maîtriser sa complexité : garantir la maintenabilité et l'évolutivité de la solution.

Le modèle le plus connu est l'architecture à trois niveaux.

Dans un souci de réutilisation et de maintenance, le modèle 'cinq couche' tend à se développer. Il comprend :

- une couche présentation.
- Une couche application.
- Une couche composants métier.
- Une couche accès aux données.
- Une couche de stockage des données.

##### • L'Architecture logicielle :

L'architecture logicielle décrit les différentes applications, leurs interdépendances et les règles associées. Pour un projet web, elle comprend en général :

- un niveau incontournable, à savoir l'infrastructure de base :
  - Pare-feu.
  - Serveur http.

- Serveur de messagerie.
- Serveur d'application.
- Moteur de recherche.
- Connecteurs base de données.
- Base de données.
- Système d'exploitation...
- un niveau spécifique (exemple pour un site institutionnel) :
  - Serveur de publication.
  - Serveur de contenu, etc.

#### • L'Architecture matérielle :

L'architecture matérielle décrit les ordinateurs (poste client, serveur etc.) et les réseaux sollicités par la solution. Elle permet de dimensionner précisément le budget et d'effectuer des choix stratégiques tel que l'hébergement.

#### • La sécurité :

Les règles de sécurité précisent trois niveaux : la sécurité physique (accès à la salle machine, alimentation électrique, sauvegarde, reprise sur incidents...), la sécurité logique (login / mot de passe, cryptographie, pare-feu), la sécurité organisationnelle (attribution / modification / suppression des droits, procédure, mesure de vérification, test, gestion des sessions...).

#### 4.4.2. Modélisation :

En France, la méthode de modélisation la plus utilisée reste Merise (relationnel) bien que des méthodes orientées objet telle qu'UML (Unified Modeling Language) soient en passe de devenir le standard, grâce, notamment, au développement de J2EE.

Chaque méthode correspond à une approche conceptuelle et à une vision différente de la programmation. Merise est adaptée à des structures de données relationnelles et à une programmation procédurale. UML est adaptée à des langages et des structures de données objet.

Il n'y a pas donc une bonne ou une mauvaise méthode, il y a en revanche des méthodes plus ou moins adaptées à un type de projet.

Présenter une méthode telle que Merise en quelques lignes n'est pas possible. Nous ne survolons ici que les livrables essentiels : MCD, MCT, MOPT, MLD, MPD. Le but est de comprendre à quoi ils représentent un choix stratégique pour le projet.

Cycle d'abstraction	Domaine des données	Domaine des traitements
Conceptuel	MCD	MCT
Logique	MLD	MOT
Physique	MPD	MOPT

Tableau III.1 : Vue d'ensemble et articulation des différents modèles de Merise.

- **Le modèle conceptuel des données :**

Le modèle conceptuel des données (MCD) permet de formaliser une représentation claire des données du domaine considéré. Il définit en outre les relations fonctionnelles existant entre elles. Ce modèle est de loin le plus important. Il permet aux intervenants fonctionnels et techniques de se mettre d'accord sur une même présentation de la réalité. Les éléments utilisés pour la formalisation d'un MCD sont les suivants :

- Entité type : définition d'entité (objets physiques ou abstraits) qui ont des caractéristiques comparables.
- Relation type : définition d'une association qui lie plusieurs entités type. Signification d'un lien entre deux ou plusieurs types d'entité.
- Propriété type : définition d'une caractéristique d'une entité ou d'une association. Une propriété type est elle-même caractérisé par un type (chiffre ou texte...) et une longueur. L'ensemble des propriétés type du MCD compose le dictionnaire des données.
- Identifiant : propriété type ou concaténation de propriété types qui permet de distinguer une entité parmi toutes les autre dans une entité type.
- Cardinalité minimale : nombre minimal de fois où une entité est concernées par l'association.
- Cardinalité maximale : nombre maximal de fois où une entité est concernées par l'association.

- **Le modèle conceptuel des traitements :**

Le modèle conceptuel des traitements (MCT) permet quant à lui de formaliser les traitements en fonction des événements extérieur sans s'intéresser à l'organisation qui régira ces traitements. Il est moins employé que MCD. Les éléments utilisés pour la formalisation d'un MCT sont les suivants [50] :

- Événement : il peut être interne ou externe au SI (système d'information) ; il s'agit d'un déclencheur pour le lancement d'une opération ou le résultat d'une opération à destination du monde extérieur.
- Synchronisation : règle qui indique les événements et l'enchaînement de ces derniers nécessaire au lancement d'une opération. Il s'agit d'une expression logique composée essentiellement de OU et ET.
- Opération : liste des actions à réaliser si la synchronisation associée est réalisée. L'ensemble des actions de l'opération s'exécute sans interruption ni attente d'événement.
- Emission : expression logique qui indique selon le résultat de l'opération quels événements internes au SI sont créés.

- **Le modèle opérationnel des traitements :**

Le but du modèle opérationnel des traitements (MOPT) est de préparer les développements. Pour ce faire, il décompose chaque application en composants dont il précise les données et les traitements (présentation et appel du traitement, information en entrée et de sortie, résultat, données internes au traitement, algorithme)

Le MOPT est obtenu suite à une analyse descendante (décomposition du résultat en élément de plus faible granularité) ou ascendante (factorisation des composants utilisés

par l'application). Cette dernière se rapproche à la logique objet. Le MOPT est important car il conditionne directement la maintenabilité et l'évolutivité de la solution. [50]

• **Le modèle logique des données :**

Le modèle logique des données (MLD) se situe à un niveau intermédiaire entre le modèle conceptuel et le modèle physique des données. La réalisation d'un schéma logique des données relationnelles permet d'obtenir le nom des tables de la base et le nom des champs. [50]

• **Le modèle physique des données :**

Le modèle physique des données (MPD) est l'implantation concrète du MCD dans la base de données cible.

**4.5. Spécification technique :**

Les spécifications techniques ont pour objectif d'industrialiser les développements en précisant le cadre et les règles. Elles doivent être les plus précises possibles tout en restant digestes.

**4.6. Tests utilisateurs :**

Les tests utilisateurs sont essentiels dans la réussite du projet Web. Les sites étant créés pour les utilisateurs, ce sont eux qui peuvent le mieux dire si leurs préoccupations ont été bien comprises.

Les tests utilisateurs peuvent être utilisés à chaque phase du projet, cependant, ils sont incontournables lors de la conception.

**5. La réalisation**

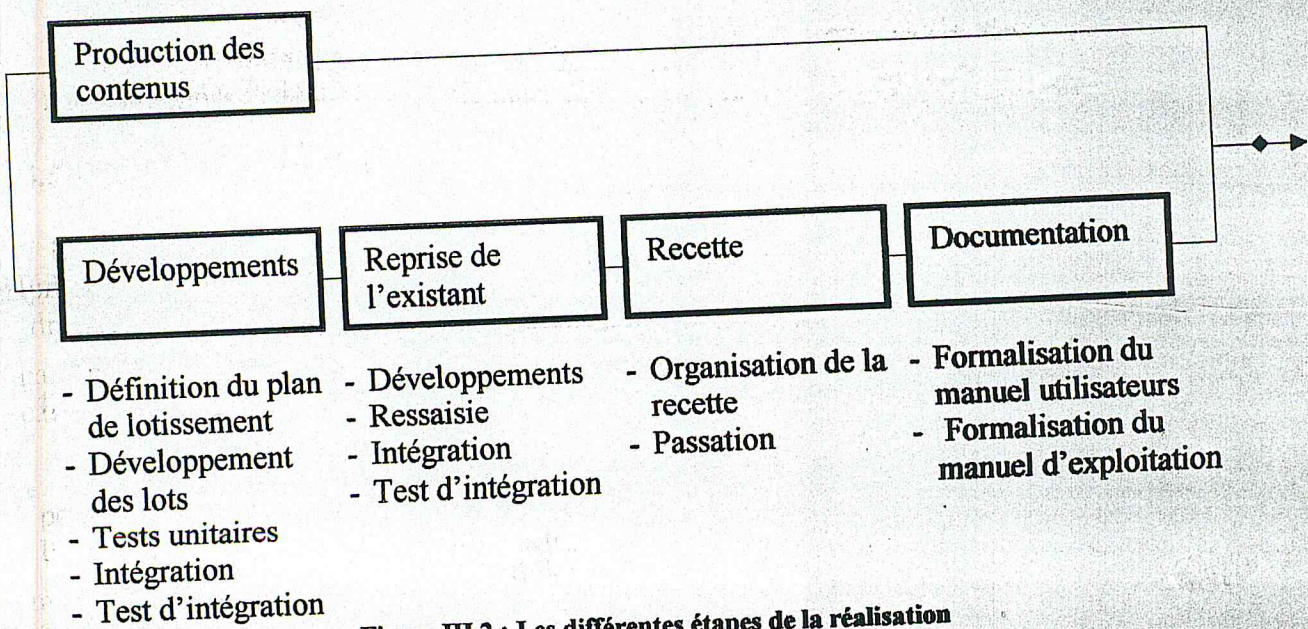


Figure III.2 : Les différentes étapes de la réalisation

La phase de réalisation consiste soit à mettre en œuvre un logiciel soit à réaliser des développements spécifiques. Normalement, l'étude de faisabilité a tranché entre ces solutions. Dans le premier cas, les efforts sont limités puisqu'il ne s'agit, en théorie, que de paramétrer les outils. Dans le second cas, les efforts sont importants et demandent un bon suivi du projet.

Dans tout les cas, cinq étapes sont incontournables : la production des contenus, le développement (ou le paramétrage), la reprise de l'existant, la recette et la documentation.[43]

## 5.1. Production des contenus :

### 5.1.1. Contenus éditoriaux :

La production des textes d'un site demande une grande rigueur et beaucoup de professionnalisme car écrire pour le web est difficile : pour que la lecture soit agréable, le calibrage doit être court, la titrairie incitative et le style très dynamique. Autant de contraintes qui ne facilitent pas la rédaction !

La pratique courante est donc de dissocier le contenu rédactionnel appelé à durer de celui plus ponctuel tel que les actualités ou les promotions. Cela permet de répartir au mieux le travail entre l'équipe interne et les prestataires, en fonction des compétences de chacun.

### 5.1.2. Contenus graphiques :

Les contenus graphiques d'un site jouent deux rôles, d'une part, ils assurent une navigation agréable et intuitive et, d'autre part, ils illustrent le propos. Leur production peut demander, dans certains cas, un effort très important, notamment lors de la création d'un catalogue. Le niveau de qualité exigé est alors très élevé, car sur Internet, les photos participent à la vente autant que le prix ou les informations produites.

## 5.2. Développements :

Avec la complexité croissante des projets web, une plus grande préparation de l'étape de développement est aujourd'hui nécessaire.

Le but est d'éviter à tout prix « l'effet tunnel » qui conduit tant de projets à ne pas respecter la charge et les délais impartis. Le chef de projet technique prend ici toute sa dimension.

Cette étape est composée de trois tâches principales : la définition du plan de lotissement, la réalisation des développements, l'exécution des tests.

### 5.2.1. Définir le plan de lotissement :

L'une des principales difficultés de la phase de réalisation est de respecter le planning de développement car de nombreux problèmes non identifiés peuvent apparaître à ce moment-là. Le lotissement permet de limiter ce risque en supprimant les goulots d'étranglement et en rendant autonomes les intervenants.

Cette étape consiste à diviser les développements en lots qui pourront être développés indépendamment par plusieurs développeurs ou plusieurs équipes de développeurs. Le but

poursuivi est de limiter les risques et de favoriser la production d'un code de qualité et documenté.

Le lotissement constitue un excellent outil de gestion de projet. Il donne de la visibilité à l'avancement réel des développements et permet de faire un point hebdomadaire constructif.[46]

#### **5.2.2. Développer les lots :**

Chaque lot est développé en fonction des spécifications détaillées et du plan de lotissement. La durée de cette tâche est très variable selon la nature du projet, du lot, et la solution retenue.

#### **5.2.3. Réaliser les testes unitaires :**

Cette tâche consiste à tester les développements de chaque lot pour s'assurer de leur conformité avec les spécifications détaillées. Elle est, en général, réalisée par le chef de projet technique.

#### **5.2.4. Opérer l'intégration :**

Il s'agit de regrouper les différents lots développés en un ensemble homogène et fonctionnel. Si l'on considère que la gestion du mailing list et la gestion des utilisateurs sont des lots différents, alors l'intégration consiste à regrouper des différents programmes qui les composent en un ensemble unique et à s'assurer que l'ensemble fonctionne correctement.

L'intégration est réalisée par l'équipe en charge des développements.

#### **5.2.5. Conduire les testes d'intégration :**

Ces testes sont fréquemment « oubliés » de manière à rattraper le retard pris lors des phases amont du projet. Ils constituent pourtant une étape très importante censée garantir la complétude et la qualité des développements.

Cette tâche est, en général, réalisée par le chef de projet technique.

### **5.3. Reprise de l'existant :**

Les projets web étant de plus en plus des deuxièmes ou troisièmes versions de site, la reprise de l'existant est une étape désormais incontournable. A tel point qu'elle détermine parfois la faisabilité de certains aspects des projets et peut même influencer les choix de solution lors de l'étude de faisabilité.

### **5.4. Recette de l'application :**

Contrairement à beaucoup d'idées reçues, la recette devrait se préparer en même temps que le cahier des charges car elle en est complémentaire. En effet, lister les scénarios de test et les erreurs potentielles permet d'affiner le cahier des charges en mettant l'accent sur les aspects délicats du projet. E, outre, travailler sur la recette dès le cahier des charges permet d'intégrer en suite plus facilement le ou les prestataire(s) et dédramatise cette étape mal aimée.

Précisons que l'objectif d'une recette n'est pas d'identifier le plus d'anomalies possible (ce n'est qu'un moyen) mais bien de trouver des solutions avec le prestataire dans une démarche constructive et bénéfique au projet.[43]

### **5.5. Documentations :**

La documentation est souvent considérée, côté client comme côté prestataire, comme secondaire : soit la charge est déjà dépassé.

C'est pourtant sur elle que s'appuient les supports de formation, c'est aussi vers elle que les utilisateurs se tourneront au quotidien.

#### **5.5.1. Formaliser le manuel d'exploitation :**

Cette tâche consiste à rédiger le manuel d'exploitation de l'application. Il comprend des informations sur :

- La mise en service ;
- L'exploitation régulière ;
- L'interruption ou l'arrêt d'exploitation ;
- Les consignes liées à la sécurité.

Il est en général rédigé par le chef de projet technique.[17]

#### **5.5.2. Formaliser le manuel utilisateurs :**

Cette tâche consiste à rédiger le manuel utilisateurs. Celui-ci comprend en général une description de chaque fonctionnalité (explication écrite, une ou plusieurs capture d'écran...) ainsi que des processus types (ajout d'une fiche produit, validation d'un achat...).

Elle est réalisée, selon la nature du projet, par le chef de projet technique ou fonctionnel.[17]

# Chapitre IV.

## Etude de l'existant

### **Présentation d'organisme d'accueil :**

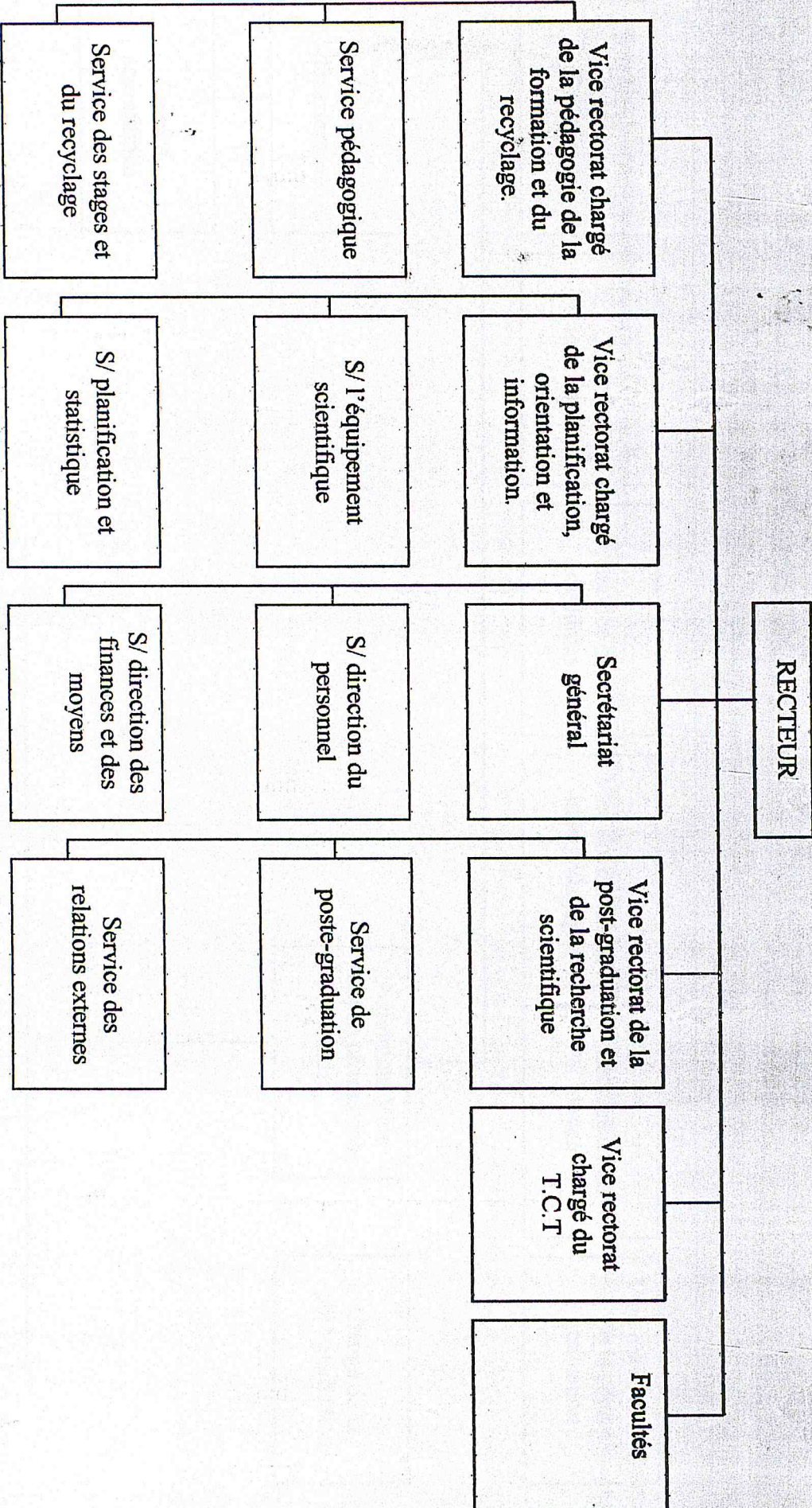
L'université SAAD DAHLAB de blida a ouvert ses portes en 1981, dirigée par le RECTORAT qui est sous l'autorité du recteur, d'après le décret 89-137 du 1<sup>er</sup> janvier 1989 portant la création de l'université de blida.

D'après le décret exécutif n° 98-253 de 24 Rabie Ethani 1419 correspondant au 17 août 1998, l'université de blida regroupe les facultés suivantes :

- Faculté des sciences.
- Faculté des sciences de l'ingénieur.
- Faculté des sciences médicales.
- Faculté de droit.
- Faculté des sciences économiques et des sciences de gestion.
- Faculté des lettres et des sciences sociales.
- Faculté des sciences agronomiques et vétérinaires.



Organigramme de l'université :



**1. Flux d'information :**

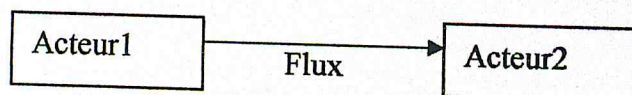
Le graphe des flux permet de mettre en évidence les informations échangées entre les différents acteurs du domaine étudié ainsi qu'avec leur environnement.

**Concepts utilisés :**

Acteur : poste capable d'échangé de l'information avec un autre acteur.

Flux : l'information échangée entre deux acteurs

**Formalisme**

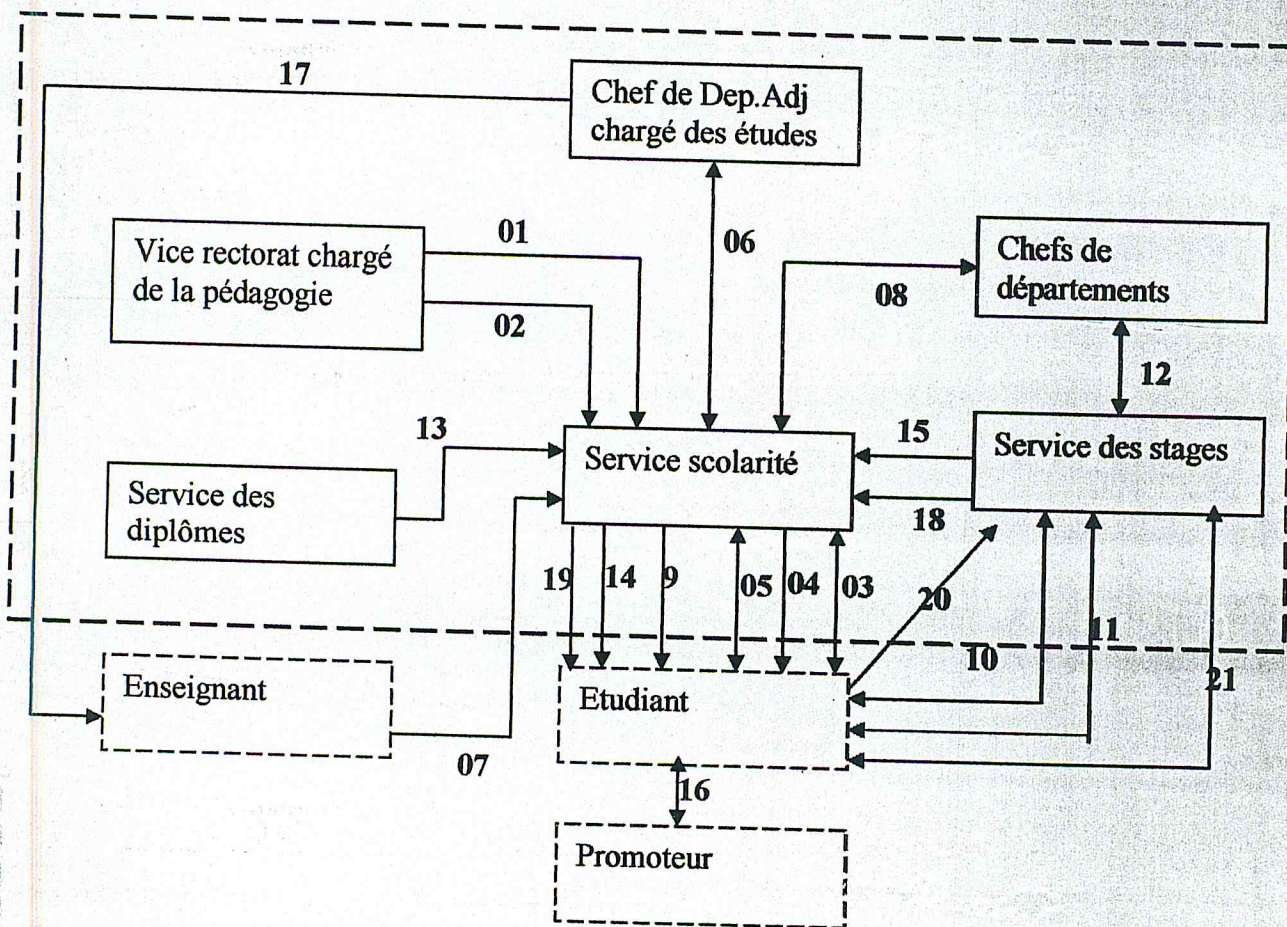


----- : Champ d'étude

-----  
Acteur : Acteur externe

Acteur : Acteur interne

**Flux d'informations :**



**Description du diagramme de flux :**

N° opération	Désignation de l'opération
01	Liste des étudiants orientés.
02	Certificats de scolarité.
03	Dossier d'inscription
04	Certificats de scolarité.
05	Fiche de transfert
06	Avis de chef département sur le transfert
07	Procès verbal modulaire.
08	Visa des attestations des succès et des relevés des notes.
09	Attestations des succès et des relevés des notes.
10	Appréciation du sujet de fin d'étude.
11	Dossier de diplôme.
12	Visa Dossier de diplôme.
13	Diplôme visé.
14	Diplôme.
15	Fiche d'évaluation du soutenance
16	Autorisation du soutenance visée
17	Invitation du soutenance
18	Fiche des réserves et corrections
19	Attestation du classement
20	Fiche projet fin d'étude
21	Autorisation du soutenance non visée

**2. Etude des postes de travail :**

Les postes de travail concernés directement par l'application à développer doivent être étudiés de manière détaillée. Il faut définir :

- Les responsabilités.
- Les tâches à accomplir
- Les documents manipulés.

**Poste 1**

Désignation : chef de département.

Structure de rattachement : Faculté.

Nombre du personnel : 01

**Tâche à accomplir**

- Exerce le pouvoir hiérarchique et disciplinaire sur l'ensemble du personnel et des étudiants du département.
- Gère et contrôle l'ensemble des activités administratives et pédagogiques du département.
- Contrôle et vise les documents.
- Assure le bon déroulement des examens.

**List des documents**

- attestation de succès.
- Fiche d'évaluation.
- Relevé des notes.
- Procès verbal annuel.
- Procès verbal de soutenance.
- Procès verbal de validation du diplôme.
- Diplôme provisoire.

**Poste 2**

Désignation : adjoint chef département chargé des études.

Structure a la quelle il est rattachée : Département.

Nombre du personnel : 01

**Tâche à accomplir**

- Elaboration des emplois du temps.
- Programmation des examens.
- Signature des documents pédagogique.
- Recrutements des enseignant.
- Affectation des modules aux enseignant.
- Résolution des problèmes pédagogiques soulevés par des enseignants et les étudiants au cours de l'année

**List des documents**

- Fiche de transfert.
- Attestation de non sanction disciplinaire.
- Attestation de classement.
- Invitation de soutenance.

**Poste 3**

Désignation : chef de service de scolarité.

Structure a la quelle il est rattachée : Adjoint chef département chargé des études.

Nombre du personnel : 01

**Tâche à accomplir**

- Assurer l'inscription et la réinscription et veiller aux transferts éventuels des dossiers des étudiants.
- Assurer le suivi des étudiants au cours de l'année d'étude (EMD, NOTE). CPM, CPC
- Etablissements des emplois du temps, planning des examens et les notes des EMD, synthèses et rattrapages.
- Contrôle des absences.
- Assurer les saisies des notes (relevé des notes, procès verbal modulaire).
- Assure le bon déroulement des examens.

**Liste des documents**

- Listes des étudiants orientés.
- Certificat de scolarités.
- Attestation de succès.
- Relevé des notes.
- Fiche de transfert.
- Procès verbal modulaire.
- Procès verbal annuel.
- Attestation de non sanction disciplinaire.
- Attestation de classement des étudiants.

**Poste 4**

Désignation : responsable service des stage.

Structure a la quelle il est rattachée : service de scolarité.

Nombre du personnel : 02

**Tâche à accomplir**

- Proposition des sujets de projet de fin d'étude par les enseignants aux étudiants.
- Appréciation du sujet de projet de fin d'étude proposé par l'étudiant.
- Validation du sujet de projet de fin d'étude par les membres de la commission du P.F.E (l'accord définitif pour entamer le sujet).
- Etablissements des documents de soutenance.
- Préparation du dossier de diplôme.

**Les des documents**

- Fiche de renseignements.
- Fiche d'appréciation du P.F.E.
- Autorisation de soutenance de P.F.E.
- Fiche d'évaluation réserves et correction.
- Procès de validation du diplôme.
- Attestation de soutenance.
- Diplôme provisoire.

### 3. Etudes des documents :

L'étude d'un document permet de spécifier sa description et son volume ainsi que les caractéristiques des données qu'il support.

Avant d'étudier séparément chaque document il est intéressant de faire l'inventaire dans les documents relatifs au service concerné.

#### Légende : Types des données

Lettre utilisée	Type
A	Alphanumérique
N	Numérique
D	Date
T	Heure
F	Réel
BL	Booléen
MN	Argent

#### La liste des documents :

Code	Désignations
ATT.S	Attestation de succès.
C.S	Certificat de scolarité.
L.E	Liste des étudiants.
R.N	Relevé des notes.
<i>Dans Annexe D.</i>	
ATT.CLA	Attestation de classement.
A.S	Autorisation des soutenances.
F.A.P.F.E	Fiche d'appréciation de projet fin d'étude.
F.E	Fiche d'évaluation du soutenance.
F.P.F.E	Fiche de projet fin d'étude.
F.R	Fiche de renseignement.
F.T	Fiche de transfert.
INV	Invitation de soutenance.
P.S	Planning des soutenances.
P.V.A	Procès verbal annuel.
P.V.M	Procès verbal modulaire.
P.V.V.D	Procès verbal de validation des diplômes.
R.C	Réserves et corrections.

## Fiche d'analyse du document

## A. Attestation de succès

## Caractéristique du document :

- Code : ATT.S
- Désignation : Attestation de succès
- Nature : Externe
- Utilité: Apprécier l'étudiant après ses résultats
- Origine : Service de scolarité.
- Destination : Etudiant
- Langue : Française.

## Description du document :

Code	Désignation	Type	Longueur
DESC DEP	Libelle de département	A	35
NOM ETD	Nom d'étudiant	A	35
PRN ETD	Prénom d'étudiant	A	35
DATE NAIS ETD	Date de naissance d'étudiant	D	8
LIEU NAIS ETD	Lieu de naissance d'étudiant	A	25
ANN UNIV	Année universitaire	N	8
MAT ETD	Matricule d'étudiant	N	10
LIB FIL	Filière	A	25
CYL	Cycle	A	10
DATE_ETAB_DOC	Date d'établissement du document	D	8
OBS	Observation	A	250

## B. Certificat de scolarité

## Caractéristique du document :

- Code : C.S
- Désignation : Certificat de scolarité
- Nature : Externe.
- Utilité: Evaluation du niveau d'étudiant
- Origine : Service de scolarité
- Destination : Etudiant
- Langue : Française.

## Description du document :

Code	Désignation	Type	Longueur
NOM ETD	Nom d'étudiant	A	35
PRN ETD	Prénom d'étudiant	A	35
DATE NAIS ETD	Date de naissance d'étudiant	D	8
LIEU NAIS ETD	Lieu de naissance d'étudiant	A	025
AN ETUD	Année d'étude	N	1

Code	Désignation	Type	Longueur
ANN UNIV	Année universitaire	N	8
LIB FIL	Filière	A	25
CYL	Cycle	A	10
MAT ETD	Matricule d'étudiant	N	10
NAT	Nationalité	A	30
DATE ETAB DOC	Date d'établissement du document	D	8
OBS	Observation	A	250

### C. Liste des étudiants.

#### Caractéristique du document :

- Code : L.E
- Désignation : Liste des étudiants.
- Nature : Externe
- Utilité: la classification des étudiants par groupes
- Origine : Service de scolarité.
- Destination : Affichage.
- Langue : Française.

#### Description du document :

Code	Désignation	Type	Longueur
DESC DEP	Libelle de département	A	35
CYL	Cycle	A	10
ANN UNIV	Année universitaire	N	9
LIB SEC	Section	A	1
LIB GRP	Groupe	N	2
NUM ETD	Numéro d'ordre	N	2
MAT ETD	Matricule d'étudiant	N	10
NOM ETD	Nom de l'étudiant	A	35
PRN ETD	Prénom de l'étudiant	A	35

### D. Relevé des notes.

#### Caractéristique du document :

- Code : R.N
- Désignation : Relevé des notes.
- Nature : Externe.
- Utilité: Evaluation des connaissances.
- Origine : Service de scolarité.
- Destination : Etudiant
- Langue : Française.



**Description du document :**

Code	Désignation	Type	Longueur
DESC DEP	Libelle de département	A	35
NOM ETD	Nom de l'étudiant	A	35
PRN ETD	Prénom de l'étudiant	A	35
DATE NAIS ETD	Date de naissance de l'étudiant	D	8
LIEU NAIS ETD	Lieu de naissance de l'étudiant	A	025
ANN UNIV	Année universitaire	N	8
MAT ETD	Matricule d'étudiant	N	10
LIB FIL	Filière	A	25
CYL	Cycle	A	10
LIB MOD	Intitule du module	A	35
COEF MOD	Coefficient	N	1
NOTE	Notes	F	4
MOY GEN	Moyenne générale	F	4
SESS	Session	A	10
DATE ETAB DOC	Date d'établissement de document	D	8
OBS	Observation	A	10

**4. Etude des procédures :**

Une procédure présente un enchaînement d'opérations dans le temps, exécutées pour atteindre une finalité dans le système.

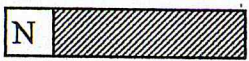

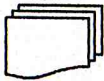
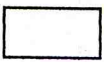

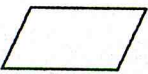

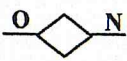


L'étude des procédures a pour but de suivre la circulation de l'information entre les différents postes de travail afin de dégager les goulots d'étranglement s'il y'en a.

Dans cette phase nous allons représenter les procédures par les diagrammes de circulation des documents ou diagramme et de traitement des informations (DCD, DCTI).

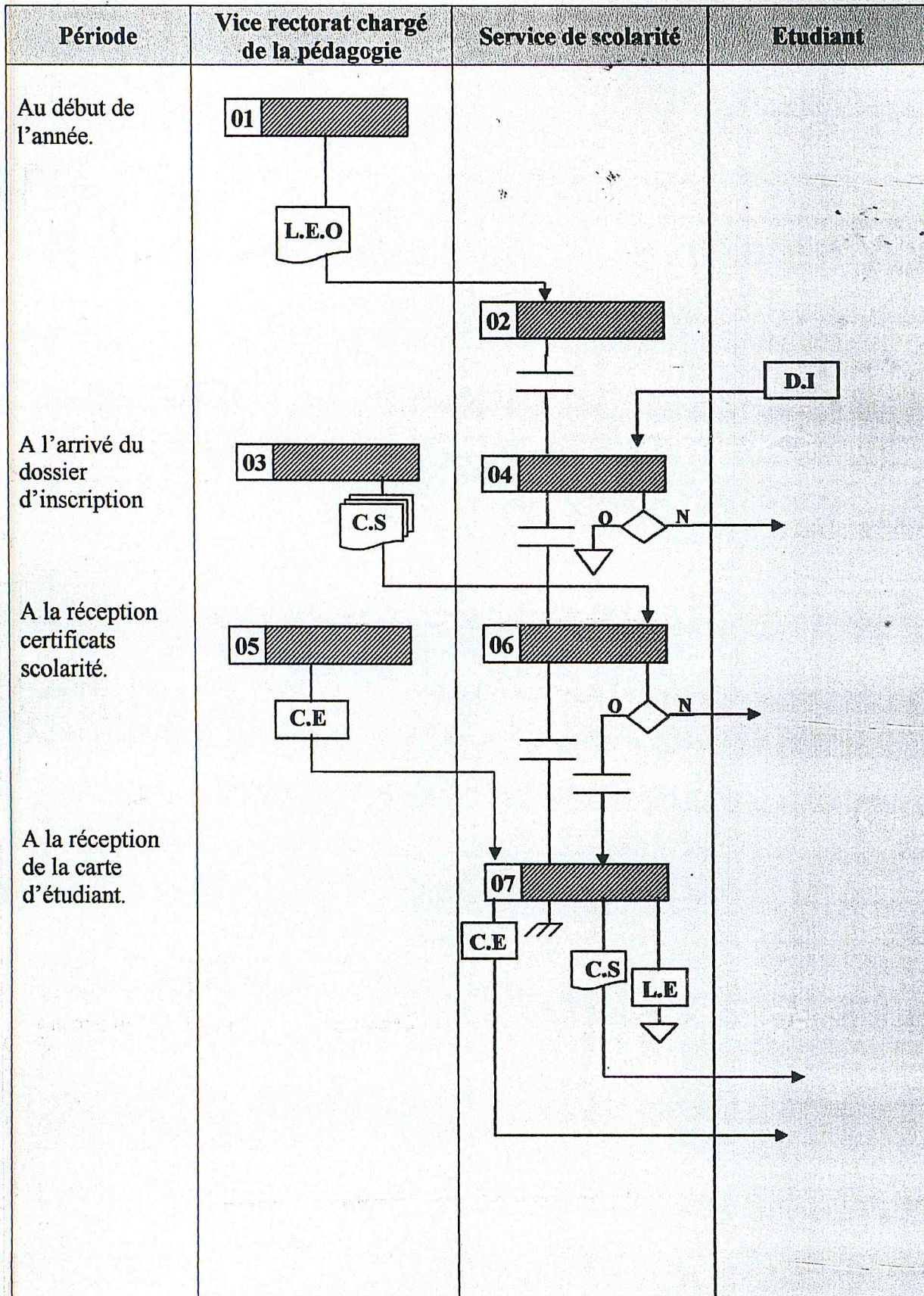
**Les procédures étudiées dans notre système sont :**

- 1- Procédure inscription.
- Dans Annexe D.*
- 2- Procédure réinscription.
- 3- Procédure transfert.
- 4- Procédure absentéisme.
- 5- Procédure délibération.

Légende :

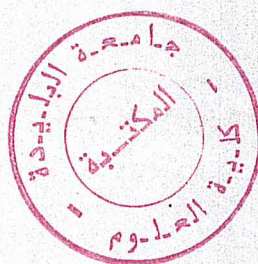
Symbole	Désignation
	Opération numéro N.
	Documents en un seul exemplaire.
	Documents en 'n' exemplaire.
	Dossier.
	Archivage.
	Fichier, registre
	Transmission des documents
	Contrôle, prise de décision.
	Mise en attente.
	Classement.

Procédure N°1 'Inscription' :



Descriptif N°1 :

N° d'opération	Désignation
01	Orientation des étudiants.
02	Réception des listes des étudiants orientés (L.E.O). La mise en attente des listes des étudiants orientés (L.E.O)..
03	Etablissement des (C.S)
04	Réception des dossiers d'inscription (D.I) des étudiants par le service de scolarité. Le contrôle du dossier d'inscription (D.I) Si accepté : D.I classé. Sinon : D.I remis à l'étudiant. La mise en attente des listes des étudiants orientés (L.E.O).
05	Etablissement des (C.E).
06	Réception des certificats de scolarité (C.S) en 04 exemplaires et carte étudiant. Vérification des (L.E.O) avec les (C.S) Si accepté : certificats de scolarités mise en attente. Sinon : étudiant informer
07	Réception de la carte de l'étudiant (C.E). Envoi de (C.E). Envois de (C.S). Archivage (L.E.O). Etablissement et affichage des listes des étudiants par section et groupe (L.E)



## 5. Les règles de gestion:

- **Règle 1 (article 11) :**

L'accès à l'Université se fait avec le titre du baccalauréat de l'Enseignement Secondaire Algérien ou d'un titre équivalent étranger.

- **Règle 2 (article 12) :**

La première inscription à l'Université se fait au niveau du Vice-Rectorat chargé de la pédagogie et doit être confirmée au niveau de département.

- **Règle 3 (article 13) :**

Seuls les étudiants régulièrement inscrits et ayant en leur possession les documents de scolarité (carte d'étudiant et certificat de scolarité) peuvent suivre les enseignements. Le suivi des cours en qualité d'auditeur libre peut être autorisé sur la base de l'étude d'un dossier par l'administration centrale (Rectorat chargé de la pédagogie) et celle de départements.

- **Règle 4 (article 14) :**

Les enseignements sont organisés en cours magistraux, travaux dirigés et travaux pratiques auxquels peuvent s'ajouter les sorties d'études et les stages.

- **Règle 5 (article 15) :**

L'affectation en groupe de TD et TP est de la seule compétence de l'administration pédagogique.

- **Règle 6 (article 16) :**

Les emplois du temps élaborés par l'administration pédagogique sont la base de la gestion et du contrôle des enseignements, leur respect est de rigueur.

- **Règle 7 (article 18) :**

L'organisation des enseignements est modulaires.

- **Règle 8 (article 19) :**

L'évaluation pédagogique est basée sur les épreuves de moyenne durée (EMD), l'examen en TD et/ou TP, l'évaluation du travail personnel (exposés, rapports de sorties et de stages).

- **Règle 9 (article 20) :**

L'étudiant doit passer tous les examens de l'ensemble des modules de son année pédagogique.

- **Règle 10 (article 21) :**

Une absence justifiée à une épreuve de moyenne durée (EMD) entraîne systématiquement l'étudiant à passer l'épreuve de synthèse. La note obtenue lors de la synthèse remplace la note de l'EMD où l'étudiant s'est absenté.

- **Règle 11 (article 22) :**

En cas d'absence même justifiée à deux EMD du même module, l'étudiant est exclu du module.

**• Règle 12 (article 25) :**

L'affectation des sujets de projets de fin d'études dans les filières est régie par le règlement des PFE en vigueur. L'évaluation des PFE est régie par le même règlement.

**• Règle 13 (article 26) :**

La programmation de la période des EMD est effectuée en conseil pédagogique de l'Université regroupant l'ensemble des responsables des études de toutes les facultés.

**• Règle 14 (article 27) :**

Le planning des épreuves de moyennes durées est effectué par l'administration pédagogique. Enseignants et étudiants sont tenus de respecter les dates et les lieux du déroulement des examens.

**• Règle 16 (article 30) :**

Lors des examens, les surveillants doivent veiller à établir un PV de surveillance. Ce dernier doit être remis à l'administration accompagné d'une liste émergée des étudiants présents et d'une copie du sujet d'examen.

**• Règle 17 (article 33) :**

Toute communication écrite ou verbale durant l'examen expose l'étudiant à sa traduction devant un conseil de discipline ; à cet effet, le surveillant devra rédiger un rapport circonstancié et le remettre au responsable du module qui saisira le conseil de discipline.

**• Règle 18 (article 47) :**

Les étudiants non admis lors de la session de rattrapage peuvent être autorisés à doubler après étude du dossier pédagogique. Dans ce cas, l'étudiant n'est autorisé à s'inscrire qu'aux moules où la moyenne obtenue est inférieure à 10 et conserve le bénéfice des modules acquis.

**• Règle 19 (article 48) :**

Une délibération annuelle est organisée après les examens de synthèse et de rattrapage.

**• Règle 20 (article 49) :**

L'étudiant peut être autorisé à se réinscrire à un module acquis. Dans ce cas, il perd le bénéfice de ce module.

**• Règle 21 (article 50) :**

Le conseil scientifique doit siéger après chaque délibération pour valider les résultats de délibération.

**• Règle 22 (article 51) :**

Lors des déroulements des examens, l'administration communique par voix d'affichage la liste d'affectation des étudiants dans les locaux pédagogiques et celle des enseignants. Le respect des planning est exigé.

L'enseignant surveillant doit :

- Vérifier l'identité des étudiants,
- Etablir un PV de surveillance qui doit comporter les éléments suivants :
  1. date, module, numéro de salle, etc...,
  2. liste des enseignants de surveillance avec leur émargement,
  3. liste des étudiants avec émargement,
  4. le nombre de copie remises,

5. incidents éventuels pendant la surveillance,
6. sujet de l'examen.

• **Règle 23 (article 63) :**

Le jury de délibération est constitué de tous les responsables de modules, il est présidé par un des enseignants le plus haut gradé.

Les membres de jury sont convoqués par écrit par l'administration des études. La présence de enseignants est obligatoire.

• **Règle 24 (article 66) :**

Le jury de délibération a pour tâches :

- Vérifier les moyennes semestrielle et annuelles,
- Valider les résultats des étudiants,
- Se prononcer sur l'admission en année supérieure des étudiants,
- Etablir le PV de délibération qui doit faire ressortir :
  1. les enseignants présents,
  2. le nombre d'admis,
  3. le nombre d'ajournés,
  4. le nombre d'abandons.

• **Règle 25 (article 67) :**

Après chaque délibération, le PV de délibération doit être rédigé. Il doit faire ressortir :

- Le relevé de note qui doit être :
  1. net et sans rature,
  2. signé par chacun des membres du jury, le nom étant lisible,
  3. daté,
  4. clair quant aux résultats des étudiants admis ou ajournés.
- La note zéro doit être justifiée par un rapport accompagnant le procès verbal comportant les éléments suivants :
  1. Pourcentage des admis par rapport aux inscrits,
  2. Pourcentage des ajournés par rapport aux inscrits,
  3. Pourcentage des abandons par rapport aux inscrits,
  4. Pourcentage des exclusions par rapport aux inscrits.

• **Règle 26 (article 71) :**

Le rachat est une faveur du jury. Il ne constitue nullement un droit dont l'étudiant pourrait se prévaloir. La procédure de rachat est différente d'un jury à un autre.

• **Règle 27 (article 72) :**

Durant son cycle de formation, l'étudiant ne peut refaire la même année plus de deux fois (triplement).

### Les règles de calcul:

• **Règle 1 (article 23) :**

Le nombre d'EMD est de un (01) pour les modules semestriels et de deux (02) pour les modules annuels.

**Règle 2 (article 24) :**

Selon les spécificités des filières, les notes des sorties, stages ou rapports individuels sont comptabilisées comme une (01) note de TD et/ou TP.

**• Règle 3 (article 37) :**

La moyenne modulaire  $M_1$  de la session normale est calculée à partir de l'ensemble des EMD, des TD et des TP (Les notes des sorties pédagogique, rapport et mémoires sont considérées comme des notes e TP).

$$M_1(1^{\text{er}} \text{ session}) = \frac{(\text{somme des EMD} + \text{Notes des TD} + \text{Moyenne des TP})}{(\text{Nombre des EMD} + 2)}$$

**• Règle 4 (article 38) :**

La moyenne générale est calculée à partir des moyennes de tous les modules pondérés par leurs coefficients respectifs, sauf réglementation spécifique.

**• Règle 5 (article 39) :**

Les étudiants non admis en session normale sont autorisés à passer l'examen de synthèse pour les modules non acquis et obligatoirement pour les notes inférieures à 05/20.

**• Règle 6 (article 40) :**

À l'issue des épreuves de synthèses, la moyenne modulaire  $M_2$  est calculée comme suite :

$$M_2 = \frac{(\text{Synthèse} * \text{Nombre des EMD} + \text{Notes des TD} + \text{Moyenne des TP})}{(\text{Nombre des EMD} + 2)}$$

**• Règle 7 (article 41) :**

La moyenne de la session de synthèse  $M$  est le SUP entre la moyenne de la session normale et celle calculée à partir de la synthèse pondérée par le nombre des épreuves et des notes de TD et TP.

$$\text{La moyenne modulaire } M = \text{SUP} (M_1, M_2)$$

**• Règle 8 (article 42) :**

La moyenne générale annuelle  $M_G$  est calculée à partir des moyennes modulaires pondérées par leurs coefficients respectifs.

$$M_G = \frac{(\text{Somme des moyennes modulaires } M \text{ pondérée par leurs coefficients respectifs})}{(\text{Sommes des coefficients})}$$

**• Règle 9 (article 43) :**

Les étudiants non admis au cours de la session synthèse sont autorisés à passer l'examen de rattrapage pour les modules non acquis et obligatoirement pour les notes inférieures à 05/20, , sauf réglementation spécifique



• **Règle 10 (article 44) :**

La moyenne générale compensée de la session de rattrapage est calculée à partir des notes obtenues à la session précédente pour les modules acquis et de la meilleure des notes obtenues entre les trois sessions.

• **Règle 11 (article 45) :**

La moyenne modulaire de la session de rattrapage est celle obtenue ans l'examen de rattrapage.

La moyenne générale de la session de rattrapage est calculée comme suite :

$$M_R = \text{SUP des moyennes modulaires pondérées par leurs coefficients respectifs}$$

• **Règle 12 (article 46) :**

L'étudiant est admis en année supérieure lorsque la moyenne générale est supérieure ou égale à 10/20 et aucune moyenne modulaire inférieure à 05/20.

• **Règle 13 (article 116) :**

La charge horaire hebdomadaire de l'enseignant, quel que soit son grade est :

- L'équivalence de 06h de cours : elle peut se répartir en cours, TD et TP selon le rapport : 01h de cours correspond à 01h30 en TD ou 02h de TP.

• **Règle 14 (article 117) :**

La charge horaire hebdomadaire d'un professeur est :

1<sup>er</sup> cas : Existence de la post-graduation :

deux cours non répétitifs (02 modules différents) et un cours en post-graduation

2<sup>eme</sup> cas : Absence de la post-graduation

trois cours différents (trois modules différents)

• **Règle 15 (article 118) :**

La charge horaire hebdomadaire d'un maître de conférence est :

1<sup>er</sup> cas : Existence de la post-graduation :

deux cours non répétitifs (02 modules différents) et un cours en post-graduation

Si l'administration n'est pas en mesure d'affecter à l'enseignant deux cours non répétitifs, la charge comportera : un cours en graduation, un cours en post-graduation et un complément en travaux dirigés (TD).

2<sup>eme</sup> cas : Absence de la post-graduation

deux cours différents (deux modules différents) et un complément en travaux dirigés (TD).

• **Règle 16 (article 119) :**

La charge horaire hebdomadaire du chargé de cours est :

1<sup>er</sup> cas : Existence de la post-graduation :

deux cours non répétitifs (02 modules différents) et un complément en cours de post-graduation ou en travaux dirigés (TD).

2<sup>eme</sup> cas : Absence de la post-graduation

deux cours différents (deux modules différents) et un complément en travaux dirigés (TD).

**• Règle 17 (article 120) :**

La charge horaire hebdomadaire du maître assistant est :

- L'équivalence de douze 12h équivalente de TP qui peuvent se répartir en cours, TD et TP.

**• Règle 18 (article 121) :**

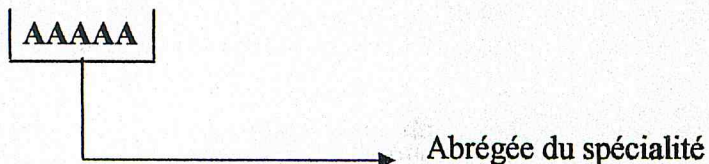
La charge horaire hebdomadaire d'un assistant est :

- L'équivalence de douze 12h équivalente de TP qui peuvent se répartir en TD et TP.

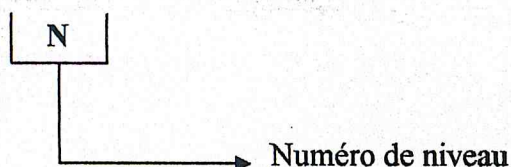
**6. Codification existante :**

La codification est un système de symbole permettant de transformer des informations littéraires en une présentation abrégée, elle peut apparaître comme une combinaison alphanumérique.

Dans cette présentation, on cherche non seulement la simplicité, la facilité et la concision de la codification, mais aussi une souplesse de codification ou de reconnaissance de l'attribut codifié.

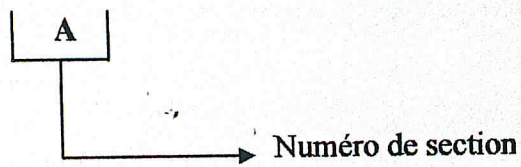
**Code spécialité**

Exemple :CMARK :Marketing.

**Code niveau**

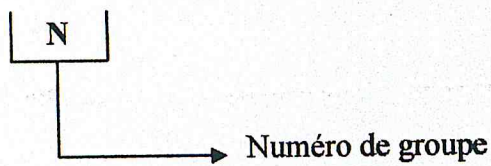
Exemple :2 :2<sup>eme</sup> année.

**Code section**



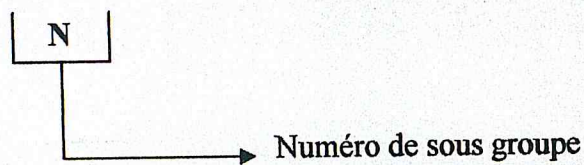
Exemple :B :Section B.

**Code groupe**



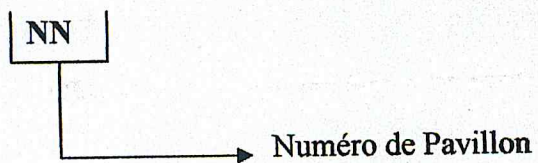
Exemple :3 :Groupe :3.

**Code sous groupe**



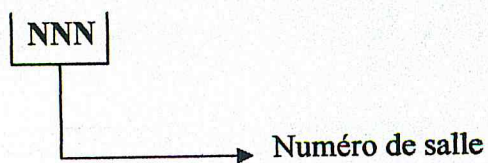
Exemple :2 :Sous-groupe 2.

**Code Pavillon**

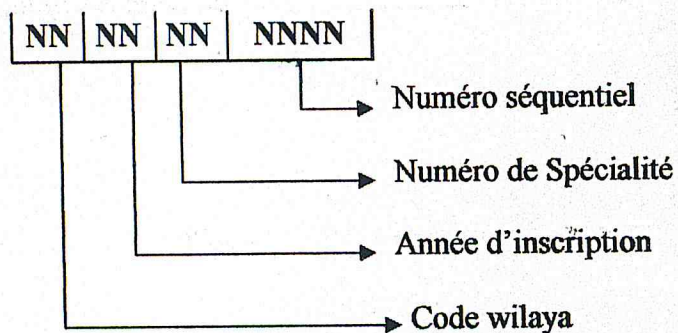


Exemple :23 :Pavillon 23.

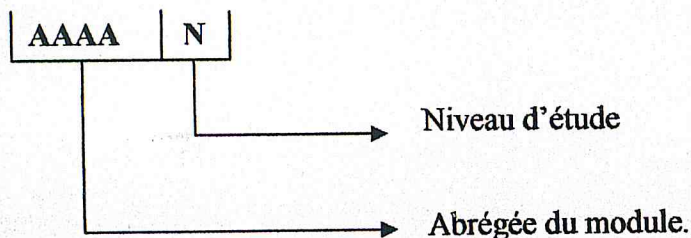
**Code local**



Exemple :226 :Salle 226.

**Matricule étudiant**

Exemple :0901110534 .

**Module**

Exemple : CCCO3 : COMPORTEMENT DU CONSOMMATEUR.

**7. Critiques et suggestions :**

Après une étude approfondie sur la situation actuelle de l'université, concernant le service de scolarité, nous avons pu recenser quelques anomalies qui ont un impact considérable sur le fonctionnement de ce dernier. Les critiques seront présentées en deux aspects:

- Aspect informationnel,
- Aspect organisationnel

**Les critiques :****A- Aspect informationnel :**

- Les documents sont établis manuellement, d'où un grand risqué d'erreur dans le traitement d'information.
- Mauvaise circulation de l'information entre les différents services.
- L'affichage de procès verbal modulaire ne se fait pas en temps voulu du au retard commis par les enseignants lors du remplissage du PV modulaire.
- L'information n'est pas disponible à tout moment.

**B- Aspect organisationnel :**

- Mauvaise codification de certains objets : l'étudiant change de matricule lors d'un transfert ce qui pose un grand problème dans le suivi de l'historique.
- Double attribution des matricules pour les étudiants.
- L'étudiant doit être présent pour consulter ces résultats.
- Il y a un grand problème dans le système d'affichage :
- Manque l'espace pour l'affichage.
- Affichage enlevé après un certain temps.
- Trop de consommation de papier pour l'affichage.
- L'enseignant doit être présent sur le lieu pour rendre les PV modulaires.
- Problème de chevauchement des salles, il existe par exemple une salle attribuée à deux groupes différents.
- L'impossibilité de suivi de la charge des enseignants.
- Les statistiques se font d'une façon manuelle qui réduit la certitude et qui nécessite un temps considérable.

**Les suggestions :****A-Aspect informationnel :**

- Automatiser les documents établis manuellement, ce qui va réduire le risque d'erreur et nous permettre un gain de temps.
- Centraliser l'information pour quelle puisse être disponible pour tout le monde et à tout moments, ce qui va nous permettre de résoudre le problème des statistiques
- Permettre l'accès à distance pour les utilisateurs ce qui va nous assurer un gain de temps dans l'établissement des procès verbaux modulaires par exemple.

**B-Aspect organisationnel :**

- Proposer des codifications robustes et souples dans la mesure du possible.
- Contrôler l'attribution des codes.
- Contrôler le chevauchement dans l'établissement des emplois du temps, des charges et l'affectation des salles.
- La possibilité de consulter à distance et à tout moment.

**8. Solution informatique :****A. Solution monoposte :**

Un micro-ordinateur et une imprimante doivent être dédiés à chaque utilisateur, chacun à une base de données pour faire ses traitements.

**Avantage :**

- Coût d'installation faible.
- Bonne protection des données.
- Simple à mettre en œuvre.

**Inconvénient :**

- L'information n'est pas disponible en temps réel.
- Très grande difficulté à mettre à jour la base de données globale.
- Pas de partage de l'information.

**9. Conclusion :**

L'étude de l'existant nous a permis de prendre connaissance du système existant de gestion de scolarité qui joue un rôle très important dans le cursus des étudiants, où on a dégagé des insuffisances auxquelles on a remédié afin d'avoir une meilleure gestion de scolarité.

Les résultats de cette analyse constituent une base pour aborder la phase suivante qui est l'élaboration du système d'information futur.

# Chapitre V.

## Etude conceptuelle

### 1. Introduction :

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse.

### 2. Dictionnaire de donnée :

Le dictionnaire de données est l'ensemble des informations utiles que nous voulons mémoriser dans la future base de données et cela après rajout de celles qui n'étaient pas prévues et la suppression de celles qui ne sont plus utilisées comme les informations calculables.

Code	Désignation	Type	Longueur
ADR_ENS	Adresse enseignant	A	100
ADR_ETD	Adresse étudiant	A	100
ADR_MBRE_EXT	Adresse membre externe	A	100
ADR_ORG	Adresse organisme	A	100
ADR_PART_ETD	Adresse parents étudiant	A	100
ANN	Année	N	4
ANN_UNIV	Année universitaire	A	9
CAP_LOC	Capacité locale	I	3
CHEF_PRJ	Chef projet	A	60
CODE_BAC	Code bac	A	10
CODE_COMM	Code communication	A	10
CODE_CYCLE	Code cycle	A	1
CODE_DEPT	Code département	A	3
CODE_DIPM	Code diplôme	A	4
CODE_EQUIPE	Code équipe	A	3
CODE_EXAM	Code examen	A	4
CODE_FAC	Code faculté	A	2
CODE_GRADE_PEDAG	Code grade pédagogique	A	5
CODE_GRADE_RECH	Code grade de recherche	A	5
CODE_GRP	Code groupe	A	8
CODE_LABO	Code laboratoire	A	3
CODE_LOC	Code local	A	5
CODE_MBRE_EXT	Code membre externe	A	10
CODE_MOD	Code module	A	5

Code	Désignation	Type	Longueur
CODE_NATIO	Code nationalité	N	3
CODE_NIV	Code niveau	A	5
CODE_NIV_INS_MERE	Code niveau instruction mère	A	3
CODE_NIV_INS_PERE	Code niveau instruction père	A	3
CODE_ORG	Code organisme	A	5
CODE_PARTIC	Code participation	A	5
CODE_PAV	Code pavillon	A	2
CODE_PRJ	Code projet	A	3
CODE_PUBL	Code publication	A	4
CODE_S_GRP	Code sous groupe	A	9
CODE_SANC	Code sanction	A	5
CODE_SEC	Code section	A	6
CODE_SEM	Code séminaire	A	10
CODE_SNS	Code séance	A	5
CODE_SOUT	Code soutenance	A	10
CODE_SPEC	Code spécialité	A	4
CODE_STG	Code stage	A	5
CODE_THM	Code thème	A	7
CODE_TYPE_MOD	Code type module	A	3
CODE_TYPE_ORG	Code type organisme	A	4
CODE_TYPE_SNS	Code type séance	A	3
CODE_TYPE_UNITE	Code type unité pédagogique	A	4
CODE_UNITE_PEDAG	Code unité pédagogique	A	3
CODE_WIL	Code wilaya	N	2
CODE_RESP	Code responsabilité	A	6
CODE_SERV	Code services centraux	A	6
COEF_EXAM	Coefficient	I	3
COEF_MOD	Coefficient module	I	3
CRD_MOD	Crédit modulaire	F	4
DATE	Date	D	8
DATE_DBT_SEM	Date début séminaire	D	8
DATE_DBT_STG	Date début stage	D	8
DATE_EDIT	Date édition	D	8
DATE_EXAM	Date examen	D	8
DATE_FIN_SEM	Date fin séminaire	D	8
DATE_FIN_STG	Date fin stage	D	8
DATE_INSC_ENS	Date d'inscription enseignant	D	8
DATE_NAIS_ENS	Date naissance enseignant	D	8
DATE_NAIS_ETD	Date naissance étudiant	D	8
DATE_OBT_DIPM	Date obtention diplôme	D	8
DATE_OBT_GRAD	Date obtention grade	D	8
DATE_SOUT	Date soutenance	D	8
DATE_RECLAM	Date réclamation	D	8
DUREE_SANC	Durée sanction	I	2
DUREE_SNS	Durée séance	T	
FAX_DEPT	Fax département	A	15
FAX_FAC	Fax faculté	A	15
FAX_ORG	Fax organisme	A	15
HEURE_DBT_EXAM	Heure début d'examen	T	
HEURE_DBT_MAT	Heure début matin	T	
HEURE_FIN_EXAM	Heure fin d'examen	T	
IGNORER_REC	Réclamation ignorer	BL	
JOUR	Jours	A	9
LIB_CYCLE	Libelle cycle	A	10
LIB_DEPT	Libelle département	A	25
LIB_DIPM	Libelle diplôme	A	35
LIB_EQUIPE	Libelle équipe	A	25
LIB_EXAM	Libelle examen	A	35
LIB_FAC	Libelle faculté	A	25

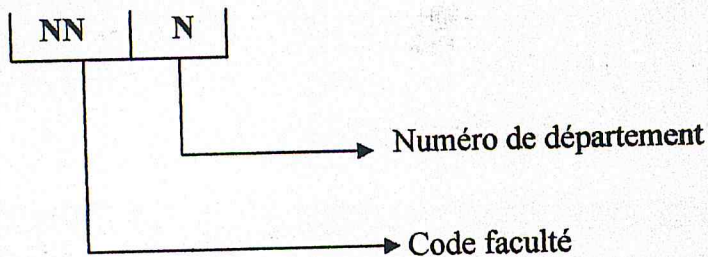


Code	Désignation	Type	Longueur
LIB_GRADE_RECH	Libelle grade recherche	A	25
LIB_GRADE_PEDAG	Libelle grade pédagogique	A	25
LIB_GRP	Libelle groupe	A	15
LIB_NIV_INS_MERE	Libelle niveau d'instruction mère	A	25
LIB_NIV_INS_PERE	Libelle niveau d'instruction père	A	25
LIB_LABO	Libelle laboratoire	A	35
LIB_LOC	Libelle local	A	25
LIB_MOD	Libelle module	A	25
LIB_NATIO	Libelle nationalité	A	35
LIB_NIV	Libelle niveau	A	25
LIB_ORG	Libelle organisme	A	35
LIB_PARTIC	Libelle participation	A	35
LIB_PAV	Libelle pavillon	A	25
LIB_PRJ	Libelle projet	A	35
LIB_RESP	Libelle responsabilité	A	35
LIB_S_GRP	Libelle sous groupe	A	25
LIB_SANC	Libelle sanction	A	35
LIB_SEC	Libelle section	A	25
LIB_SERV	Libelle services centraux	A	35
LIB_SPEC	Libelle spécialité	A	25
LIB_STG	Libelle stage	A	25
LIB_TYPE_MOD	Libelle type module	A	15
LIB_TYPE_ORG	Libelle type organisme	A	25
LIB_TYPE_UNITE	Libelle type unité pédagogique	A	35
LIB_UNITE_PEDAG	Libelle unité pédagogique	A	35
LIB_WIL	Libelle wilaya	A	35
LIEU_OBT_DIPM	Lieu d'obtention diplôme	A	60
MAT_ENS	Matricule enseignant	A	10
MAT_ETD	Matricule étudiant	A	10
MONT_INDEM_MENT	Montant d'indemnisation mensuelle	MN	
MONTION_ETD	Montions de bac étudiant	A	15
MOT_CLE_SOUT	Mot clé soutenance	A	250
MOY_ACQUIT	Moyenne d'un module acquit	F	4
MOYENNE	Moyenne	F	4
NBRE_F_S	Nombre de frères et sœur	I	2
NBRE_F_S_UNIV	Nombre de frères et sœur universitaire	I	2
NBRE_HEURE_COURS	Nombre d'heure cours	F	4
NBRE_HEURE_TD	Nombre d'heure TD	F	4
NBRE_HEURE_TP	Nombre d'heure TP	F	4
NBRE_SNS	Nombre de séance	I	2
NOM_ENS	Nom enseignant	A	25
NOM_ETD	Nom étudiant	A	25
NOM_MBRE_EXT	Nom membre externe	A	25
NOM_MERE_ETD	Nom de la mère d'étudiant	A	25
NOTE	Note obtenue dans un module	F	4
NUM_BAC_ETD	Numéro de bac étudiant	A	10
NUM_COMPTE	Numéro de compte	A	25
NUM_SALLE	Numéro de salle	I	3
NUM_PAV	Numéro de pavillon	I	2
OBS_ETD	Observation étudiant	A	250
PRN_ENS	Prénom enseignant	A	25
PRN_ETD	Prénom étudiant	A	25
PRN_MBRE_EXT	Prénom membre externe	A	25
PRN_MERE_ETD	Prénom mère d'étudiant	A	25
PRN_PERE_ETD	Prénom père d'étudiant	A	25
PROF_MERE_ETD	Profession mère étudiant	A	25
PROF_PERE_ETD	Profession père étudiant	A	25
QUALITE_ENCAD_INT	Qualité encadreur interne	A	25

Code	Désignation	Type	Longueur
QUALITE_JURY_EXT	Qualité jury externe	A	25
QUALITE_JURY_INT	Qualité jury interne	A	25
RECL_ETD	Réclamation d'étudiant	A	250
REPO_MIDI	Repos de midi	T	
REPO_SNS	Repos entre les séances	T	
RESUME	Résumé	A	250
RESUME_SOUT	Résumé soutenance	A	250
SERIE_BAC	Série de bac	A	25
SEXE_ETD	Sexe étudiant	A	2
SIT_FAM_ENS	Situation familiale enseignant	A	10
SUJET	Sujet	A	250
TEL_DEPT_INT	Téléphone département interne	A	15
TEL_DEPT_EXT	Téléphone département externe	A	15
TEL_ETD	Téléphone étudiant	A	15
TEL_FAC_EXT	Téléphone faculté externe	A	15
TEL_FAC_INT	Téléphone faculté interne	A	15
TEL_ORG	Téléphone organisme	A	15
TITRE_COMM	Titre communication	A	35
TITRE_PUBL	Titre publication	A	35
TITRE_SEM	Titre séminaire	A	35
TITRE_SOUT	Titre soutenance	A	35
TITRE_THM	Titre Thème	A	35
TRAIT_REC	Traitement d'une réclamation	A	250
TYPE_COMM	Type de communication	A	25
TYPE_COMPTE	Type de compte	A	25
TYPE_LOC	Type de local	A	25
TYPE_RECRUT	Type de recrutement	A	25
VALID_ETD	Validation d'inscription étudiant	BL	
VISITE_RECL	Réclamation visitée	BL	
VOL_HORR_MOD	Volume horaire d'un module	A	25

### 3. Codification proposée :

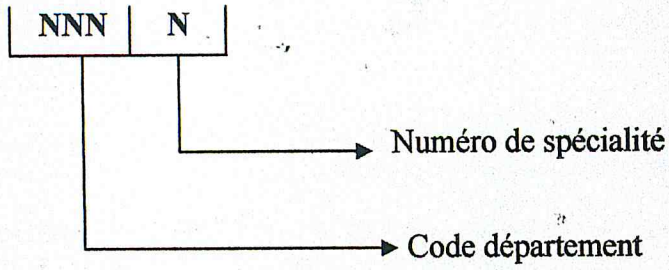
#### Code département



#### Exemple :

011 : Département informatique

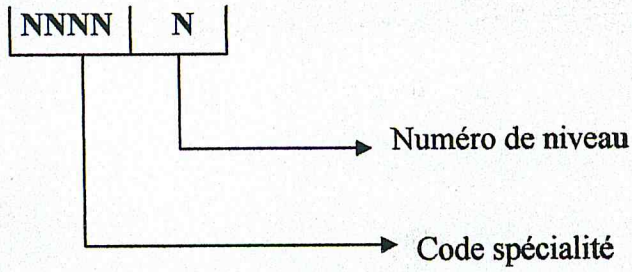
**Code spécialité**



**Exemple :**

0113 : Intelligence artificielle.

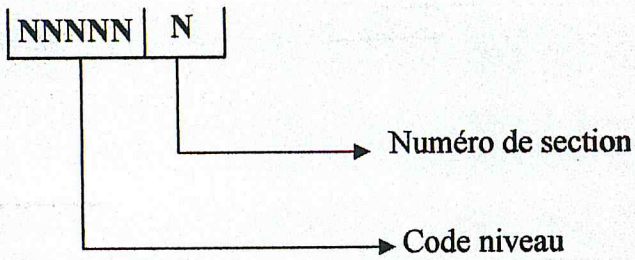
**Code niveau**



**Exemple :**

01135 : 5<sup>ème</sup> année

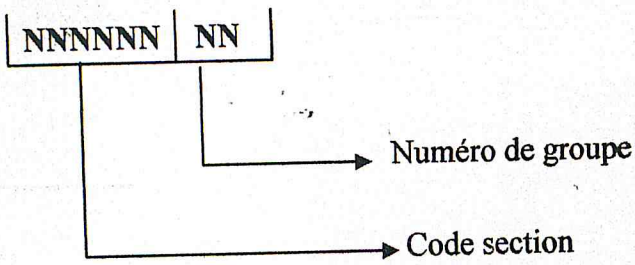
**Code section**



**Exemple :**

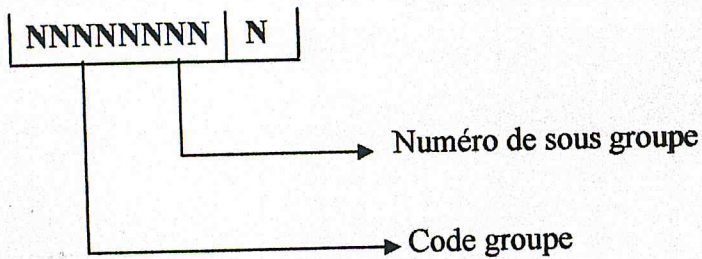
011351 : Section 1

**Code groupe**



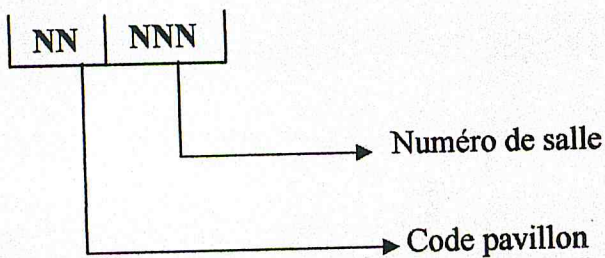
**Exemple :**  
0113513 :groupe 3.

**Code sous groupe**



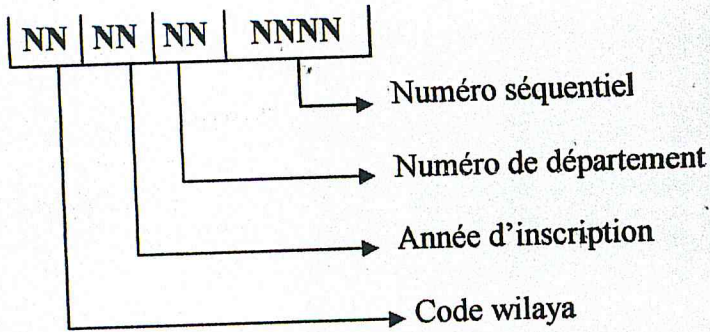
**Exemple :**  
01135132 Sous-groupe 2.

**Code local**



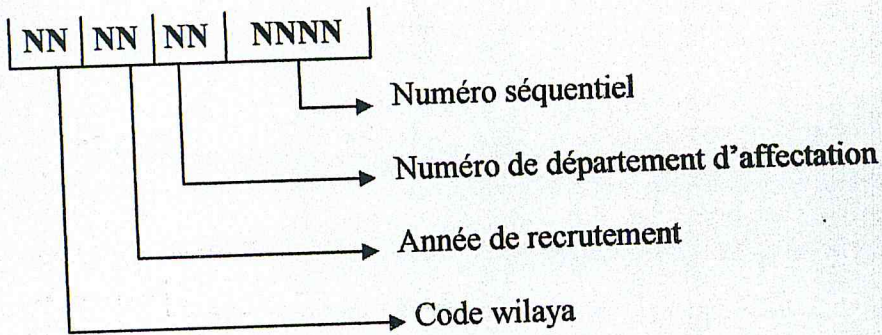
**Exemple :**  
17187 :salle 187 pavillon 17.

**Matricule étudiant**



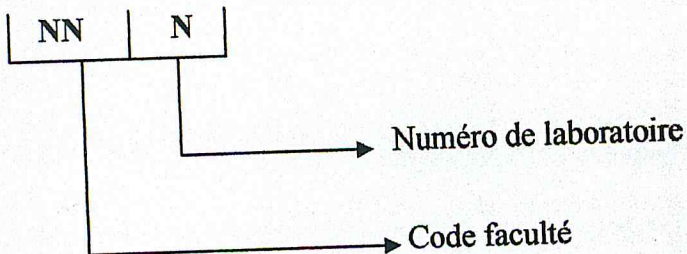
**Exemple :**  
0901110534 :

**Matricule enseignant :**



**Exemple :**  
0111010438

**Code laboratoire**



**Exemple :**  
131 : Laboratoire 1 de la faculté des sciences.

#### 4. Modèle conceptuel de données :

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

## Description des individus et des relations :

## 1- Liste des individus

Individu	Identifiant	Propriété
Année	ANN	ANN
Bac	CODE_BAC	CODE_BAC SERIE_BAC
Communication	CODE_COMM	CODE_COMM TITRE_COMM TYPE_COMM
Cycle	CODE_CYCLE	CODE_CYCLE LIB_CYCLE
Date	DATE	DATE
Département	CODE_DEPT	CODE_DEPT LIB_DEPT TEL_DEPT_EXT TEL_DEPT_INT FAX_DEPT
Diplome	CODE_DIPM	CODE_DIPM LIB_DIPM
Enseignant	MAT_ENS	MAT_ENS NOM_ENS PRN_ENS DATE_NAIS_ENS ADR_ENS TYPE_RECRUT SIT_FAM_ENS TYPE_COMPTE NUM_COMPTE
Equipe	CODE_EQUIPE	CODE_EQUIPE LIB_EQUIPE
Etudiant	MAT_ETD	MAT_ETD NOM_ETD PRN_ETD DATE_NAIS_ETD ADR_ETD SEXE_ETD NBRE_F_S NBRE_F_S_UNIV PRN_PERE_ETD NOM_MERE_ETD PRN_MERE_ETD ADR_PART_ETD PROF_PERE_ETD PROF_MERE_ETD TEL_ETD VALID_ETD OBS_ETD

Individu	Identifiant	Propriété
Faculte	CODE_FAC	CODE_FAC LIB_FAC TEL_FAC_EXT TEL_FAC_INT FEX_FAC
Grade recherche	CODE_GRADE_RECH	CODE_GRADE_RECH LIB_GRADE_RECH MONT_INDEMENT
Grade_pedagogique	CODE_GRADE_PEDAG	CODE_GRADE_PEDAG LIB_GRADE_PEDAG
Groupe	CODE_GRP	CODE_GRP LIB_GRP
Journé	JOURS	JOURS
Labo recherche	CODE_LABO	CODE_LABO LIB_LABO
Local	CODE_LOC	CODE_LOC NUM_SALLE CAP_LOC TYPE_LOC LIB_LOC
Membre externe	CODE_MBRE_EXT	CODE_MBRE_EXT NOM_MBRE_EXT PRN_MBRE_EXT ADR_MNRE_EXT
Module	CODE_MOD	CODE_MOD LIB_MOD COEF_MOD CRD_MOD NBRE_HEURE_COURS NBRE_HEURE_TD NBRE_HEURE_TP VOL_HORR_MOD
Niveau	CODE_NIV	CODE_NIV LIB_NIV
Niveau d'instruction mere	CODE_NIV_INS_MERE	CODE_NIV_INS_MERE LIB_NIV_INS_MERE
Niveau d'instruction pere	CODE_NIV_INS_PERE	CODE_NIV_INS_PERE LIB_NIV_INS_PERE
Organisme	CODE_ORG	CODE_ORG LIB_ORG ADR_ORG TEL_ORG FAX_ORG
Pavillon	CODE_PAV	CODE_PAV NUM_PAV LIB_PAV
Projet	CODE_PRJ	CODE_PRJ LIB_PRJ



Individu	Identifiant	Propriété
		CHEF_PRJ DATE DBT_PRJ DATE FIN_PRJ
Publication	CODE_PUBL	CODE_PUBL TITRE_PUBL
Responsabilite	CODE_RESP	CODE_RESP LIB_RESP
Sanction	CODE_SANC	CODE_SANC LIB_SANC
Seance	CODE_SNS	CODE_SNS
Section	CODE_SEC	CODE_SEC LIB_SEC
Seminaire	CODE_SEM	CODE_SEM TITRE_SEM DATE DBT_SEM DATE FIN_SEM
Services centraux	CODE_SERV	CODE_SERV LIB_SERV
Sous groupe	CODE_S_GRP	CODE_S_GRP LIB_S_GRP
Soutenance	CODE_SOUT	CODE_SOUT DATE_SOUT TITRE_SOUT RESUME_SOUT MOT_CLE_SOUT
Spécialité	CODE_SPEC	CODE_SPEC LIB_SPEC
Stage	CODE_STG	CODE_STG LIB_STG
Theme	CODE_THM	CODE_THM TITRE_THM
Type de participation	CODE_PARTIC	CODE_PARTIC LIB_PARTIC
Type d'examen	CODE_EXAM	CODE_EXAM LIB_EXAM COEF_EXAM
Type d'unite	CODE_TYPE_UNITE	CODE_TYPE_UNITE LIB_TYPE_UNITE
Type module	CODE_TYPE_MOD	CODE_TYPE_MOD LIB_TYPE_MOD
Type organisme	CODE_TYPE_ORG	CODE_TYPE_ORG LIB_TYPE_ORG
Type séance	CODE_TYPE_SNS	CODE_TYPE_SNS HEURE_DBT_MAT DUREE_SNS REPO_SNS REPO_MIDI NBRE_SNS

Individu	Identifiant	Propriété
Unite enseignement	CODE_UNITE_PEDAG	CODE_UNITE_PEDAG LIB UNIT PEDAG
Wilaya	CODE_WIL	CODE_WIL LIB WIL

## 2- Liste des relations

Relations	Individus	Identifiant	Card	Propriétés
A comme	Theme Seminaire		0,n 1,1	
a comme cycle	Spécialité Cycle		1,1 0,n	
A comme nationalite1	Etudiant Nationalite		1,1 0,n	
A comme nationalite2	Enseignant Nationalite		1,1 0,n	
A comme theme	Theme Soutenance		0,n 1,1	
A comme type	Type module Module		0,n 1,1	
A lieu	Communication Soutenance		1,n 1,1	
A un grade	Enseignant Grade pedagogique	MAT_ENS CODE_GRADE_PED AG	1,n 0,n	DATE_OBT_GRAD
A un type	Type unite Unite enseignement		1,n 1,1	
A une	Soutenance Enseignant Diplome	CODE_SOUT MAT_ENS CODE_DIPM	0,n 0,n 0,n	
A une soutenance	Etudiant Soutenance	MAT_ETD CODE_SOUT	0,n 1,n	
Abandonne	Annee Etudiant	ANN MAT_ETD	0,n 0,n	
Absence examen	Etudiant Type examen Module	MAT_ETD CODE_EXAM CODE_MOD	0,n 0,n 0,n	
Absent	Etudiant Seance Date	MAT_ETD CODE_SNS DATE	0,n 0,n 0,n	
Affecter	Departement Enseignant		1,n 0,n	
Affecter dep	Local Journee Seance Departement	CODE_LOC JOUR CODE_SNS	0,n 0,n 0,n 0,n	
Appartient	Groupe Section		1,1 1,n	
Appartient equi	Equipe Enseignant	CODE_EQUIPE MAT_ENS	1,n 0,n	
Appartient fac	Labo recherche faculte		1,1 0,n	

Relations	Individus	Identifiant	Card	Propriétés
Appartient groupe	Sous groupe Groupe		1,1 1,n	
Appartient org	Organisme Membre externe	CODE_ORG CODE_MBRE_EXT	0,n 0,n	
Appartient_niveau	Etudiant Niveau	MAT_ETD CODE_NIV	1,1 0,n	
Avoir cours	Séance Journee Local Section	CODE_LOC CODE_SNS JOUR	0,n 0,n 0,n 0,n	
Avoir grade rech	Enseignant Grade recherche		0,1 0,n	
Avoir grade rech1	Grade recherche Membre externe		0,n 0,1	
Avoir lieu	Type examen Module Local Section	CODE_LOC CODE_EXAM CODE_MOD	1,n 0,n 0,n 1,n	
Avoir module	Local Journee Séance Module	CODE_LOC JOUR CODE_SNS	0,n 0,n 0,n 0,n	
Avoir note	Type d'examen Module Etudiant	CODE_EXAM CODE_MOD MAT_ETD	0,n 0,n 0,n	NOTE
Avoir sanction ens	Enseignant Sanction Date	MAT_ENS CODE_SANC DATE	0,n 0,n 0,n	
Avoir sanction etd	Sanction Etudiant Date	CODE_SANC MAT_ETD DATE	0,n 0,n 0,n	
Avoir TD	Séance Journé Local Groupe	CODE_LOC JOUR CODE_SNS	0,n 0,n 0,n 0,n	
Avoir TP	Sous groupe Journé Séance Local	CODE_LOC JOUR CODE_SNS	0,n 0,n 0,n 0,n	
Bloque	Année Etudiant	ANN MAT_ETD	0,n 0,n	
Contient	Departement Faculte		1,1 1,n	
Contient dep	Pavillon Departement		0,n 1,1	
Contient fac	Pavillon Faculte		0,n 1,1	
Contient fac bibl	Faculte Bibliotheque		0,1 0,1	
Contient salle	Local Pavillon		1,1 0,n	
Correspond	Type séance Séance		1,n 1,1	
Dans une	Communication Theme		1,1 0,n	
Deligue de groupe	Groupe Etudiant		1,1 0,1	

Relations	Individus	Identifiant	Card	Propriétés
Deligue de section	Etudiant Section		0,1 1,1	
Diplomer	Etudiant Diplome	MAT_ETD CODE_DIPM	0,n 0,n	ANN_UNIV
D'un	Niveau Section		0,n 1,1	
Encadre ens	Enseignant Enseignant Diplome	MAT_ENS ENS_MAT_ENS CODE_DIPM	0,n 0,n 0,n	
Encadre mem ext	Membre externe Enseignant Diplome	CODE_MBRE_EXT MAT_ENS CODE_DIPM	0,n 0,n 0,n	
Encadrer ens	Stage Etudiant Enseignant	CODE_STG MAT_ETD MAT_ENS	0,n 0,n 0,n	
Encadrer mem ext	Etudiant Stage Membre externe	MAT_ETD CODE_STG CODE_MBRE_EXT	0,n 0,n 0,n	
Encadreur interne	Soutenance Enseignant	CODE_SOUT MAT_ENS	0,n 0,n	QUALITE_ENCAD_INT
Est de type	Type organisme Organisme		0,n 1,1	
Est inscrit	Enseignant Diplome Specialite	MAT_ENS CODE_DIPM CODE_SPEC	0,n 0,n 0,n	DATE_INSC_ENS SUJET MOT_CLE RESUME
Est responsable dep	Responsabilite Departement Enseignant	MAT_ENS	1,n 1,n 0,1	
Est responsable fac	Responsabilite Faculte Enseignant	MAT_ENS	1,n 1,n 0,1	
Est responsable labo	Labo recherche Enseignant		1,1 0,1	
Est responsable sc	Responsabilite Service centraux Enseignant	MAT_ENS	1,n 1,n 0,1	
Exposer	Seminaire Organisme	CODE_SEM CODE_ORG	0,n 1,n	
Faire comm	Communication Enseignant	CODE_COMM MAT_ENS	0,n 0,n	
Faire Cours	Enseignant Module Section	CODE_SEC CODE_MOD	0,n 0,n 0,n	
Faire pub	Enseignant Publication	MAT_ENS CODE_PUBL	0,n 1,n	
Faire TD	Groupe Module Enseignant	CODE_GRP CODE_MOD	0,n 0,n 0,n	
Faire TP	Sous groupe Module Enseignant	CODE_S_GRP COE_MOD	0,n 0,n 0,n	
Fait	Communication Membre externe	CODE_COMM CODE_MBRE_EXT	0,n 0,n	
Fait des recherches	Labo recherche Equipe		1,n 1,1	

Relations	Individus	Identifiant	Card	Propriétés
Fait partie	Etudiant Sous groupe		1,1 1,n	
Jury externe	Soutenance Membre externe	CODE_SOUT CODE_MBRE_EXT	0,n 0,n	QUALITE_JURY_EXT
Jury int	Soutenance Enseignant	CODE_SOUT MAT_ENS	0,n 0,n	QUALITE_JURY_INT
Lieu de nai ens	Wilaya Enseignant		0,n 1,1	
Lieu de nai etd	Etudiant Wilaya		1,1 0,n	
Module_Acquit	Etudiant Module	MAT_ETD CODE_MOD	0,n 0,n	MOY_ACQUIT
Niveau d'enseignement	Niveau Module		1,n 1,1	
Niveau mere	Niveau d'instruction mere Etudiant		0,n 1,1	
Niveau pere	Niveau d'instruction pere Etudiant		0,n 1,1	
Obtenir	Etudiant Wilaya Bac Année	MAT_ETD	1,1 0,n 0,n 0,n	NUM_BAC_ETD MOYENNE MONTION_ETD
Obtenir un	Enseignant Diplome Specialite	MAT_ENS CODE_DIPM CODE_SPEC	1,n 0,n 0,n	DATE_OBT_DIPM LIEU_OBT_DIPM
Participe	Type de participation Seminaire Enseignant	CODE_PARTIC CODE_SEM MAT_ENS	0,n 1,n 0,n	
Possede	Departement Specialite		1,n 1,1	
Possede les specialite	Niveau Specialite		1,1 1,n	
Pour suit	Etudiant Organisme Stage	MAT_ETD CODE_ORG CODE_STG	1,n 0,n 1,n	DATE_DBT_STG DATE_FIN_STG
Propriété de module	Unite enseignement Module		0,1 1,n	
Rattacher	Labo recherche Projet		0,n 0,1	
Reclame	Module Etudiant	CODE_MOD MAT_ETD	0,n 0,n	RECL_ETD DATE_RECLAM TRAIT_REC VISITE_RECL IGNORER_REC
Reside ens	Wilaya Enseignant		0,n 1,1	
Reside etd	Etudiant Wilaya		1,1 0,n	
Responsable de section	Enseignant Section		0,n 1,1	
Responsable_Module	Module Section Enseignant	CODE_MOD CODE_SEC	0,n 1,n 0,n	

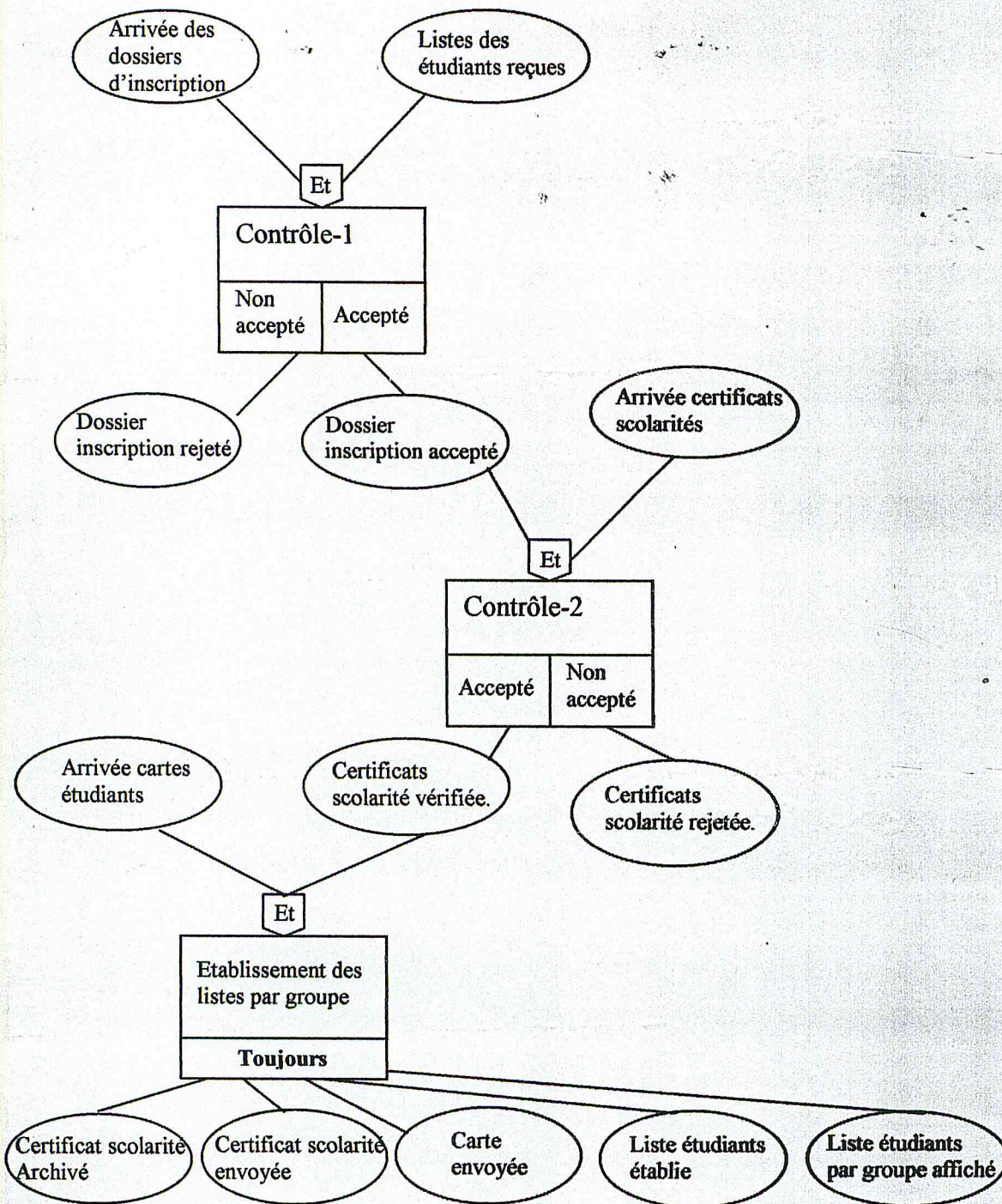
Relations	Individus	Identifiant	Card	Propriétés
S'absente	Enseignant Séance Date	MAT_ENS CODE_SNS DATE	0,n 0,n 0,n	
se déroule	Type d'examen Module	CODE_EXAM CODE_MOD	1,n 1,n	DATE_EXAM HEURE_DBT_EXAM HEURE_FIN_EXAM
Sponsorer	Seminaire Organisme	CODE_SEM CODE_ORG	0,n 0,n	
Surveiller	Type d'examen Module Enseignant Local	CODE_EXAM CODE_MOD CODE_LOC	0,n 0,n 0,n 0,n	
Transférer	Année Organisme Etudiant	MAT_ETD	0,n 0,n 0,1	
Travaille	Projet Enseignant	CODE_PRJ MAT_ENS	0,n 0,n	
Travaille mem ext	Membre externe Projet	CODE_MBRE_EXT CODE_PRJ	0,n 0,n	

### 5. Modèle conceptuel des traitements (M.C.T) :

Le modèle conceptuel des traitements permet de traiter la dynamique du système d'information, c'est-à-dire les opérations qui sont réalisées en fonction d'événements.

Ce modèle permet donc de représenter de façon schématique l'activité d'un système d'information sans faire référence à des choix organisationnels ou des moyens d'exécution, c'est-à-dire qu'il permet de définir simplement ce qui doit être fait, mais il ne dit pas quand, comment ni où...

### Processus N°1 'Inscription'



**Descriptif N°1 :**

Opération	Action
Contrôle-1	Contrôle des dossiers d'inscriptions (D.I) Si Accepté : D.I vérifié. Sinon accepté : D.I rejeté.
Contrôle-2	Vérification des certificats scolarité (C.S) Si Accepté : C.S vérifié. Sinon Accepté : C.S rejeté.
Etablissement des listes des étudiants par groupe	Archivage L.E.O. Envoie carte étudiant. Envoie certificats scolarité à l'étudiant. Envoie certificats scolarité au V/Rectorat.

*N.B Les autres processus dans l'Annexe E.*

**6. Modèle logique de données :**

Le modèle logique des données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisées lors des traitements.

Ainsi, le modèle logique est dépendant du type de base de données utilisé

**Le modèle logique de données :**

- A\_UN\_GRADE(MAT\_ENS, CODE\_GRADE, PEDAG, DATE\_OBT\_GRAD)
- A\_UNE(CODE\_SOUT, MAT\_ENS, CODE\_DIPM)
- A\_UNE\_SOUTENANCE(MAT\_ETD, CODE\_SOUT)
- ABONDONNE(ANN, MAT\_ETD)
- ABSENCE\_EXAMEN(MAT\_ETD, CODE\_EXAM, CODE\_MOD)
- ABSENT(MAT\_ETD, CODE\_SNS, DATE)
- AFFECTER\_DEP(CODE\_LOC, JOUR, CODE\_SNS, CODE\_DEPT)
- ANNEE(ANN)
- APPARTIENT\_EQUI(CODE\_EQUIPE, MAT\_ENS)
- APPARTIENT\_ORG(CODE\_ORG, CODE\_MBRE\_EXT)
- AVOIR\_COURS(CODE\_LOC, CODE\_SNS, JOUR, CODE\_SEC)
- AVOIR\_LIEU(CODE\_LOC, CODE\_EXAM, CODE\_MOD, CODE\_SEC)
- AVOIR\_MODULE(CODE\_LOC, JOUR, CODE\_SNS, CODE\_MOD)
- AVOIR\_NOTE(CODE\_EXAM, CODE\_MOD, MAT\_ETD, NOTE)
- AVOIR\_SANCTION\_ENS(MAT\_ENS, CODE\_SANC, DATE)
- AVOIR\_SANCTION\_ETD(MAT\_ETD, CODE\_SANC, DATE)
- AVOIR\_TD(CODE\_LOC, JOUR, CODE\_SNS, CODE\_GRP)
- AVOIR\_TP(CODE\_LOC, JOUR, CODE\_SNS, CODE\_S\_GRP)
- BAC(CODE\_BAC, SERIE\_BAC)
- BLOQUE(ANN, MAT\_ETD)
- COMMUNICATION(CODE\_COMM, CODE\_THM, TITRE\_COMM, TYPE\_COMM)



- DATE(DATE)
- DEPARTEMENT(CODE\_DEPT,CODE\_FAC,CODE\_PAV,LIB\_DEPT,TEL\_DEPT\_EXT,TEL\_DEP\_INT,FAX\_DEPT)
- DIPLOME(CODE\_DIPM,LIB\_DIPM)
- DIPLOMER(MAT\_ETD,CODE\_DIPM,ANN\_UNIV)
- ENCADRE\_ENS(MAT\_ENS,ENS\_MAT\_ENS,CODE\_DIPM)
- ENCADRE\_MEM\_EXT(MAT\_MBRE\_EXT,MAT\_ENS,CODE\_DIPM)
- ENCADRER\_ENS(CODE\_STG,MAT\_ETD,MAT\_ENS)
- ENCADRER\_MEM\_EXT(CODE\_STG,MAT\_ETD,CODE\_MBRE\_EXT)
- ENCADREUR\_INTERNE(CODE\_SOUT,MAT\_ENS,QUALITE\_ENCAD\_INT)
- ENSEIGNANT(MAT\_ENS,CODE\_NATIO,CODE\_WIL,WIL\_CODE\_WIL,CODE\_DEPT,CODE\_GRADE\_RECH,NOM\_ENS,PRN\_ENS,DATE\_NAIS\_ENS,ADR\_ENS,TYPE\_RECRT,SIT\_FAM\_ENS,TYPE\_COMPTE,NUM\_COMPTE)
- EQUIPE(CODE\_EQUIPE,CODE\_LABO,LIB\_EQUIPE)
- EST\_INSCRIT(MAT\_ENS,CODE\_DIPM,CODE\_SPEC,DATE\_INSC\_ENS,SUJET,MOT\_CLE,RESUME)
- EST\_RESPONSABLE\_DEP(MAT\_ENS,CODE\_DEPT,CODE\_RESP)
- EST\_RESPONSABLE\_FAC(MAT\_ENS,CODE\_FAC,CODE\_RESP)
- EST\_RESPONSABLE\_SC(MAT\_ENS,CODE\_SERV,CODE\_RESP)
- ETUDIANT(MAT\_ETD,CODE\_NIV\_INS\_PERE,CODE\_NIV\_INS\_MERE,CODE\_NATIO,CODE\_WIL,WIL\_CODE\_WIL,CODE\_S\_GRP,CODE\_NIV,NOM\_ETD,PRN\_ETD,DATE\_NAIS\_ETD,ARD\_ETD,SEXE\_ETD,NBRE\_F\_S,NBRE\_F\_S\_UNIV,PRN\_PERE\_ETD,NOM\_MERE\_ETD,PRN\_MERE\_ETD,ADR\_PART\_ETD,PROF\_PERE\_ETD,PROF\_MERE\_ETD,TEL\_ETD,VALID\_ETD,OBES\_ETD)
- EXPOSER(CODE\_SEM,CODE\_ORG)
- FACULTE(CODE\_FAC,CODE\_PAV,LIB\_FAC,TEL\_FAC\_EXT,TEL\_FAC\_INT,FAX\_FAC)
- FAIRE\_COMM(CODE\_COMM,MAT\_ENS)
- FAIRE\_COURS(CODE\_SEC,CODE\_MOD,MAT\_ENS)
- FAIRE\_PUB(MAT\_ENS,CODE\_BUBL)
- FAIRE\_TD(CODE\_GRP,CODE\_MOD,MAT\_ENS)
- FAIRE\_TP(CODE\_S\_GRP,CODE\_MOD,MAT\_ENS)
- FAIT(CODE\_COMM,CODE\_MBRE\_EXT)
- GRADE\_RECHERCHE(CODE\_GRADE\_RECH,LIB\_GRADE\_RECH,MONT\_INDEMENT)
- GRADE\_PEDAGOGIQUE(CODE\_GRADE\_PEDAG,LIB\_GRADE\_PEDAG)
- GROUPE(CODE\_GRP,CODE\_SEC,MAT\_ETD,LIB\_GRP)
- JOURNE(JOUR)
- JURY\_EXTERNE(CODE\_SOUT,CODE\_MBRE\_EXT,QUALITE\_JURY\_EXT)
- JURY\_INT(CODE\_SOUT,MAT\_ENS,QUALITE\_JURY\_INT)
- LABO\_RECHERCHE(CODE\_LABO,CODE\_FAC,MAT\_ENS,LIB\_LABO)
- LOCAL(CODE\_LOC,CODE\_PAV,NUM\_SALLE,CAP\_LOC,TYPE\_LOC,LIB\_LOC)
- MEMBRE\_EXTERNE(CODE\_MBRE\_EXT,CODE\_GRADE\_RECH,NOM\_MBRE\_EXT,PRN\_MBRE\_EXT,ADR\_MBRE\_EXT)
- MODULE(CODE\_MOD,CODE\_UNITE\_PEDAG,CODE\_NIV,CODE\_TYPE\_MOD,LIB\_MOD,COEF\_MOD,CRD\_MOD,NBRE\_HEURE\_COURS,NBRE\_HEURE\_TD,NBRE\_HEURE\_TP,VOL\_HORR\_MOD)
- MODULE\_ACQUIT(MAT\_ETD,CODE\_MOD,MOY\_ACQUIT)
- NIVEAU(CODE\_NIV,CODE\_SPEC,LIB\_NIV)

- NIVEAU\_D\_INSTRUCTION\_MERE(CODE\_NIV\_INS\_MERE,LIB\_NIV\_INS\_MERE)
- NIVEAU\_D\_INSTRUCTION\_PERE(CODE\_NIV\_INS\_PERE,LIB\_NIV\_INS\_PERE)
- OBTENIR(MAT\_ETD,CODE\_WIL,CODE\_BAC,ANN,NUM\_BAC\_ETD,MOYENNE, MONTION\_ETD)
- OBTENIR\_UN(MAT\_ENS,CODE\_DIPM,CODE\_SPEC,DATE\_OBT\_DIPM,LIEU\_OBT\_DIPM)
- ORGANISME(CODE\_ORG,CODE\_TYPE\_ORG,LIB\_ORG,ADR\_ORG,TEL\_ORG,FA X\_ORG)
- PARTICIPE(CODE\_PARTIC,CODE\_SEM,MAT\_ENS)
- PAVILLON(CODE\_PAV,NUM\_PAV,LIB\_PAV)
- POUR\_SUIT(MAT\_ETD,CODE\_ORG,CODE\_STG,DATE\_DBT\_STG,DATE\_FIN\_ST G)
- PROJET(CODE\_PRJ,CODE\_LABO,LIB\_PRJ,CHEF\_PRJ,DATE\_DBT\_PRJ,DATE\_FI N\_PRJ)
- PUBLICATION(CODE\_PUBL,TITRE\_PUBL)
- RECLAME(CODE\_MOD,MAT\_ETD,RECL\_ETD,DATE\_RECLAM,TRAIT\_REC,VISI TE\_RECL,IGNORER\_RECL)
- RESPONSABILITE(CODE\_RESP,LIB\_RESP)
- RESPONSABLE\_MODULE(CODE\_MOD,CODE\_SEC,MAT\_ENS)
- S\_ABSENTE(MAT\_ENS,CODE\_SNS,DATE)
- SANCTION(CODE\_SANC,LIB\_SANC)
- SE\_DEROULE(CODE\_EXAM,CODE\_MOD,DATE,EXAM,HEURE\_DBT\_EXAM,HE URE\_FIN\_EXAM)
- SEANCE(CODE\_SNS,CODE\_TYPE\_SNS)
- SECTION(CODE\_SEC,CODE\_NIV,MAT\_ENS,MAT\_ETD,LIB\_SEC)
- SEMINAIRE(CODE\_SEM,CODE\_THM,TITRE\_SEM,DATE\_DBT\_SEM,DATE\_FIN\_ SEM)
- SERVICE\_CENTRAUX(CODE\_SERV,LIB\_SERV)
- SOUS\_GROUPE(CODE\_S\_GRP,CODE\_GRP,LIB\_S\_GRP)
- SOUTENANCE(CODE\_SOUT,CODE\_THM,DATE\_SOUT,TITRE\_SOUT,RESUME\_S OUT,MOT\_CLE\_SOUT)
- SPECIALITE(CODE\_SPEC,CODE\_DEPT,CODE\_CYCLE,LIB\_SPEC)
- SPONSORER(CODE\_SEM,CODE\_ORG)
- STAGE(CODE\_STG,LIB\_STG)
- SURVEILLER(CODE\_EXAM,CODE\_MOD,CODE\_LOC,MAT\_ENS)
- THEME(CODE\_THM,LIB\_THM)
- TRANSFERER(MAT\_ETD,CODE\_ORG,ANN)
- TRAVAILLE(CODE\_PRJ,MAT\_ENS)
- TRAVAILLE\_MEM\_EXT(CODE\_MBRE\_EXT,CODE\_PRJ)
- TYPE\_DE\_PARTICIPATION(CODE\_PARTIC,LIB\_PARTIC)
- TYPE\_D\_EXAMEN(CODE\_EXAM,LIB\_EXAM,COEF\_EXAM)
- TYPE\_D\_UNITE(CODE\_TYPE\_UNITE,LIB\_TYPE\_UNITE)
- TYPE\_MODULE(CODE\_TYPE\_MOD,LIB\_TYPE\_MOD)
- TYPE\_ORGANISME(CODE\_TYPE\_ORG,LIB\_TYPE\_ORG)
- TYPE\_SEANCE(CODE\_TYPE\_SNS,HEURE\_DBT\_MAT,DUREE\_SNS,REPO\_SNS, REPO\_MIDI,NBRE\_SNS)
- UNITE\_ENSEIGNEMENT(CODE\_UNITE\_PEDAG,CODE\_TYPE\_UNITE,LIB\_UNIT E\_PEDAG)
- WILAYA(CODE\_WIL,LIB\_WIL)

## 7. Modèle organisationnel des traitements (M.O.T) :

Le Modèle Conceptuel des Traitements (MCT) donne une purement fonctionnelle de l'entreprise, tandis que le Modèle organisationnel des traitements (MOT) prend en considération le concept d'organisation existant qu'il faut prendre en compte dans le nouveau système, pour décrire le MOT on a utilisé un formalisme qui est le même que celui décrit lors de la description de MCT.

Le découpage se fait de la même façon que celui établi au niveau du MCT sauf le terme PROCESSUS devient PROCEDURE, OPERATION devient PHASE, ACTION devient TACHE.

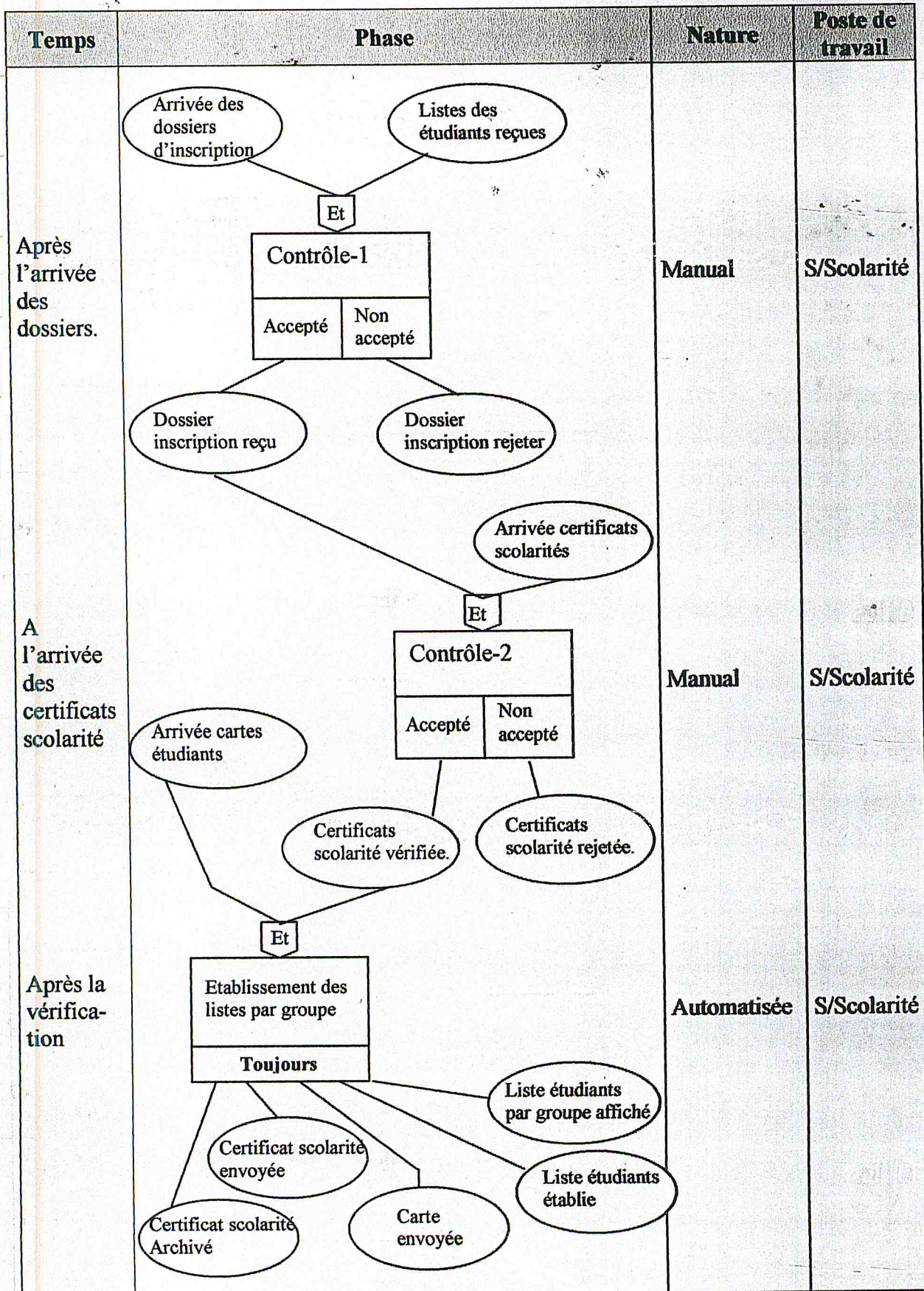
Chaque phase sera déterminée par :

- La nature de terme traitement (automatique, manuel),
- Le poste de travail où s'exécute la phase,
- Le terme d'exécution de la phase.

Les règles d'organisation :

Code	Désignation de la règle
01	Après chaque fin de trimestre le procès verbal modulaire est établie au niveau du service de scolarité.
02	A la fin de l'année le procès verbal annuel est établie au niveau du service de scolarité.
03	L'étudiant peut faire le transfert lorsqu'il a l'avis favorable de l'organisme d'accueil.
04	Au début de 2 <sup>ème</sup> semestre de la cinquième année d'étude, les sujets de projet de fin d'étude des étudiants sont validés par les commissions de P.F.E.
05	Dans chaque période de soutenance les autorisations de soutenances sont établies par le poste PFE.
06	A chaque fin d'étude les fiches de renseignements sont établies par le service des stages.

Procédure N° 1 'Inscription' :



## Descriptif N°1:

Opération	Action
Contrôle 1	Contrôle des dossiers d'inscriptions - Si accepté : dossier accepté - Si non : dossier rejeté
Contrôle 2	Contrôle des certificats de scolarité - Si accepté : Certificat de scolarité accepté - Si non : Certificat de scolarité rejeté
Etablissement liste d'étudiants par groupe	Etablissement liste des étudiants par groupe Affichage des listes des étudiants par groupe Envoi copie C.S au V/Rectorat Archivage d'une copie au niveau de la scolarité Renouvellement de la carte d'étudiant

*N.B :*

*Les autres procédures dans Annexe E.*

## 8. Le modèle physique de données :

Cette étape consiste à implémenter le modèle dans le SGBD, c'est-à-dire le traduire dans un langage de définition de données.

Le langage généralement utilisé pour ce type d'opération est le SQL, et plus spécialement le langage de définition de données du SQL.

*N.B :*

*La liste des tables dans Annexe E.*

## 9. Validation des vues externes:

Cette étape concerne que les phases automatisées, pour les phases manuelles le travail s'est achevé avec la description des tâches qu'elles englobent et les règles d'organisation qui les gouvernent.

Il s'agit donc de construire pour chacune des phases automatisées des vues externes en utilisant le même formalisme que celui utilisé pour présenter le MCD mais l'élaboration des vues externes doit se faire en ignorant le MCD.

Son intérêt principal est de permettre la validation du MCD, c'est à dire mettre en accord les données avec les traitements.

Les étapes de la validation sont :

1. Elaboration des vues externes en utilisant le formalisme individuel,
2. Validation du MCD par les modèles externes comme suit :
  - Validation des propriétés,
  - Validation des individus,
  - Validation des relations,
  - Validation des cardinalités.

On distingue deux types de validation :

### a)- Validation par les modèles externes en M.A.J :

A partir des documents en entrée, on élabore le modèle externe en M.A.J

#### Règles de validation :

- Pour chaque propriété externe, rechercher l'individu ou la relation conceptuelle équivalent,
- Vérifier que chaque individu conceptuel invoqué est identifié au niveau du modèle externe,
- Vérifier que les cardinalités des relations externes sont incluses dans les cardinalités des relations conceptuelles,
- Identifier pour chaque propriété externe son rôle dans l'individu conceptuel.

### b)- Validation par les modèles externes en consultation :

A partir des documents en sortie, on élabore le modèle externe en consultation.

**Règles de validation :**

- Pour chaque propriété externe, rechercher l'individu ou la relation conceptuelle équivalent,
- Vérifier que chaque individu conceptuel invoqué est accessible au niveau du modèle externe,
- Vérifier que chaque relation conceptuelle invoquée est accessible au niveau du modèle externe,
- Vérifier que les cardinalités des relations externes sont incluses dans les cardinalités des relations conceptuelles.

**9.1. Validation en consultation :**

**1-Edition Relevé des notes**

Le document se présente comme suit :

Libellé département						
Matricule						
Nom						
prénom						
spécialité						
cycle						
année d'étude						
année universitaire						
Code module	Intitulé module	Juin	Synth	Ratt	Coefficient	Pondérée
-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----
					<b>Total coefficient</b>	<b>Total note</b>
					<b>Moyenne général</b>	

**Liste des données manipulées :**

Données	Désignation
Mat	Matricule étudiant
Nom etd	Nom étudiant
Prn etd	Prénom étudiant
Spec	Spécialité
Niv	Année d'étude
Code mod	Code module
Lib mod	Intitulé module
Coef mod	Coefficient module
Note Exam	Note modulaire dans un examen.
Juin	Moyenne session normale
Synth	Moyenne session synthèse

Données	Désignation
Ratt	Moyenne session rattrapage.
Pond	Pondérée module
Tot coef	Total coefficient
Tot note	Total notes
Moy gen	Moyenne général.

**Les propriétés :=**

$$* \text{Juin} = \frac{(\text{somme des EMD} + \text{Notes des TD} + \text{Notes de TP})}{(\text{Nombre des EMD} + 2)}$$

$$* \text{Synth} = \frac{(\text{Synthèse} * \text{Nombre des EMD} + \text{Notes des TD} + \text{Notes de TP})}{(\text{Nombre des EMD} + 2)}$$

\* Ratt = La moyenne modulaire de la session de rattrapage est celle obtenue ans l'examen de rattrapage.

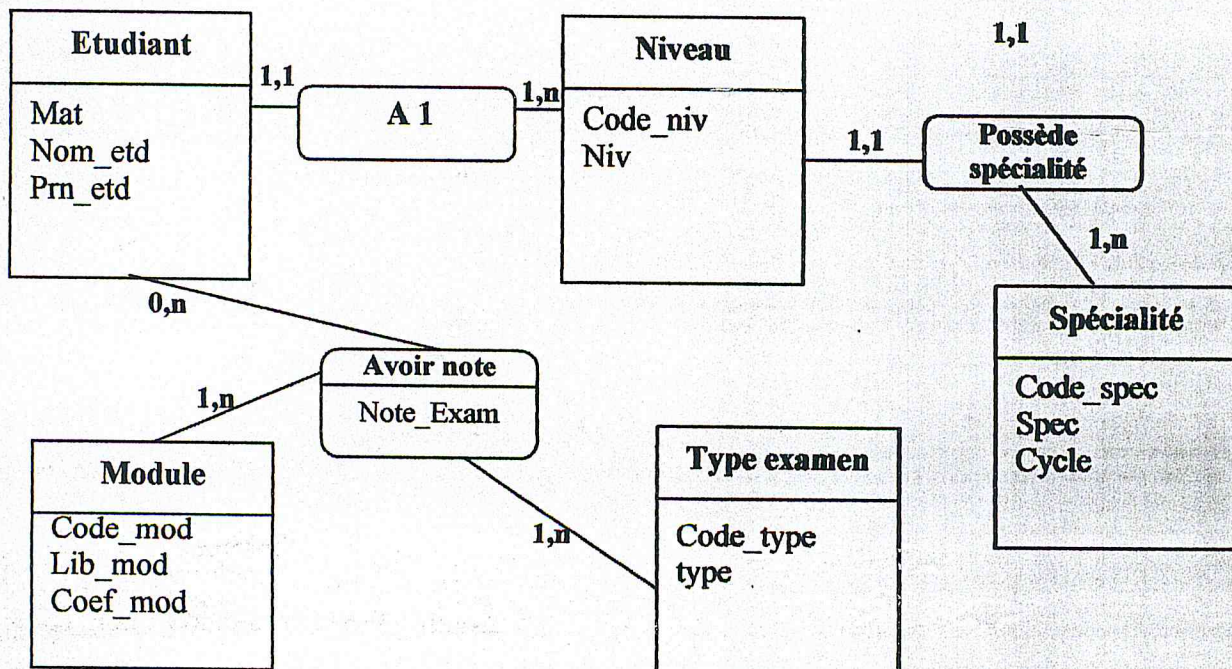
$$* \text{Pond} = \text{sup}(\text{Juin}, \text{Synth}, \text{Ratt}) * \text{Coef\_mod.}$$

$$* \text{Tot\_coef} = \sum \text{coef\_mod}$$

$$* \text{Moy\_gen} = \sum \text{Pond} / \text{Tot\_coef}$$

*N.B : On doit l'éliminer du model externe.*

**Modèle externe :**





**Validation des propriétés :**

Propriété externe	Propriété conceptuelle	Fonction	Individu invoqué	Validation
Mat	MAT_ETD	Identification	Etudiant	Oui
Nom_etd	NOM_ETD	Consultation		Oui
Prn_etd	PRN_ETD	Consultation		Oui
Code_niv	CODE_NIV	Identification	Niveau	Oui
Niv	LIB_NIV	Consultation		Oui
Code_spec	CODE_SPEC	Identification	Spécialité	Oui
Spec	LIB_SPEC	Consultation		Oui
Cycle	CYCLE	Consultation		Oui
Code_mod	CODE_MOD	Identification	Module	Oui
Lib_mod	LIB_MOD	Consultation		Oui
Coef_mod	COEF_MOD	Consultation		Oui
Note_Exam	NOTE	Consultation	Avoir note	Oui
Code_type	CODE_TYPE	Identification	Type examen	Oui
type	LIB_TYPE	Consultation		Oui

**Validation des objets :**

Les objets externes sont validés car les propriétés sont valides.

**Validation des relations et cardinalités :**

Relation externe	Objet externe	Cardinalité externe	Relation conceptuelle	Objet conceptuel	Cardinalité conceptuelle	Validation
A 1	Etudiant	1,1	A 1	Etudiant	1,1	Oui
	Niveau	1,n		Niveau	0,n	Oui
Avoir note	Etudiant	0,n	Avoir note	Etudiant	0,n	Oui
	Module	1,n		Module	0,n	Oui
	Type examen	1,n		Typeexamen	0,n	Oui
Possède spécialité	Niveau	1,1	Possède spécialité	Niveau	1,1	Oui
	Spécialité	1,n		Spécialité	1,n	Oui

Les relations et les cardinalités sont validées.

**Conclusion :**

Le modèle externe est validé.

**Validation des propriétés :**

Propriété externe	Propriété conceptuelle	Fonction	Individu invoqué	Validation
Mat	MAT_ETD	Identification	Etudiant	Oui
Nom_etd	NOM_ETD	Consultation		Oui
Prn_etd	PRN_ETD	Consultation		Oui
Code_niv	CODE_NIV	Identification	Niveau	Oui
Niv	LIB_NIV	Consultation		Oui
Code_sec	CODE_SEC	Identification	Section	Oui
Sec	LIB_SEC	Consultation		Oui
Code_grp	CODE_GRP	Identification	Groupe	Oui
grp	LIB_GRP	Consultation		Oui
Code_s_grp	CODE_S_GRP	Identification	Sous groupe	Oui
grp	LIB S_GRP	Consultation		Oui

**Validation des objets :**

Les objets externes sont validés car les propriétés sont valides.

**Validation des relations et cardinalités :**

Relation externe	Objet externe	Cardin externe	Relation conceptuelle	Objet conceptuel	Cardin conceptuelle	Validation
Fait partie	Etudiant	1,1	Fait partie	Etudiant	1,1	Oui
	Sous groupe	1,n		Sous groupe	1,n	Oui
Appartient groupe	Sous groupe	1,1	Appartient groupe	Sous groupe	1,1	Oui
	Groupe	1,n		Groupe	1,n	Oui
Appartient	Groupe	1,1	Appartient	Groupe	1,1	Oui
	Section	1,n		Section	1,n	Oui
D'un	Section	1,1	D'un	Section	1,1	Oui
	Niveau	1,n		Niveau	1,n	Oui

Les relations et les cardinalités sont validées.

**Conclusion :**

Le modèle externe est validé.

**9.2. Validation en mise à jours :**

**1- Inscription des étudiants**

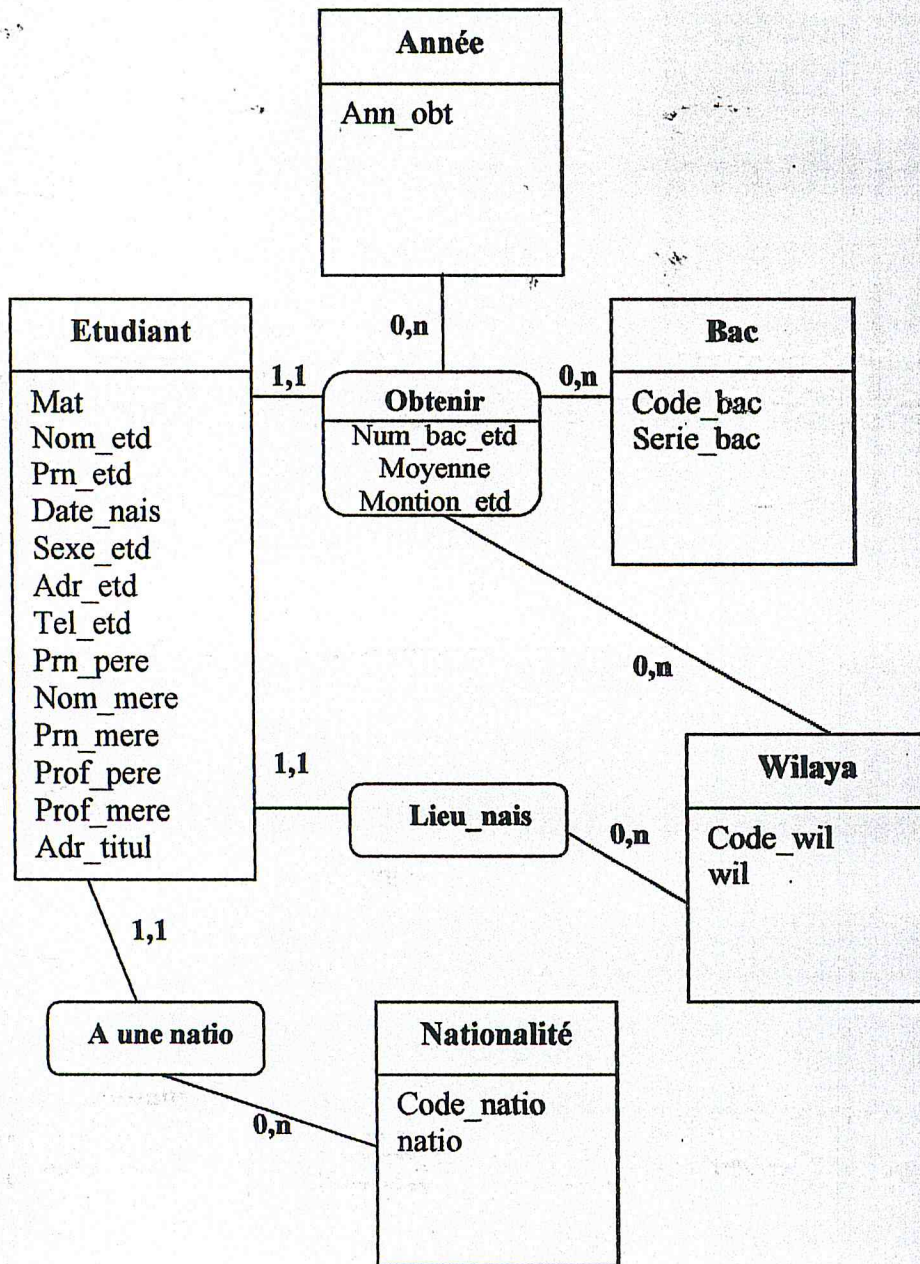
Le document se présente comme suit :

Specialité : .....	Niveau : .....	Année universitaire : .....
<b>Personnel</b>	<b>Fils de</b>	
Mat	Prn_pere	
Nom_etd	Nom_mere	
Prn_etd	Prn_mere	
Date_nais	Prof_pere	
Lieu_nais	Prof_mere	
Sexe_etd	Adr_titul	
Adr_etd		
Natio_etd		
Tel_etd		
<b>Bac</b>		
N_bac		
Serie_bac		
Moy_bac		
Mont_bac		
Wil_bac		
Ann_obt		

**Liste des données manipulées :**

Données	Désignation
Mat	Matricule étudiant
Nom_etd	Nom étudiant
Prn_etd	Prénom étudiant
Date_nais	Date de naissance étudiant
Wil	Lieu de naissance étudiant
Sexe_etd	Sexe étudiant
Adr_etd	Adresse étudiant
Natio_etd	Nationalité étudiant
Tel_etd	Téléphone étudiant
N_bac	Numéro de bac
Serie_bac	Série de bac
Moy_bac	Moyenne de bac
Mont_bac	Montions de bac
Wil_bac	Wilaya d'obtention du bac
Ann_obt	Année d'obtention bac
Prn_pere	Prénom du père.
Nom_mere	Nom de la mère.
Prn_mere	Prénom de la mère.
Prof_pere	Profession du père.
Prof_mere	Profession du mère.
Adr_titul	Adresse du titul

Modèle externe :



Validation des propriétés :

Propriété externe	Propriété conceptuelle	Fonction	Individu invoqué	Validation
Mat	MAT_ETD	Identification	Etudiant	Oui
Nom_etd	NOM_ETD	Consultation		Oui
Prn_etd	PRN_ETD	Consultation		Oui
Date_nais	DATE_NAIS_ETD	Consultation		
Sexe_etd	SEXE_ETD	Consultation		
Adr_etd	ADR_ETD	Consultation		
Tel_etd	TEL_ETD	Consultation		
Prn_pere	PRN_PERE_ETD	Consultation		
Nom mere	NOM_MERE_ETD	Consultation		

Propriété externe	Propriété conceptuelle	Fonction	Individu invoqué	Validation
Prn_mère Prof_pere Prof_mere Adr_titul	PRN_MERE_ETD PROF_PERE_ETD PROF_MERE_ETD ADR_PART_ETD	Consultation Consultation Consultation Consultation		
Code_bac Serie_bac	CODE_BAC SERIE_BAC	Identification Consultation	Bac	Oui Oui
Code_wil wil	CODE_WIL LIB_WIL	Identification Consultation	Wilaya	Oui Oui
Code_natio natio	CODE_NATIO LIB_NATIO	Identification Consultation Consultation	Nationalité	Oui Oui Oui
Ann_obt	ANN	Consultation	Année	Oui

Validation des objets :

Individu invoqué	Propriété externe	Propriété conceptuelle	Rôle
Etudiant	Mat	MAT_ETD	Identifiant
	Nom_etd	NOM_ETD	M.A.J
	Prn_etd	PRN_ETD	M.A.J
	Date_nais	DATE_NAIS_ETD	M.A.J
	Sexe_etd	SEXE_ETD	M.A.J
	Adr_etd	ADR_ETD	M.A.J
	Tel_etd	TEL_ETD	M.A.J
	Prn_pere	PRN_PERE_ETD	M.A.J
	Nom_mere	NOM_MERE_TED	M.A.J
	Prn_mere	PRN_MERE_ETD	M.A.J
	Prof_pere	PROF_PERE_ETD	M.A.J
	Prof_mere	PROF_MERE_ETD	M.A.J
	Adr_titul	ADR_PART_ETD	M.A.J
	Bac	Code_bac	CODE_BAC
Serie_bac		SERIE_BAC	Afficher
Wilaya	Code_wil	CODE_WIL	Identifiant
	wil	LIB_WIL	Afficher
Nationalité	Code_natio	CODE_NATIO	Identifiant
	natio	LIB_NATIO	Afficher
Année	Ann_obt	ANN	Identifiant

Les objets externes sont validés car les propriétés sont valides.

## Validation des relations et cardinalités :

Relation externe	Objet externe	Cardin externe	Relation conceptuelle	Objet conceptuel	Cardin conceptuelle	Validation
Fait partie	Etudiant	1,1	Fait partie	Etudiant	1,1	Oui
	Sous groupe	1,n		Sous groupe	1,n	Oui
Appartient groupe	Sous groupe	1,1	Appartient groupe	Sous groupe	1,1	Oui
	Groupe	1,n		Groupe	1,n	Oui
Appartient	Groupe	1,1	Appartient	Groupe	1,1	Oui
	Section	1,n		Section	1,n	Oui
D'un	Section	1,1	D'un	Section	1,1	Oui
	Niveau	1,n		Niveau	1,n	Oui

Les relations et les cardinalités sont validées.

**Conclusion :**

Le modèle externe est validé.

**10. Conclusion :**

A travers la partie de l'étude conceptuelle, on a pu constater qu'une architecture à 3 niveaux été plus que nécessaire pour le projet et son bon fonctionnement et cela par les divers services qu'elles présente.

Mais la chose la plus importante qu'on a remarqué dans cette étape est sans aucun doute la diversité et l'abondance des informations récoltées pour la conception de la base de données.

Tous cela nous pousse à dire que cette conception a permis de décrire la solution à mettre en œuvre et spécifier de manière détaillée la structure générale du système d'information envisagé.

# Chapitre VI.

## Implémentation et réalisation.

### 1. Implémentation

L'implémentation consiste en premier lieu de mettre en œuvre tous les outils nécessaires pour le fonctionnement de notre système. Cela se fait en trois étapes :

- a) choix des outils : c'est une étape qui consiste à faire une étude comparative entre les différents outils dans le marché pour répondre aux questions suivantes :
  - Quels sont les avantages et les inconvénients de chacun ?
  - Qu'est ce que va nous apporter l'utilisation d'un outil par rapport aux autres ?
  - Et surtout la faisabilité.
- b) La préparation de l'environnement : cela se fait par l'installation des outils acquis, et les configurer pour qu'ils coopèrent entre eux.
- c) Le développement de l'applicatif.

#### 1.1. Choix de l'environnement de développement

##### 1.1.1. Côte serveur

- **Serveur de données**

La couche serveur de données contient les bases de données offrant une évolutivité suffisante pour répondre aux immenses besoins des applications Internet en matière de stockage, de manipulation, d'extraction et d'analyse de données.

Dans cette couche nous avons choisi Borland Interbase6 comme un système de gestion de base des données (SGBD), ce choix est basé sur les raisons suivantes :

- 1- Il est disponible sous une licence gratuite pour Windows.
- 2- Il est un SGBD basé sur une logique relationnelle.
- 3- Il respecte la norme SQL92.
- 4- Il possède une interface graphique ce qui facilite l'administration et le contrôle de la base des données.
- 5- Il supporte un grand nombre de transactions simultanées.



- **Serveur de présentation et d'applications**

Son rôle est de pouvoir recevoir les informations des clients, de les traiter, d'envoyer des réponses attendues, et de communiquer avec une base des données. Il contient deux parties

- **Couche présentation**

La couche présentation est constituée de serveurs Web qui traitent les requêtes HTTP et génèrent de manière dynamique le code de présentations pour exécution et affichage sur le client.

- **Couche applications**

Les serveurs d'applications abritent la logique métier qui, dans l'idéal, est réutilisable sur une variété de clients et d'applications. Il importe aussi que cette logique de gestion puisse être appelée aisément depuis des serveurs et des clients externes en vue de faciliter l'intégration de l'application. Ces serveurs d'applications contiennent des composants métier distribués réutilisables.

Parmi les serveurs disponibles sur le marché, Le serveur Tomcat4.0 a été choisi pour ces nombreux avantages :

- Il est indépendant des plates formes (multi plates forme)
- Il est disponible sous une licence gratuite.
- Il fait partie des logiciels libres (OPEN SOURCE).
- Il est dédié spécialement aux servlet et jsp.
- Assure un mécanisme de sécurité très élevé.
- Offre la configuration textuelle basée sur des fichiers XML.
- Il joue le double rôle d'un serveur Web et serveur Wap.

- **Langage de programmation :**

Le choix d'un langage de programmation est ainsi posé. JAVA, PHP, ASP et CGI sont donc permis aux candidats qui figurent dans notre application.

Java a le privilège d'être choisi notamment pour ses avantages incomparables par rapport aux autres. Parmi ces avantages :

- Simple : toutes les fonctionnalités pour implémenter des programmes complexes sans fonctions exotiques
- Orienté objet : tous les éléments dans Java sont des classes, des méthodes ou des objets. Quelques primitives et types de base ( while, for, int, ... )
- Une plateforme indépendante : un programme compilé génère du « byte code », celui-ci peut-être lu et interprété par n'importe quel système (Windows 95, Windows NT, Linux, ...)
- Performant : Java peut-être compilé avec un compilateur JIT (Just In Time), gain de temps avec des performances proche du C++
- Multi-tâches ou « Multi-Threaded » : Un programme peut avoir plusieurs processus qui s'exécutent en même temps.
- La portabilité : java possède sa propre machine virtuelle (JVM) qui le rend indépendant de l'environnement dont il s'exécute.

- Présente la base des Servlet et des JSP.

### 1.1.2. Côte clients

- **Clients web :**  
Au niveau client web, des postes légers basés sur un navigateur répondent à l'exigence d'accessibilité à tout moment et en tout lieu de l'utilisateur final se servant du Web en libre-service.
- **Clients wap :**  
Au niveau clients Wap, il suffit d'avoir un mobile possédant un navigateur simple, et un point d'accès à Internet pré-configuré par l'opérateur téléphonique.

### 2. Réalisation :

La partie réalisation consiste à développer une interface graphique pour les différents utilisateurs. Ces interfaces offrent à chaque utilisateur un certain nombre de fonctionnalités permettant de mettre à jour et consulter facilement les données à distance. Les étapes de réalisation sont :

- **La définition des utilisateurs :** les personnes qui agissent sur le système sont décomposés en groupes d'utilisateurs, chaque groupe contient un ou plusieurs utilisateurs qui partagent une vue sur le système global.
- **Définition et développement des lots :** pour des raisons de bien partager le travail et d'éviter un chevauchement des tâches, on a préféré décomposer le système global en parties plus petites et indépendantes des autres. Pour cela, on a considéré chaque fonctionnalité du système comme un lot (tâche).  
Le développement de ces tâches se fait d'une façon indépendante et en fonction des spécifications détaillées de chacune.
- **Regroupement des lots et tests d'intégrité :** cette étape consiste à regrouper l'ensemble des lots développés et les effectuer aux groupes concernés en respectant le profil de chacun. C'est la notion des portails internet.  
A fin de garantir le bon fonctionnement du système, il est indispensable de faire un test d'intégrité sur l'ensemble global des lots.

### 3. Sécurité de l'applicatif :

La sécurité des systèmes informatiques se cantonne généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leur ont été octroyés.

La sécurité informatique doit toutefois être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité qui assure:

- L'intégrité, c'est-à-dire garantir que les données sont bien celles qu'on croit être
- La confidentialité, consistant à assurer que seules les personnes autorisées ont accès aux ressources
- La disponibilité, permettant de maintenir le bon fonctionnement du système informatique
- La non répudiation, permettant de garantir qu'une transaction ne peut être niée

Notre système possède deux niveaux de sécurité. le premier concerne l'accès aux ressources, le deuxième est dédié au privilège approprié à l'utilisateur.

#### 4. Présentation de l'application :

##### 4.1. Partie I : Wap

La page d'accueil de l'étudiant comporte les points suivants :

- Identification :
- Affectation :
- Notes

Cette page commence avec une image qui reste 3 secondes suivie par le menu principal

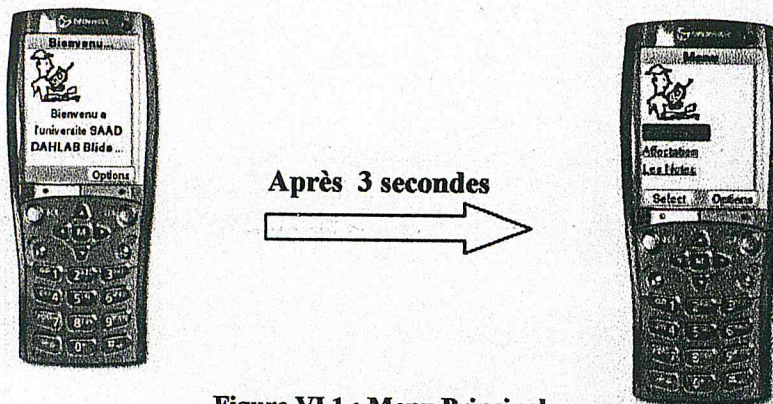


Figure VI.1 : Menu Principal

##### 4.1.1. Identification :

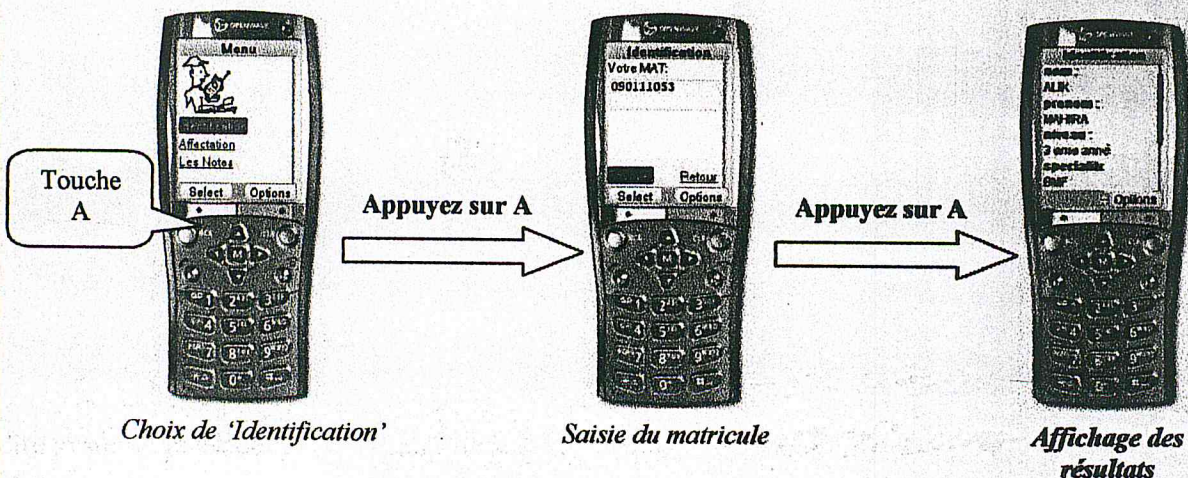


Figure VI.2 : les étapes de l'identification

4.1.2. Affectation :

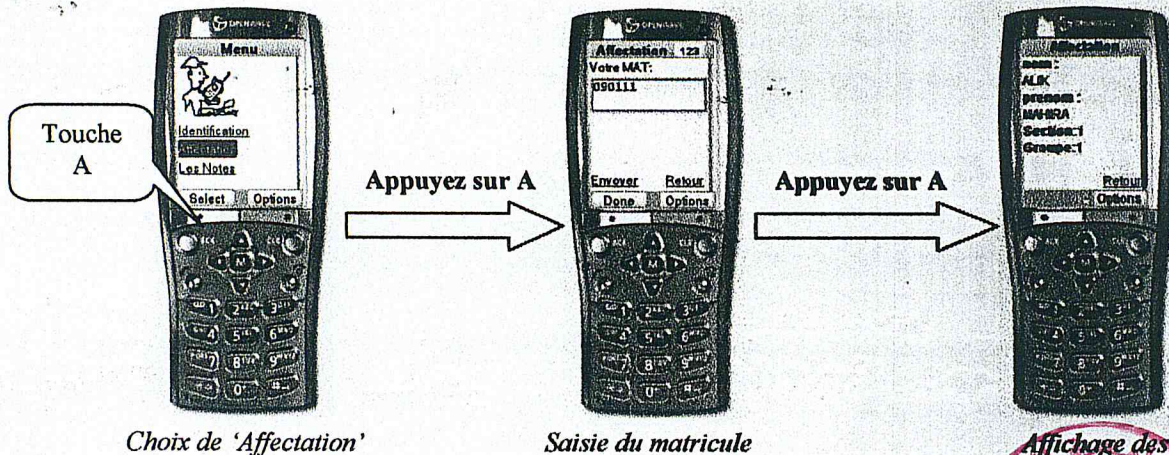


Figure VI.3 : les étapes de consultation de l'affectation



4.1.3. Notes :

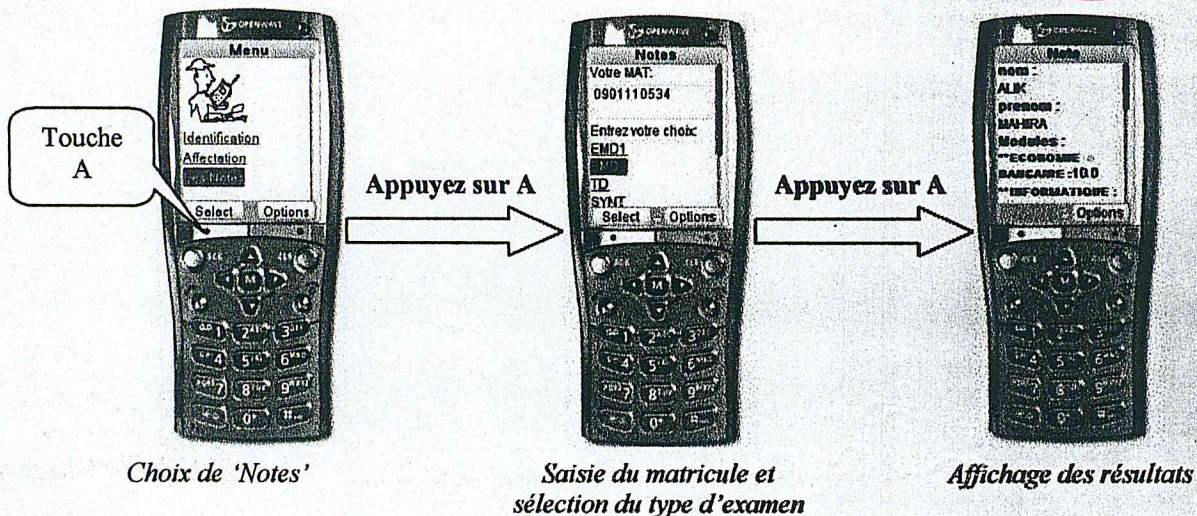


Figure VI.4 : les étapes de consultation des notes

## 4.2.Partie II : Web

L'application réalisée est un portail Internet (ou intranet). Elle offre un point d'accès commun pour tous les internautes et un mécanisme d'authentification permettant d'orienter chaque utilisateur vers son propre espace dans le site.

Une page web intitulée « le site officiel de la faculté des sciences économiques et de gestion » est téléchargée après appel à l'URL suivant : <http://serveur/Economie/index.htm>.

Lorsque l'utilisateur demande cette adresse, la page d'accueil suivante est affichée.

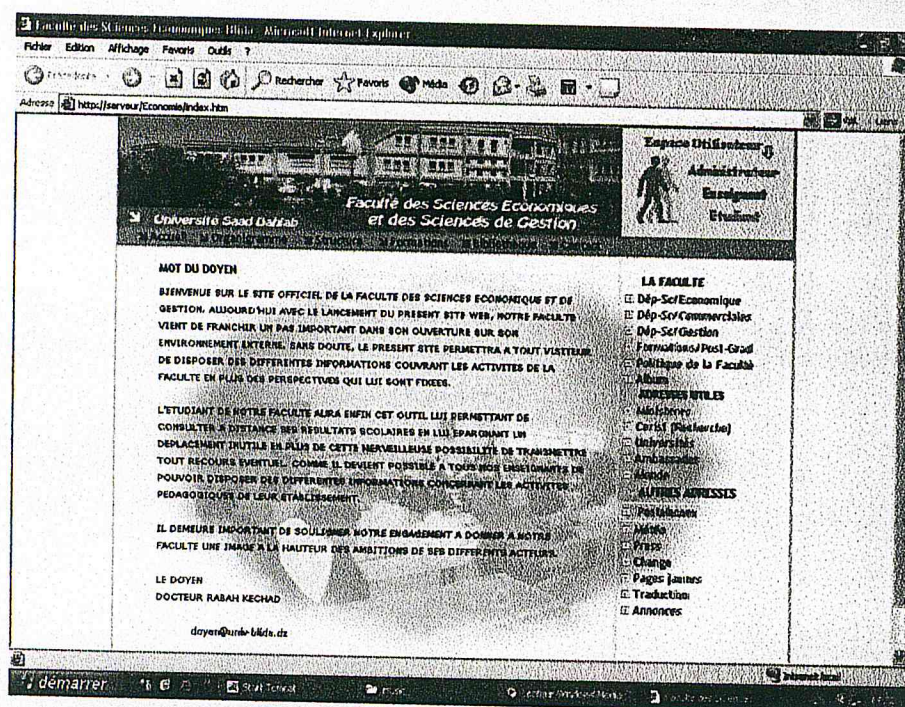


Figure VI.5: Présentation de la page d'accueil

Cette page est constituée de :

- Une partie présentative de la faculté à accès libre (sans authentification). Elle propose certaines informations qui peuvent être utiles pour les internautes.
- Un espace utilisateurs à accès contrôlé, leur permettant de se connecter à leurs propre espace qui contient trois types d'utilisateurs :
  1. Etudiant
  2. Enseignant
  3. Administrateur.

**4.2.1. L'espace 'Etudiant' :** cet espace est destiné à tous les étudiants da la faculté. Pour y accéder, l'étudiant doit faire la saisie de son matricule.

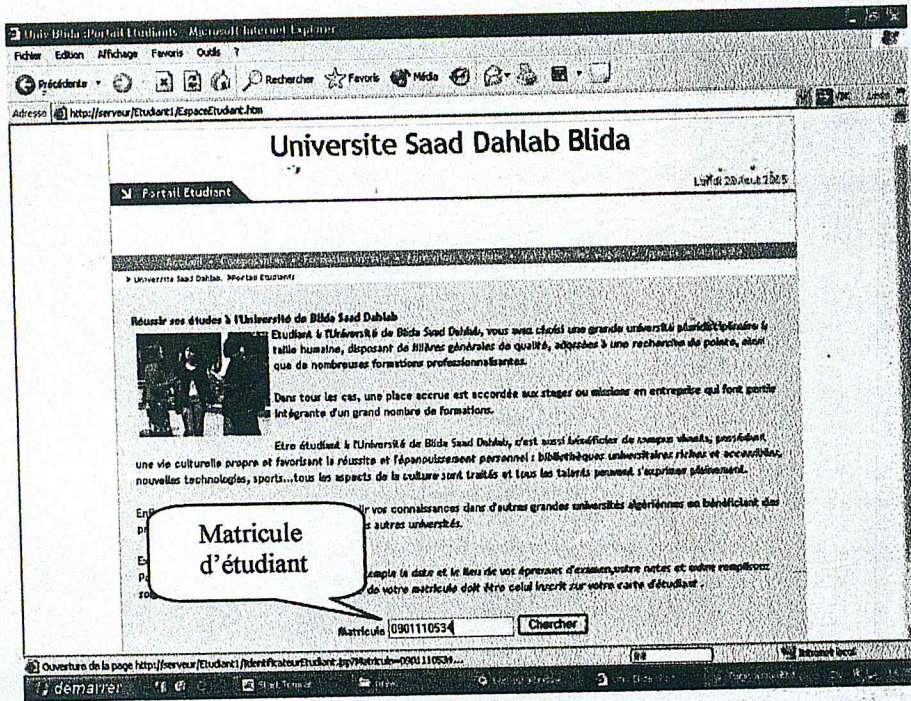


Figure VL6: Page identification d'étudiant

Après la vérification du matricule, une deuxième fenêtre sera affichée.

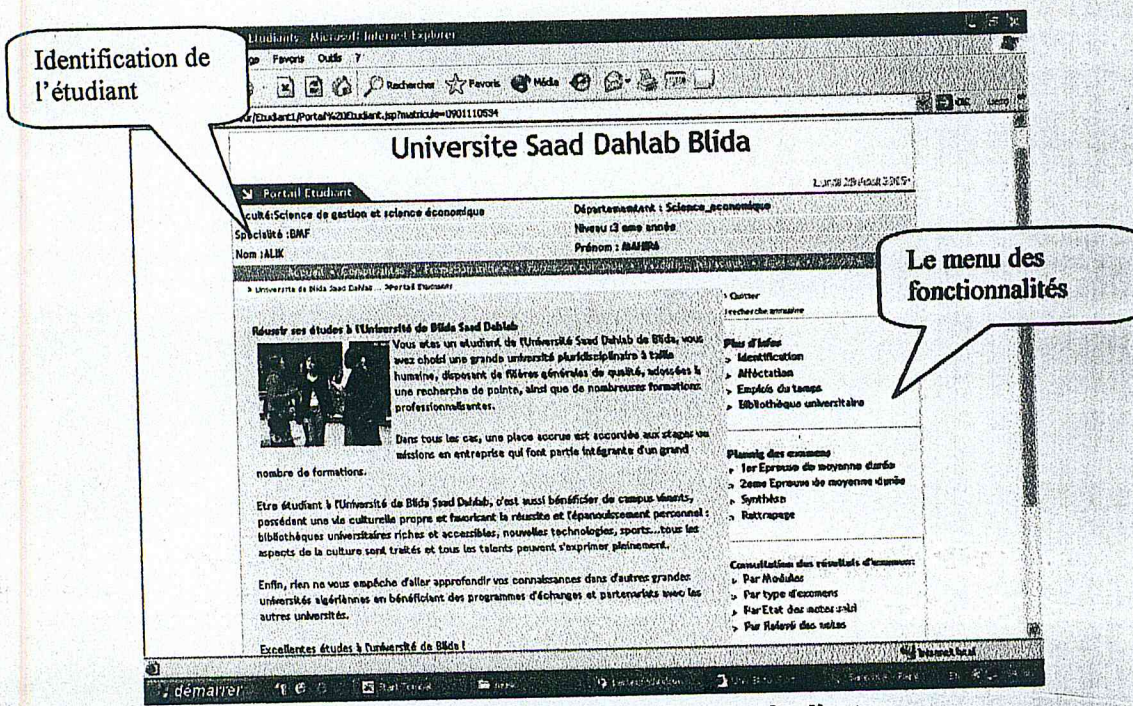
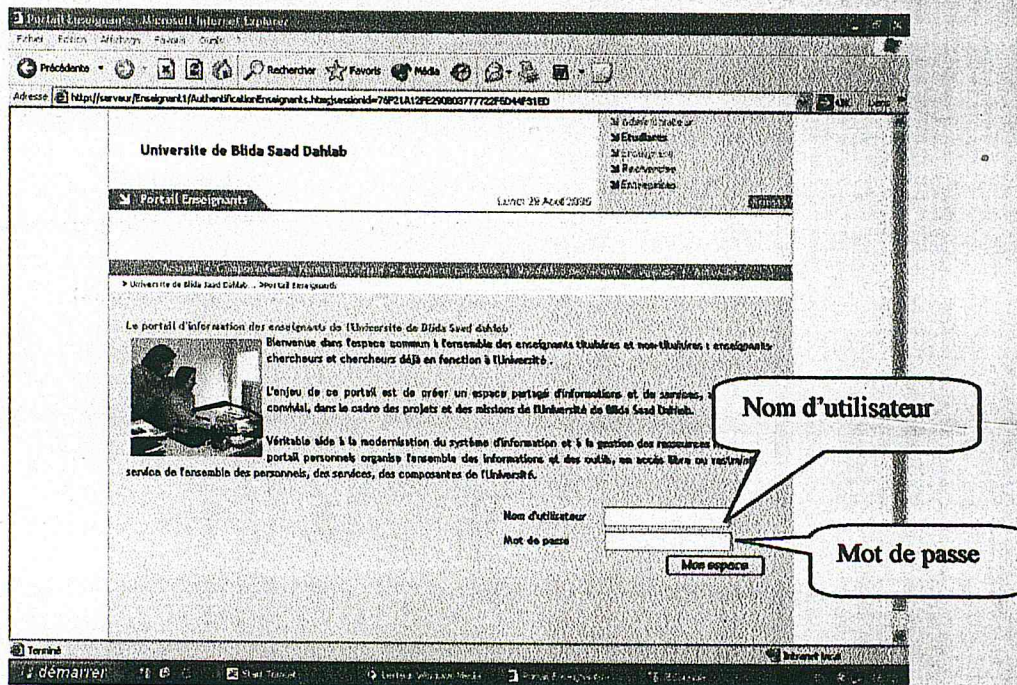


Figure VL7: Présentation de l'espace étudiant.

- Dans la barre 'Identification de l'étudiant' on voit :
  - La faculté,
  - Le département,
  - Spécialité,

- Niveau,
  - Nom
  - Et Prénom de l'étudiant
- Le menu des fonctionnalités contient les points suivants :
    - Identification : Afficher les informations civiles de l'étudiant.
    - Affectation : section et groupe (figure 1 Annexe B).
    - Emplois du temps (figure 2 Annexe B).
    - Consultation des notes
      - Par module (figure 4 Annexe B).
      - Par type d'examen (figure 5 Annexe B).
      - Par état des notes saisies (figure 3 Annexe B).
      - Par relevé des notes (figure 6 Annexe B).
    - Envois des réclamations (figure 7 Annexe B).
    - Boite de réception des messages (figure 8 Annexe B).

**4.2.2. L'espace 'Enseignant' :** cet espace est destiné à tous les enseignants de la faculté. Pour y accéder, l'enseignant doit faire la saisie de son nom et de son mot de passe.



**Figure VL8: Page identification d'enseignant.**

Après la vérification du « Nom d'utilisateur » et de « Mot de passe », une deuxième fenêtre sera affichée.

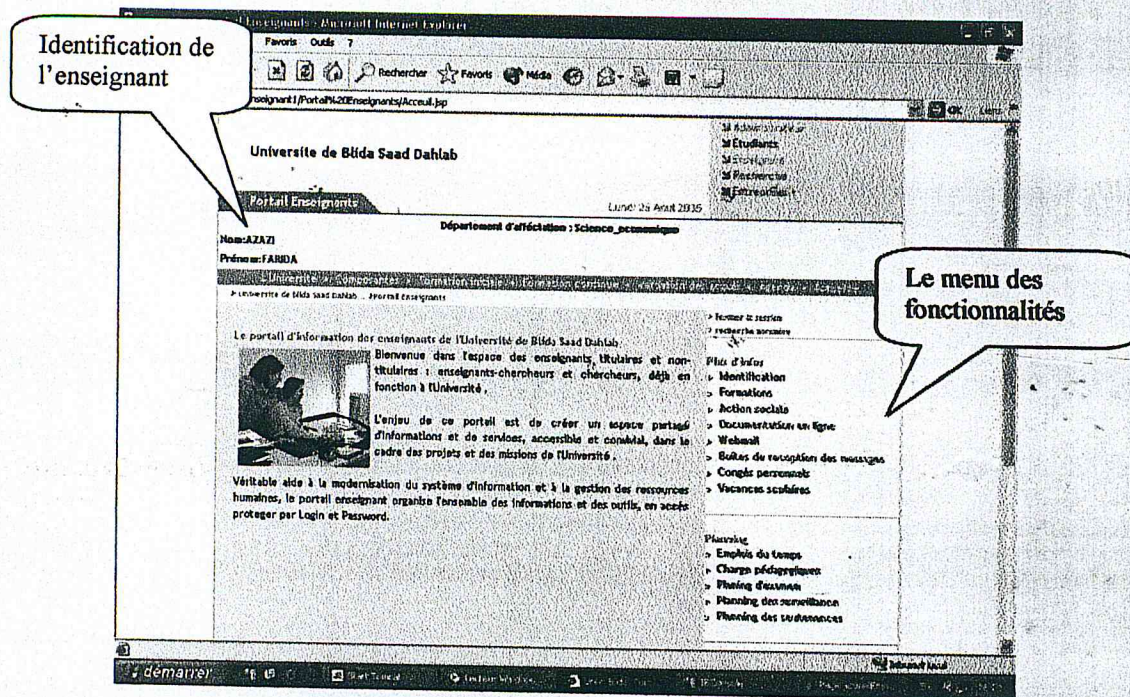


Figure VI.9: Présentation de l'espace enseignant.

- Dans la barre 'Identification de l'enseignant' on voit :
  - Le département d'affectation,
  - Nom
  - Et Prénom de l'enseignant.
- Le menu des fonctionnalités contient les points suivants :
  - Identification : Afficher les informations civiles de l'enseignant.
  - La consultation des charges pédagogiques (figure 10 Annexe B).
  - Emplois du temps (figure 11 Annexe B).
  - Saisie des notes (figure 12 Annexe B).
  - PV modulaire (figure 14 Annexe B).
  - Edition des réclamations.
  - Edition des listes des étudiants modulaire et par type d'examen (figure 15 Annexe B).
  - Traitement des réclamations.
  - Boite de réception des messages.
  - Possibilité de changer le mot de passe.

**4.2.3. L'espace 'Administrateurs' :** cet espace regroupe quatre groupes d'utilisateurs. Son accès se fait par la saisie du nom de l'utilisateur et son mot de passe.

Les quatre groupes d'utilisateurs sont :

- Doyen de la faculté.
- Chefs des départements.
- Agents de service de la scolarité centrale.
- Agents de service d'enseignement central.



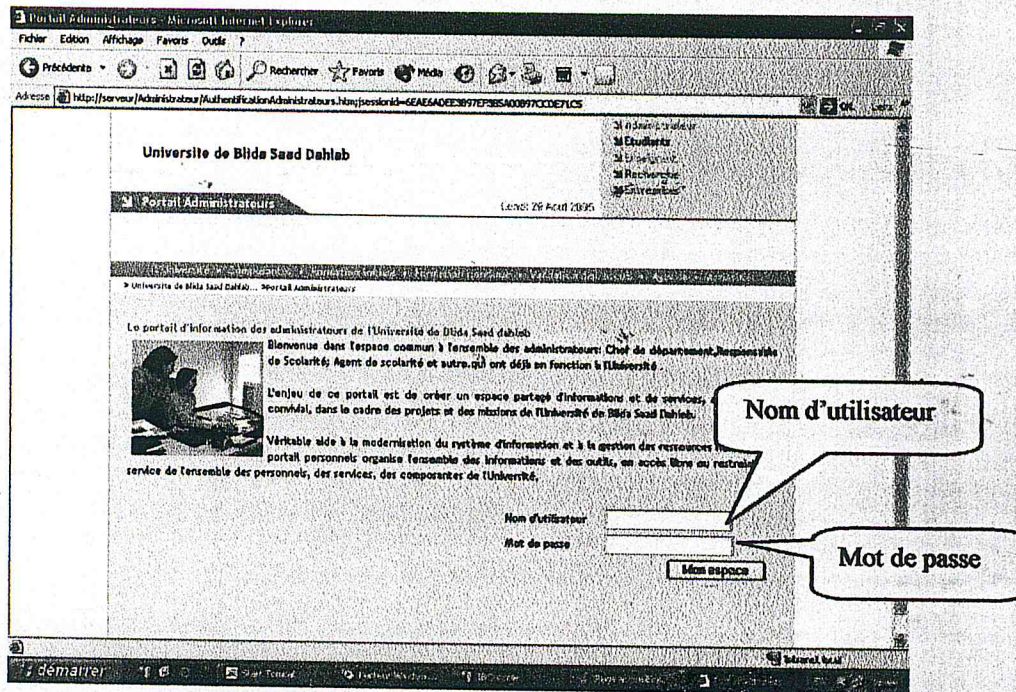


Figure VI.10: Page identification des administrateurs.

Après la vérification du « Nom de l'utilisateur » et de « Mot de passe » le mécanisme d'authentification offre à l'utilisateur son propre espace et une deuxième fenêtre sera affichée.

1- Doyen de la faculté.

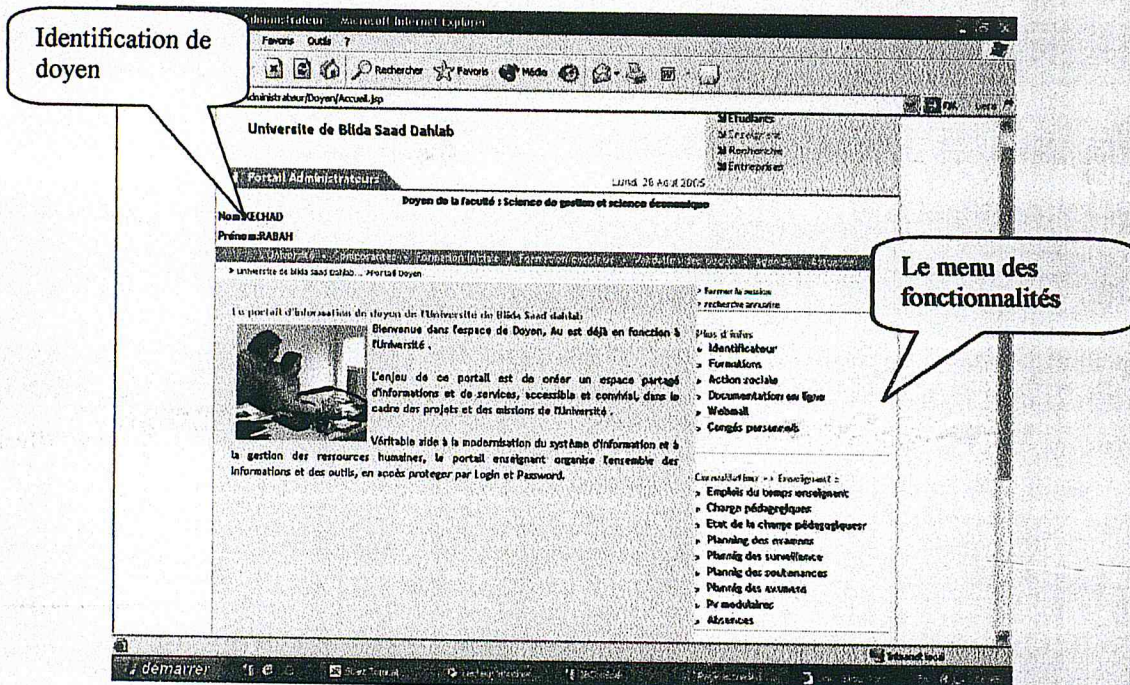


Figure VI.11: Présentation de l'espace doyen.

- Dans la barre 'Identification de l'enseignant' on voit :
  - La faculté.
  - Nom
  - Et Prénom de doyen.
- Le menu des fonctionnalités contient les points suivants :
  - Identification : Afficher les informations civiles de doyen.
  - Les consultations cotées enseignantes
    - des emplois du temps.
    - charges pédagogiques.
    - états charges pédagogiques.
  - Les consultations cotées étudiants.
    - des emplois du temps par sections et par groupes.
    - Listes administratives des étudiants.
    - Relevé des notes.
    - Attestation de succès et certificat scolarité.
  - Possibilité de changer le mot de passe.

2- chefs des départements :

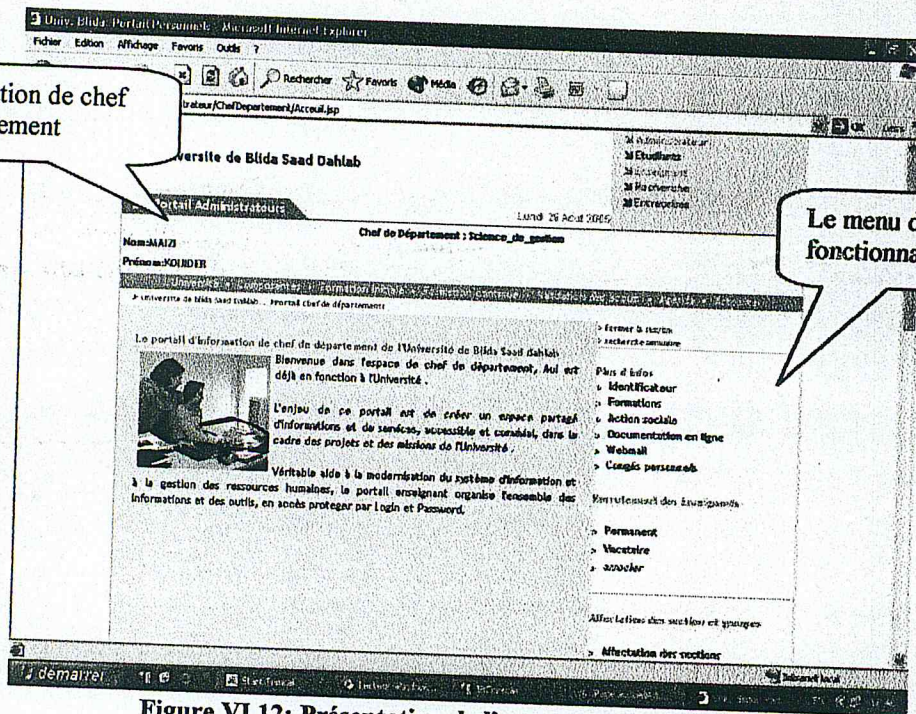


Figure VL12: Présentation de l'espace chef département.

- Dans la barre 'Identification de l'enseignant' on voit :
  - Le département.
  - Nom
  - Et Prénom de chef de département.
- Le menu des fonctionnalités contient les points suivants :
  - Identification : Afficher les informations civiles de chef département.

- Gestion des charges pédagogique (affectation, mise à jour).
- Etablissement d'emplois du temps (figure 19 Annexe B).
- Consultation des PV modulaire.
- Les consultations cotées enseignantes
  - des emplois du temps.
  - charges pédagogiques.
  - états charges pédagogiques.
- Les consultations cotées étudiants.
  - des emplois du temps par sections et par groupes.
  - Listes administratives des étudiants
  - Liste des étudiants par module.
  - Relevé des notes.
  - Attestation de succès et certificat scolarité.
- Délibération par sections et par groupes (figure 17 Annexe B).
- Des Statistique (figure 18 Annexe B).
  - nombre d'étudiants par module, session.
  - Rapport enseignants/ étudiants.
- Possibilité de changer le mot de passe.

### 3-Agent de service de scolarité central :

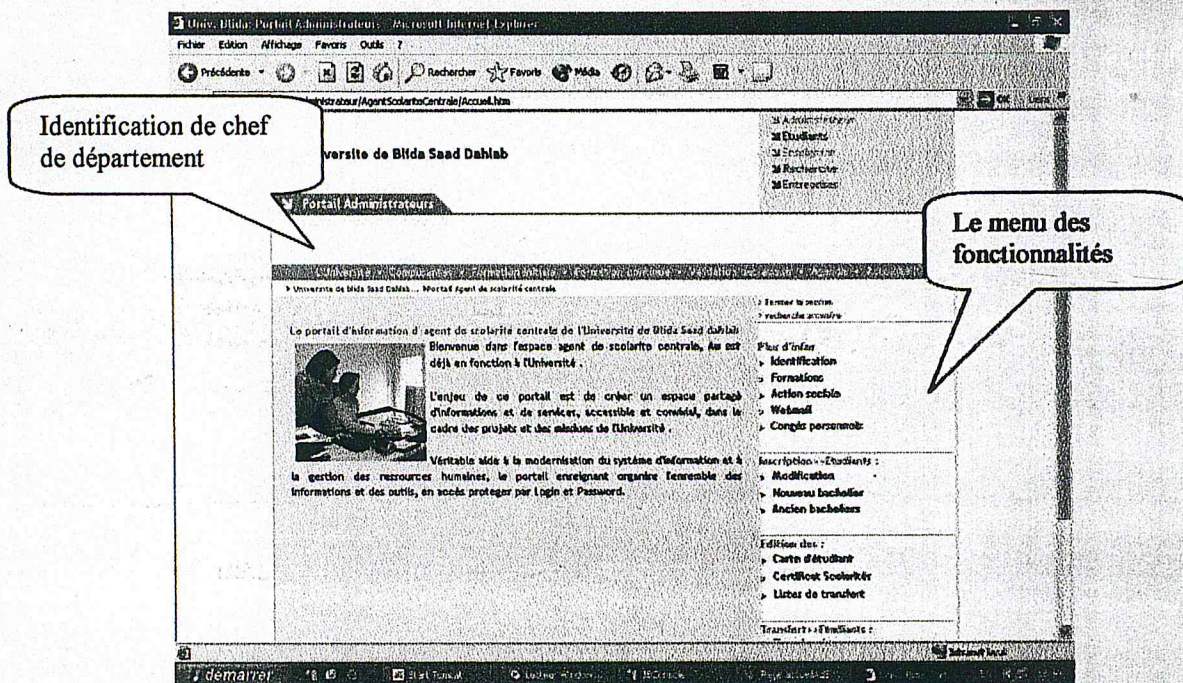


Figure VI.13: Présentation de l'espace Agent de scolarité.

- Dans la barre 'Identification de l'enseignant' on voit :
  - Le service scolarité.
  - Nom
  - Et Prénom de l'agent.
- Le menu des fonctionnalités contient les points suivants :
  - Identification : Afficher les informations civiles de l'agent.

- Inscription des étudiants (figure 20 Annexe B).
- Edition des :
  - Carte étudiant.
  - Certificat scolarité.
  - Liste des transferts.
- Transfert des étudiants (figure 21 Annexe B).
  - Interne
  - Externe.
  - départ.
- Edition des :
  - Relevé des notes (figure 22 Annexe B).
  - Attestation de succès et certificat scolarité (figure 23 Annexe B).
  - Certificats de scolarité (figure 24 Annexe B).
- Possibilité de changer le mot de passe.

### 5. Agent de service d'enseignement central :

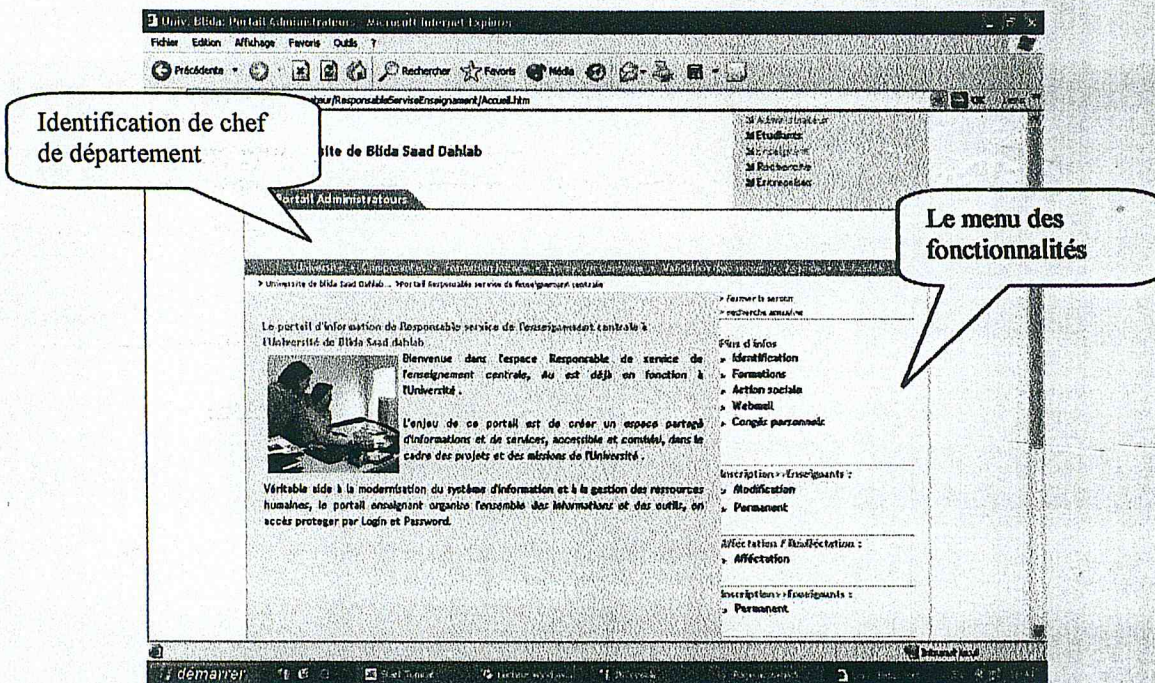


Figure VI.14: Présentation de l'espace Agent de service d'enseignement.

- Dans la barre 'Identification de l'enseignant' on voit :
  - Le service d'enseignement.
  - Nom
  - Et Prénom de l'agent.
- Le menu des fonctionnalités contient les points suivants :
  - Identification : Afficher les informations civiles de l'agent.
  - Recrutement des enseignants (figure 25 Annexe B).
  - Affectation et réaffectation des enseignants.

- Les consultations cotées enseignantes
  - Emplois du temps.
  - charges pédagogiques.
  - Affectation d'un enseignant
  - états charges pédagogiques.
  - Etats des charges par département
  - Etats des charges globales.
- Edition des listes des enseignants.
  - Permanent
  - Vacataire
  - associe.
- Possibilité de changer le mot de passe.

### **Conclusion générale :**

L'objectif primordial du présent travail consiste à la conception et la réalisation d'un portail Internet pour la gestion d'enseignement. La motivation qui sous-tend ce travail réside principalement dans le fait que la technologie Web offrent un souplesse dans le coté de l'intégration des données et une meilleur performance dans le cadre de la recherche de l'information.

Au cours de notre projet nous avons développé un système d'information dans le but capitaliser les informations, ce système d'information va permettre un accès rapide et permanant à l'information.

Sur un plan personnel se projet nous à permet de :

- Etudier les différent composant de la technologie Web et Wap.
- Toucher le domaine d'administration des bases de données.
- Avoir une expérience très utile dans la conception des systèmes d'information.

Ce projet ne représente qu'une partie d'une grande vision pour informatisation de l'université, les grands titres qui représentent le plan d'action global est le suivant :

#### **1-développement des applications pour les différents services et facultés de l'université.**

- Pedagogie : Facultés, V/R pédagogie, V/R PGRS
- Gestion des locaux : facultés, V/R planification
- Gestion Bibliothèque : Département, Facultés, Centrale
- Gestion des projets de recherche : PGRS facultés, V/R PGRS
- Gestion de Personnel : facultés, service central
- Gestion de la DFC

#### **2- Mise en Oeuvre des services INTRANET :**

- Serveur de messagerie
- Moteur de recherche
- Serveur Web & Site de l'université

#### **3- Migration vers le système Linux et les applications OPEN SOURCE**

# Bibliographie.

Nos meilleures idées viennent des autres.  
*Ralph Waldo Emerson*

- [01] : Wap : « Wireless Application Protocol » Alain Deseine - 1999.
- [02] : Alex Homer, Darren Gill, Stephen Jakab « Interfacé entre Web et base de données » Eyrolles 1998.
- [03] : Olivier Picard, Jean Michel Allemand « Le Wap :livre blanc » Alladin technologies 2000.
- [04] : Laurent Letourmy, Tomas Papiernik, Alain Hélaïli, Xavier Martzel « Construction d'une application Wap » Eyrolles 2000.
- [05] : Trad. Valéry Frémaux, Joelle Cornavin,... « Le Guide officiel Wap1.2 » Eyrolles 2001.
- [06] : comment ça marche « [www.commentcamarche.fr](http://www.commentcamarche.fr). »
- [07] : Guill.net -Les protocoles de communication-« [www.guill.net](http://www.guill.net). ».
- [08] : Nokia mobile phone, Nokia Wap Toolkit3.0 user's guide « [www.forum.nokia.com](http://www.forum.nokia.com) ».
- [09] : Tous sur le Wap «[www.wapforum.com](http://www.wapforum.com) ».
- [10] : Van Lancker Luck ,WAP@WML « [www.ccim.be/ccim328/wap](http://www.ccim.be/ccim328/wap) ».
- [11] : Georges Gardarin, Olivier Gardarin « Le client-Serveur » 1999.
- [12] : Programmer en Java CPR informatique - septembre 2002.
- [13] : Eric Giguere « Java 2 Micro Edition: Professional Developer's Guide » 2000.
- [14] : Guillaume Heilles « Core Sevelets and Java Server Page » CampusPress, 2000.
- [15] : HUNTER Jason, CRAWFOERD William «Servlets Java» O'reilly, 1998.
- [16] : Qusay Mahmoud « Learning Wireless Java » 2001.
- [17] : Lan Sommerville « Le génie logiciel et ses applications » paris, 1988.
- [18] : Le site officiel de Sun pour le développement java « <http://java.sun.com> »

## Bibliographie

---

- [19] : James White, David Hemphill « Java 2 Micro Edition » 2002.
- [20] : Damien Zéni « Java 2 Micro Edition Présentation personnelle » Février 2003.
- [21] : Jean Boileau, Thierry Thureaux « Client/Serveur à trois niveaux » Vuibert Informatique, France 2000.
- [22] : Georges Gardarin « Base de données » Eyrolles , France 2002.
- [23] : Pierre Vaudart « Les langages de scripts pour les sites Web dynamiques » Vision, Canada 2001.
- [24] : Paul Kahn, Krzysztof Lenk « Architecture des site web » Pyramid, Belgique 2001.
- [25] : Sebastien Ramtola, Eric Lemont « Technologie à multiniveaux » Aprit,2002.
- [26] : Vincent Doubier « PHP, la référence des pages dynamique » Leclerc 2003.
- [27] : Le site officiel de Apache « <http://www.apache.org> ».
- [28] : Le site officiel de Borland « [www.borland.com](http://www.borland.com) ».
- [29] : Le guide français de Tomcat « <http://www-igm.univ-mlv.fr/~dr/XPOSE2003/tomcat/tomcat.php> ».
- [30] : Julien Gilli « Les Servlets » janvier 2005.
- [31] : Richard Grin « Langage SQL » novembre 2000.
- [32] : Valère Viandier « Introduction à JDBC » 2001.
- [33] : Philippe Mathieu « Bases de Données (de Merise à JDBC) » Université des sciences et technologie de LILE
- [34] : Christophe Thiry «Les composants des architectures Web» 2001.
- [35] : Christophe Jollivet «Simplifiez vous JDBC avec Jakarta Commons» janvier 2005.
- [36] : Etienne Bar «Introduction aux Procédures Stockées d'Interbase».
- [37] : Christophe Jollivet «Gestion d'un pool de connexions SGBD par Tomcat »décembre 2004.
- [38] : Daniel K. Schneider - Vivian Synteta «Introduction à JSP »juin 2000.
- [39] : Didier Girard «OpenSourceJava – Deuxième partie : servlet/JSP/MVC»article Mars 2001.
- [40] :George Reese, Trad Hervé Soulard «JDBC et JAVA Guide du programmeur »O'Reilly.



- [41] :Cay S.Horstmann, Gary Cornell«Au Cœur de JAVA 2 Notions Fondamentales » CampusPress.
- [42] :Pierre-Yves Cloux, David Doussot, Aurélien Géron «Les architectures client-serveur Internet et Intranet »Dunod.
- [43] : Stéphane Bordage, Franklin Brousse, David Thévenon «Conduite de projet Web » Eyrolles.
- [44] :Alain Lefebvre «Web client-serveur le triomphe du client léger » Eyrolles.
- [45] :Adnane Benjelloun, Luis Fortier, Martin Rioux «Etat de l'art des logiciels libres » CRIM 2003.
- [46] :Frédéric Créplet «Ingénierie de projet intranet » Edition d'organisation 2003.
- [47] «Borland Interbase 4.0 »Borland.
- [48] «Règlement intérieur de l'université ».
- [49] : OpenWave SDK Toolkit user's guide « [www.openwave.com](http://www.openwave.com) ».
- [50]: Jean Patrick Matheron «Comprendre Merise : Outils conceptuels et organisationnels».

