

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT INFORMATIQUE

Mémoire

*En vue de l'obtention du diplôme de master II en
informatique Option : Conduite de projet informatique*

Thème

***Conception et réalisation d'une base de données répartie pour
un campus numérique***

***Cas « faculté de Génie Electrique et d'informatique de
l'UMMTO »***

Proposé et dirigé par :

M^{ME} BOUARAB-DAHMANI.F

Réalisé par :

M^{lle} GHEMOURI OUAHIBA

Promotion : 2012/2013

Remerciements

Je tiens à remercier ma promotrice M^{me} BOURAB.F, pour m'avoir proposé ce sujet, pour la qualité de son encadrement, et son suivi durant toute la durée de projet.

Je remercie vivement les membres du jury pour avoir aimablement accepté de juger mon travail.

A tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.

Dédicaces

*À mes très chers parents ,qui m'ont soutenu tout au
long de mes études
À mes frères & sœurs
À tous mes amis (es)*

Ouahiba

Sommaire

Introduction générale	1
<i>Chapitre I : Les espaces numériques de travail (ENT)</i>	
Introduction	3
I.1 Définition	3
I.2 Terminologies	3
I.3 Les types de campus numériques	4
➤ Les campus de « formation ouverte à distance » (ou FOAD)	4
➤ Les campus « environnements numériques de travail »	4
I.3 Caractéristiques.....	4
I.4 Objectifs	4
I.5 Avantages	5
I.6 Les acteurs	5
I.7 Les services offerts par les ENT	6
I.7.1 Les services de bases	6
I.7.2 Les services applicatifs	7
I.7.3 les services socle	7
I.8 Architecture générale	7
I.9 Architecture technique du socle de l'ENT	9
I.10 Les moyens à mettre en œuvre	10
I.10.1 Moyens techniques	10
I.10.2 Moyens humains	10
I.10.3 Moyens juridique	10
I.10.4 Moyens financiers	10
I.11 Les Recommandation officielle pour les ENT	10
➤ S3IT	11
➤ SDET	11
➤ SDI	11
➤ S2i2e.....	11

I.11 Exemple de projets compatibles ENT SDET pour l'université	11
Conclusion	12

Chapitre II : Les bases de données réparties

Introduction	13
II.1 Notion générales	13
II.1.1 Définition d'une base de données	13
II.1.2 Système de gestion de base de données	14
II.1.3 Architecture d'un SGBD	14
II.2 les bases de données réparties	15
II.2.1 Définition.....	15
II.2.2 Buts de la répartition des bases de données	15
II.2.3 Objectifs des bases de données répartis.....	15
➤ Autonomie locale	15
➤ Egalité entre sites	16
➤ Transparence de localisation	16
➤ transparence de partitionnement	16
➤ Fragmentation transparente	16
➤ Indépendance vis-à-vis du SE	16
II.3 Conception d'une base de données répartie	16
II.3.1 Conception descendante	16
II.3.2 Conception ascendante	17
II.4 La fragmentation	17
II.4.1 Définition	17
II.4.2 Types de fragmentation	17
a) La fragmentation horizontale	17
b) La fragmentation verticale	18
c) Fragmentation hybride (fragmentation mixte)	19
II.4.3 L'allocation des fragments	20
II.4.4 Le schéma de répartition.....	20

II.5 Architecture de schéma d'une base de données distribuée	20
II.6 Les systèmes fédérés	21
II.7 Catégories des systèmes fédérés	21
II.7.1 Les fédérations faiblement couplées	21
II.7.2 Les fédérations fortement couplées	22
II.7.3 Les bases de données fédérées	24
II.8 Les SGBD fédérés	24
II.8.1 Définition de DB2 d'IBM	25
II.8.2 Les avantages et les inconvénients du DB2 d'IBM.....	25
a) Les avantages	25
b) Les inconvénients	25
II.8.3 Les composants d'un système fédéré avec DB2	26
✓ Les sources de données	26
✓ Le catalogue système	29
✓ Le serveur fédéré	29
✓ Encapsuleurs et modules d'encapsuleur	29
✓ Les tâches effectuées par l'encapsuleur	29
II.9 Caractéristiques de la solution fédérée d'IBM	30
• Transparence.....	30
• Hétérogénéité	30
• Extensibilité et ouverture de la fédération	30
• Extensibilité et ouverture de la fédération	30
II.10 SGBD Open source MySQL	31
II.10.1 Présentation de MySQL.....	31
II.10.2 Avantages et inconvénients de MySQL	31
✓ Avantages	31
✓ Inconvénients	31
II.10.3 MySQL et la répartition des bases de données	31
1) Partitionnement par intervalles.....	32
2) Partitionnement par listes	32

3) Partitionnement par hachage	32
4) Partitionnement par clé.....	32
Conclusion	33

Chapitre III : Intégration de sources de données

Introduction	34
III.1 Les vues classiques	34
III.2 Création d'une vue	35
III.3 Les vues matérialisées	36
III.3.1 Manipulation des vues matérialisées	36
III.3.1.1 Oracle	36
• Mise à jour des vues matérialisées sous Oracle	37
• Rafraîchissement complet	37
• Rafraîchissement incrémental (rapide)	37
• Rafraîchissement forcé	37
III.3.1.2 MySQL	37
➤ Mise à jour des vues matérialisées sous MySQL	38
• Une mise à jour sur demande	38
• Une mise à jour automatique	38
III.3.2 Avantages des vues matérialisées	39
III.4 Système d'intégration de données	39
III.4.1 Classification des systèmes d'intégration de données	40
III.4.1.1 Technique du médiateur	40
➤ Le médiateur	41
➤ L'adaptateur	41
III.4.1.2 Technique d'entrepôt	41
III.5 Technique implémentée dans notre application	42
III.5.1 Les utilisateurs	43
III.5.2 Les bases de données.....	44

III.5.3 Exemple de cas de notre application	44
Conclusion	45

Chapitre IV : Analyse et conception

Introduction	46
IV.1 Proposition organisationnelle	46
IV.1.1 Proposition organisationnelle de la plateforme du campus	46
IV.1.2 Proposition organisationnelle des données	50
➤ Localisation des données	50
IV.2Analyse.....	53
IV.2.1 Spécification des besoins	53
IV.2.2 Identification des acteurs	54
IV.2.2.1 Identification des acteurs de département informatique	54
IV.2.2.2 Identification des acteurs du service budget et comptabilité	55
IV.2.2.3 Administrateurs	55
IV.3 Diagramme de contexte	55
IV.4 Identification et représentation des cas d'utilisation.....	56
IV.4.1 Les diagrammes de cas d'utilisation	59
➤ La relation d'inclusion « include ».....	59
➤ La relation d'exclusion « extend »	59
IV.5Conception	66
IV.5.1Diagrammes de séquence	66
➤ Les objets de type interface	66
➤ Les objets de type entités	66
➤ Les objets de type contrôle	67
IV.5.2 Présentation de quelques diagrammes de séquence correspondant aux cas d'utilisation déjà décrits	57
IV.5.2.1.Cas d'utilisation « S'inscrire à l'ENT »	68
IV.5.2.2Cas d'utilisation « s'authentifier »	69

IV.5.2.3Cas d'utilisation « Modifier profil »	70
IV.5.2.4Cas d'utilisation « Envoyer un message à un étudiant ou enseignant ou administration »	73
IV.5.2.5Cas d'utilisation « Remplir un état de vacation»	74
IV.5.2.6Cas d'utilisation « Modifier le mode de paiement (par banque ou CCP) »	76
IV.5.2.7Cas d'utilisation « Modifier le numéro de compte »	77
IV.5.2.8Cas d'utilisation « Valider inscription enseignant »	79
IV.5.2.9Cas d'utilisation « Consulter messagerie»	81
IV.5.2.10Cas d'utilisation « Valider un état de vacation»	82
IV.5.2.11Cas d'utilisation «Etablir une dépense»	84
IV.5.2.12Cas d'utilisation «Mandater une dépense»	86
IV.6Diagramme de classe :	87
IV.6.1Diagramme du schéma logique du site département	88
IV.6.2Diagramme du schéma logique du site service budget	89
IV.6.3 Diagramme du schéma logique de la base faculté	90
IV.6.4Les vues matérialisées créer.....	91
IV.7 Le modèle logique	92
IV.7.1Modèle logique du site département	93
IV.7.2Modèle logique du site service budget	93
IV.7.3Modèle logique du site faculté	94
IV.8Les Bases de données	95
IV.8.1Base de données de département	95
IV.8.2Base de données du service budget et comptabilité	99
IV.8.3Base de données faculté	101
Conclusion	104

Chapitre V : Réalisation

Introduction	105
V.1 Architecture de notre application	105
V.2 Environnement de développement et d'implémentation	105
V.2.1 Les langages de programmation	105
➤ HTML	106
➤ CSS	106
➤ Langage de requête SQL	106
➤ JavaScript	106
➤ PHP	107
V.2.2 Serveur web Apache	109
V.2.3 Serveur MySQL	109
V.2.4 PHPMyAdmin	109
V.2.4.1 Implémentation des bases de données	110
V.2.5 Les logiciels	110
V.2.5.1 WAMP (Windows, Apache, MySQL, PHP)	110
V.2.5.2 Macromedia Dreamweaver 8	111
V.3 présentation de quelques interfaces de notre application	112
Conclusion	128
Conclusion générale	129
Références bibliographiques	130

Introduction générale

Introduction Générale

Le monde de l'informatique évolue très rapidement, alors que son but initial, était d'offrir des services satisfaisants, du point de vue vitesse d'exécution des tâches et obtention de statistiques plus précises. Actuellement, de nouveaux besoins sont apparus, toute organisation automatisée souhaite stocker et échanger ses informations qui sont géographiquement éloignées, ce qui rend la tâche de la collecte et de traitement d'une grande quantité d'informations dispersées très délicate, de ce fait, l'amélioration des systèmes d'informations est devenue une priorité pour les gérants des entreprises.

La solution qui s'impose est de distribuer les données et les organiser dans des bases de données sur différents sites de stockage. L'ensemble de ces sites constitue un système de bases de données réparties offrant la possibilité aux utilisateurs de manipuler les différentes bases via un réseau de manière transparente, comme dans une base de données globale.

Jusqu'ici, les projets proposés à la réalisation d'un prototype de campus numérique pour l'université Mouloud Mammeri de Tizi-Ouzou (UMMTO) se basent sur la construction d'une base de données centralisée, or que ces solutions ne sont pas efficaces et optimales, et elles ne peuvent pas répondre à tous les besoins de ces usagers, vu la quantité d'informations utilisée et manipulée par l'université, et aussi la localisation des ces différentes facultés qui sont géographiquement éloignées, donc techniquement une base de données centralisée pose des inconvénients majeurs pour son administration.

Notre projet consiste à développer et d'intégrer un système d'information pour le campus numérique d'UMMTO dont les données sont intégrées dans un environnement réparti. L'objectif de ce travail est d'essayer de résoudre les problèmes de localisation des données et d'exécution distribuée des requêtes posées par la répartition d'une base de données à travers un réseau d'ordinateurs. Pour cela, nous avons conçu et mis en œuvre une base de données répartie.

Notre mémoire est structuré en 5 chapitres comme suit :

Dans le premier chapitre, nous présentons les espaces numériques de travail.

Dans le second chapitre nous expliquons les concepts de bases des bases de données réparties

Le troisième chapitre donne une vue sur la structure et les techniques d'intégration des sources de données.

Le quatrième chapitre présente l'analyse conceptuelle de la solution proposée.

Dans le dernier chapitre, nous détaillons la réalisation en deux parties, la première consiste à la présentation de l'environnement de développement, et la seconde à la présentation de quelques interfaces de notre application.

Nous finirons par une conclusion et quelques perspectives qui peuvent être envisagées afin d'améliorer notre système.

Chapitre I
Les espaces
numériques de
travail (ENT)

Introduction :

L'impact des technologies de l'information et de communication (TIC) sur le monde de l'enseignement supérieur a conduit, aujourd'hui à des grandes conversions numériques, dont des nouveaux services numériques tel que internet, courriels, publication, accès à des cours en ligne, forums, base de données ...etc. viennent de s'ajouter aux services traditionnels rendus par l'informatique de l'établissement, ils sont progressivement mis à disposition des étudiants, enseignants, personnels administratifs...

Tous ces services peuvent être regroupés, organisés et mis en cohérence. C'est précisément le rôle des environnements numériques de travail.

En cours de ce présent chapitre nous allons présenter et survoler sur quelques notions de base des espaces numériques de travail.

I.1 Définition :

Le SDET (*Schéma directeur des espaces numériques de travail*) donne une première définition dans un document de cadrage réalisé par le ministère français de l'Éducation nationale : « un espace numérique de travail désigne un dispositif global fournissant à un usager un point d'accès à travers les réseaux à l'ensemble des ressources et des services numériques en rapport avec son activité. Il est un point d'entrée pour accéder au système d'information de l'établissement ou de l'école. » [2]

Une seconde définition : l'ENT est le service en ligne accessible depuis n'importe quel navigateur connecté à l'Internet qui assemble les services numériques adaptés aux catégories d'utilisateurs : s'informer, produire des informations, consulter des ressources, organiser son travail, communiquer, travailler seul ou en groupe, apprendre, accompagner la scolarité de ses enfants, etc. [5]

I.2 Terminologies :

Les ENT apparaissent sous différentes appellations parmi lesquelles on cite :

- Virtual learning Environment (environnement d'apprentissage virtuel)
- Virtual Education Space (Espace d'apprentissage virtuel)
- Virtual learning Space (Espace d'apprentissage virtuel)
- Digital Learning Space/virtual (Espace d'apprentissage numérique)
- Educational Environment (Environnement éducationnel)
- Collaborative workspaces (Espace de travail collaboratif)
- Internet based distance-learning environments (Environnement d'apprentissage basé sur internet).
- Computer Environment for Human learning (Environnement informatique pour l'apprentissage humain)

- Computer based learning environment (Environnement d'apprentissage assisté par ordinateur).
- Learning Spaces (Espaces d'apprentissage).
- Espace Numérique de travail.
- E-Learning
- Etc.

I.3 Les types de campus numériques :

On distingue deux types de campus numérique qui sont les suivants :

➤ ***Les campus de « formation ouverte à distance » (ou FOAD) :***

Ceux-ci, assez classiques (car déjà bien répandus), permettent d'effectuer des cours totalement ou partiellement à distance, via des plateformes d'e-Learning, et via Internet.

➤ ***Les campus « environnements numériques de travail » :***

Ceux-ci sont moins répandus, mais plus englobant. Ils fournissent aux acteurs et usagers d'un système d'enseignement l'accès à tout un ensemble d'outils, de ressources, et de services numériques : documentation électronique, services en ligne, « bureau virtuel », etc.

I.4 Caractéristique : [4]

L'ENT permet d'offrir à chacun des acteurs du système éducatif et notamment aux étudiants, aux enseignants et aux personnels administratifs un accès simple à travers les réseaux à l'ensemble des services numériques en rapport avec son activité (par exemple : gestion ou consultation des absences, des notes, du cahier de texte de la classe ; diffusion et consultation de support de cours, de devoirs ; travail collaboratif, accès à des ressources ou des manuels numériques...).

Tout utilisateur peut ainsi, où qu'il soit, à partir d'un simple accès internet, retrouver son environnement de travail habituel, ses documents, ses ressources, communiquer avec son entourage, collaborer avec ses pairs, accéder à des services en ligne spécifiques. C'est particulièrement important pour l'élève qui peut ainsi, depuis tout lieu connecté à Internet (de son domicile, d'un Espace Public Numérique, de l'entreprise où il effectue son stage, entrer en communication avec ses interlocuteurs et poursuivre son activité en disposant de l'ensemble de ses ressources.

Jean-Michel Leclercq, chef de projet ENT SDTICE (sous-direction des technologies de l'information et de la communication pour l'éducation)

I.5 Objectifs : [2]

- ✓ Moderniser en permettant à chaque agent de mieux piloter son système d'information (pour manager, gérer, enseigner, etc.) ;

- ✓ Moderniser le service public en offrant à tous les étudiants des services numériques pour apprendre ou accompagner leur scolarité.
- ✓ Familiariser les étudiants avec des usages des technologies qui non seulement leur permettent de mieux apprendre mais encore de mieux comprendre la société de la connaissance dans laquelle ils auront à prendre place ;
- ✓ Rendre possible par tous et pour tous le recours à des formes d'enseignement et d'apprentissage alternatives.
- ✓ Les ENT doivent permettre de mutualiser les services entre plusieurs établissements afin de réduire les coûts de mise en œuvre et de fonctionnement, concentrer les compétences et augmenter la visibilité des services offerts.

I.6Avantages : [5]

- ✓ Les ENT permettent à chaque utilisateur de travailler à l'extérieur de l'établissement dans des conditions optimales.
- ✓ D'avoir d'un espace personnel de travail, de travailler en groupe.
- ✓ Faciliter l'organisation de la vie scolaire.
- ✓ Permettre l'accroissement de l'utilisation des technologies de l'information et de la communication dans les établissements.
- ✓ Favoriser l'autonomie des établissements,
- ✓ Permettre la réduction des coûts d'assistance et de maintenance.
- ✓ Les ENT influent sur la variation des pratiques pédagogiques.
- ✓ L'ouverture des établissements vers l'ensemble des membres de la communauté scolaire (parents d'élèves, conseillers d'orientation, ouvriers en charges de la maintenance, etc.).

De plus, l'ENT est un instrument de décroisement des activités d'apprentissage :

- Décroisement dans le temps : articuler le temps de la classe avec d'autres temps (continuité pédagogique) ;
- Décroisement dans l'espace : travailler en tout lieu de l'établissement scolaire et plus largement en tout lieu connecté à Internet ;
- Décroisement social : structure qui met en relation tous ses utilisateurs.

I.7Les acteurs : [3]

Les espaces numériques de travail fournissent à chaque usager un accès personnel à un ensemble de services en ligne. Par exemple, dans l'enseignement supérieur on trouvera les acteurs suivants:

- Les étudiants
- Les enseignants et enseignants-chercheurs (locaux et extérieurs);
- La direction ;
- Les personnels administratifs et techniques ;

- Les étudiants externes à l'établissement ;
- Les lycéens, futurs étudiants ;
- Les anciens étudiants ;
- Les établissements d'enseignement supérieur partenaires ;
- Les associations ;
- Les syndicats ;
- Les partenaires publics ;
- Les collectivités territoriales ;
- Les entreprises ;
- Les chercheurs ;
- Les administrateurs de l'ENT.

I.8 Les services offerts par les ENT :[1]

Les espaces numériques de travail vont offrir à l'utilisateur différents services en fonction de son profil, des services numériques ont été groupés en trois catégories. Tout d'abord, les services de base, qui rassemblent des services non spécifiques au domaine éducatif sur lesquels peuvent s'appuyer des services plus spécifiques.

Ensuite, certains services applicatifs devraient être fortement couplés au socle de l'espace numérique de travail dans la mesure où ils s'appuient de manière significative sur ses mécanismes intégrateurs et sur les services de base. Au-delà des services de base, un espace numérique de travail doit offrir un ensemble cohérent de services applicatifs.

Enfin, d'autres services peuvent être faiblement couplés au socle de l'espace numérique de travail, c'est-à-dire accessible par son intermédiaire à travers un lien plus ou moins riche en fonction des besoins.

I .8.1 Les services de bases :

- les services communs à tous les services :

- Inscription à l'espace de travail,
- Identification unique et gestion des profils,
- Personnalisation de l'environnement
- Gestion des groupes d'utilisateurs,
- Moteur de recherche, aide,

- les services de communication :

- Courrier électronique,
- Forum de discussion et listes de diffusion,
- Annuaire (page blanche)
- Chat, visioconférence

- les services du bureau numérique :

- Carnet d'adresses,
- Espace de travail et de stockage, agenda,
- publication Web,
- Outils de bureautique

- Les services Back-office :

- Administration de l'environnement numérique de travail,
- Indicateur d'usages

1.8.2 Les services applicatifs :

- Services d'enseignement à distance
- Services d'accès aux ressources pédagogiques
- Tutorat et assistance
- Classes virtuelles
- Services documentaires
- Services de gestion de la scolarité
- Services de la vie universitaire

1.8.3 les services socle :[6]

Les services du socle ENT sont considérés comme étant des services de supports mis à disposition des services applicatifs, ils sont donc considérés comme des services « Back Office », c'est-à-dire il gère tout ce qui est masqué vis-à-vis de l'utilisateur.

Le socle est organisé de trois grands types de services mutualisés qui sont les suivants :

- **Services AAS** (Authentification - Autorisation - Single Sign On) : c'est la gestion des identités et des accès « contrôle d'accès »
- **Services portail** : s'occupe de l'accueil des utilisateurs, personnalisation, présentation etc...
- **Services socle mutualisés** : service d'échanges pour la messagerie, administration de la plate-forme, gestion de l'espace de stockage.

1.9 Architecture générale : [3]

Un espace numérique de travail se base sur une plate-forme « socle » qui permet d'accéder aux différentes briques du système d'information.

Les fonctions gérées par le socle sont les suivantes :

- L'accès aux différentes composantes du système d'information (briques) ;
- La présentation sur le terminal de l'utilisateur ;
- La sécurité de l'ensemble du dispositif et des flux d'information ;
- L'authentification, la gestion des autorisations et le SSO ;

- La gestion du Back Office ;
- Les fonctions de personnalisation, pour la structure (dans le Back Office) ou l'utilisateur (Front Office) ;
- Un moteur de recherche sur l'ensemble des composantes du système d'information.

Remarque :

Les différentes briques du système d'information ne peuvent généralement pas être intégrées directement à l'espace numérique de travail car :

- Beaucoup d'applicatifs aujourd'hui ne disposent pas d'une interface Web qui est la cible principale de l'ENT ;
- Pour les applications, la présentation est souvent intégrée au programme et ne peut être paramétrée ;
- Les applications Web renvoient la plupart du temps un flux HTML et non-XML, ce qui rend très difficile l'utilisation d'autres clients que les navigateurs Web ;
- Les applications reposent souvent sur des annuaires différents pour l'authentification ;
- Lorsqu'elles reposent sur le même annuaire, elles ne gèrent pas forcément le SSO.

En conséquence, il sera nécessaire de développer une interface normalisée entre les applications existantes et l'espace numérique de travail, ce qu'on appelle « interopérabilité applicative ».

Le schéma suivant représente les différentes fonctionnalités de l'architecture d'un ENT proposé par Le SDET (Schéma directeur des espaces numériques de travail) du S3it (Schéma Stratégique des Systèmes d'Information et des Télécommunications).[SDET.v1]

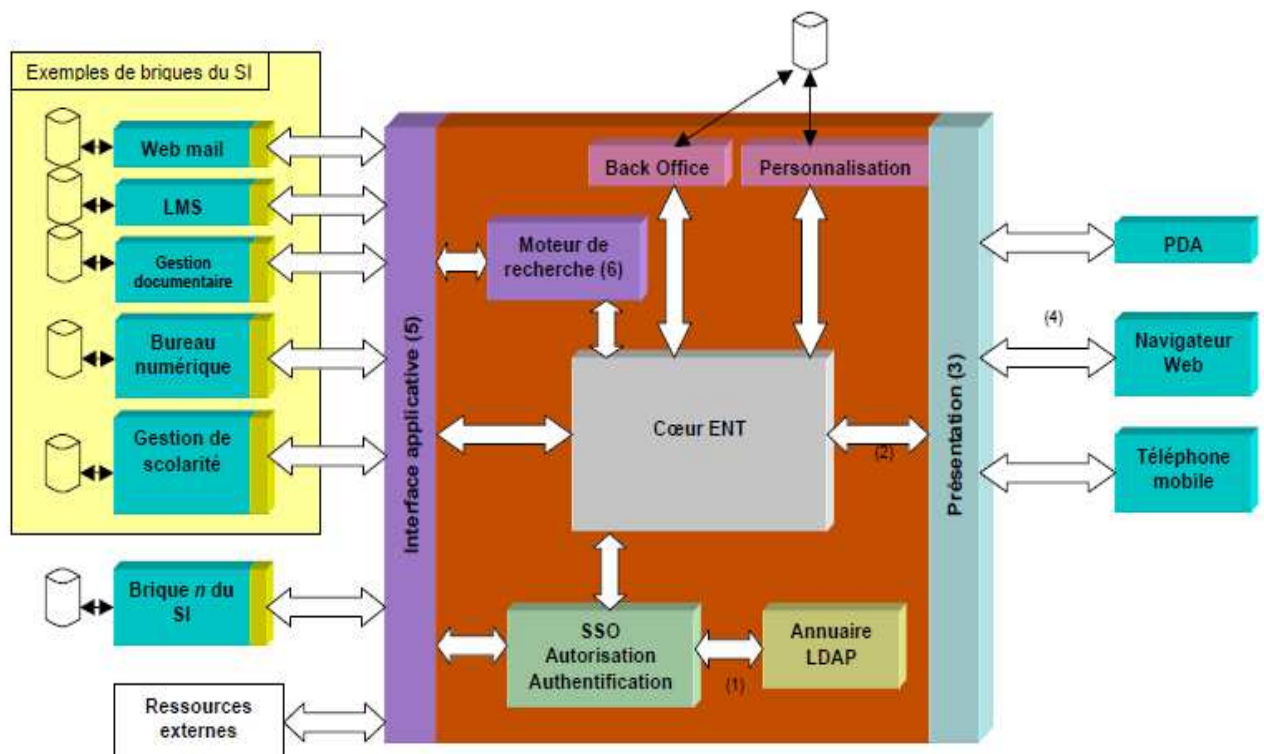


Figure I.1 : Architecture générale d'un espace numérique de travail

Ce schéma présente l'architecture logique de l'espace numérique de travail, dont les différentes briques citées le sont à titre illustratif et ne correspondent qu'à une petite partie des composantes logicielles du système d'information.

- (1) : le protocole utilisé pour communiquer avec l'annuaire est LDAP V3.
- (2) : le cœur de l'ENT doit communiquer avec le module de présentation selon un protocole utilisant le métalangage XML.
- (3) : la couche présentation convertit le flux XML en un flux adapté au client de l'utilisateur. Cette traduction a aussi lieu lors de la remontée d'information du client vers l'ENT
- (4) : la couche présentation doit communiquer avec un navigateur Internet selon le protocole HTML4.0
- (5) : l'ENT doit communiquer avec les briques à travers l'interface applicative selon des protocoles normalisés utilisant le métalangage XML (voir interface applicative)
- (6) : le moteur de recherche devrait être en mesure d'effectuer, à la demande de l'ENT, des recherches sur l'ensemble des données des briques du système d'information.

1.10 Architecture technique du socle de l'ENT :

L'architecture privilégiée pour le socle de l'ENT doit être une architecture 3 tiers logiques, composée :

- ✓ du client ou demandeur de ressources ;
- ✓ du serveur d'application ;
- ✓ du serveur de base de données.

La figure suivante illustre l'architecture du socle de l'ENT

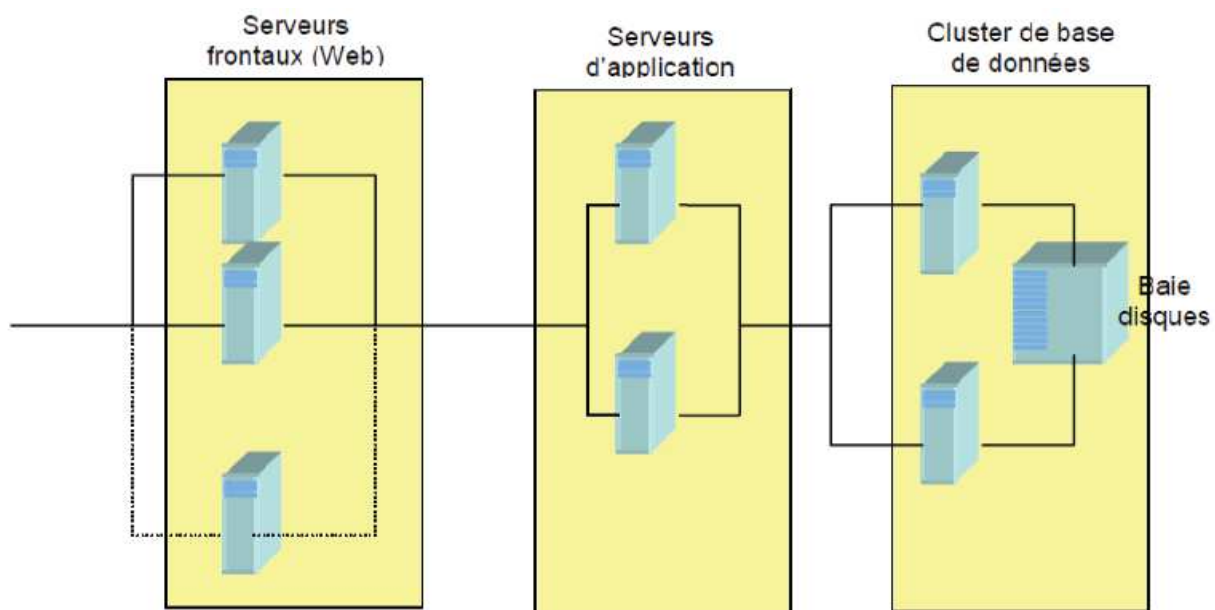


Figure I.2 : Architecture technique logique du socle de l'espace numérique de travail

I.11 Les moyens à mettre en œuvre : [6]

Pour la réussite d'un projet d'intégration d'un ENT dans un établissement scolaire il sera nécessaire de mettre en œuvre des moyens techniques, humains, juridiques et financiers.

I.11.1 Moyens techniques :

Il est important de consolider le parc informatique de l'établissement, d'assurer le bon fonctionnement des réseaux et de disposer d'une connexion internet d'un débit suffisant. Ensuite, la mise en place d'un ENT se traduit concrètement par la possibilité d'accéder à une plate-forme commune, et la distribution des comptes d'utilisateurs. Enfin, il faut mettre en place une structure d'assistance et de maintenance aux utilisateurs.

I.11.2 Moyens humains :

Se traduisent par la pleine participation de l'ensemble des acteurs à la conception de l'ENT de l'établissement (personnels de l'établissement, étudiants, enseignants, Administrateurs...etc.). Il faut ensuite assurer la formation des utilisateurs aux fonctionnalités d'un ENT pour l'intégration la plus efficace possible de ce nouvel outil. Il faut enfin mettre en place une structure d'échange et de mutualisation pour permettre aux acteurs de collaborer et de partager les différentes expériences.

I.11.3 Moyens juridique :

C'est de définir des règles communes de fonctionnement et d'accompagner l'utilisation d'ENT des droits d'auteurs et les licences d'utilisation des ressources pédagogiques ou administratives et des logiciels.

I.11.4 Moyens financiers :

Ils sont affectés à trois domaines principaux : l'équipement des établissements, l'assistance et la maintenance et l'acquisition des droits et des licences d'utilisation, d'après les études réalisées par la Caisse des Dépôts à ce sujet, ils ont constaté que des économies liées notamment à la dématérialisation des échanges (téléphone, enveloppe, courrier) peuvent compenser les coûts de l'ENT par contre l'utilisation de logiciels libres permet de limiter le budget à la maintenance du matériel, mais demande de financer une communauté de développeurs assez conséquente pour disposer d'une qualité de service optimum et fédérer les différentes contributions.

I.12 Les Recommandation officielle pour les ENT : [5]

L'État français a fait de nombreuses recommandations pour la mise en place d'un ENT, parmi lesquelles les suivantes :

- **S3IT (Le Schéma stratégique des systèmes d'informations et de télécommunications)** : apparaît en 2002 et est réactualisé en 2006. Il prévoyait la rédaction du Schéma directeur des environnements de travail (SDET), renommé depuis Schéma directeur des espaces numériques de travail (SDET)

- **SDET (Schéma Directeur des Espaces Numérique de Travail) :**

Le SDET est piloté par la Direction de la Technologie et s'inscrit dans le Schéma stratégique des systèmes d'information et de télécommunication (S3IT). Il est remis à jour en 2011.

Le SDET s'adresse aux acteurs des projets ENT, et il est destiné à être un guide dans les mises en places technique, fonctionnelle et organisationnelle, il est aussi un instrument de dialogue entre les différents partenaires des projets.

- **SDI :**

Le Schéma directeur des infrastructures (SDI) expose les préconisations pour le raccordement des établissements à l'Internet.

- **S2i2e**

La note de cadrage Services Internet et Intranet pour les établissements scolaires et les Écoles (S2i2e) détaille les grands principes d'organisation, de gestion, et de sécurisation de leurs réseaux locaux.

1.13 Exemple de projets compatibles ENT SDET pour l'université :

Les exemples suivants sont des solutions logicielles de type ENT déployées dans des établissements scolaires en France. Ces solutions ont été retenues par le ministère de l'éducation nationale et respectent le cahier des charges du SDET 2.0 (11-07-2006).

- ✓ *Ancora* : se base sur ContactOffice et est soutenu par les universités de Lyon et Grenoble
- ✓ *ESUP* : se base sur des technologies open-source (site internet Esup-Portail) ; socle uPortal (JA-SIG) + SSO cas géré par l'université de valenciennes. Esup-Portail est soutenu par un consortium universitaire, le consortium ESUP-Portail
- ✓ ENT itslearning Supérieur ENT incluant des modules auteurs pédagogiques pour les enseignants et les étudiants - site internet.
- ✓ EPPUN se base sur des technologies diverses
- ✓ K-Sup - site internet K-Sup, édité par Kosmos - site internet Kosmos
- ✓ Monte-Cristo, soutenu par l'université de Corse, se base sur Blackboard

Conclusion :

Dans ce chapitre nous avons présenté les concepts de base des espaces numériques de travail, terminologies, objectifs, et leurs avantages ainsi les services offerts, et nous avons expliqué d'autre part son architecture logique et enfin nous avons terminé par des exemples illustratifs des projets ENT réalisés.

Le chapitre suivant sera consacré à l'étude de la base de données répartie, qui sera intégré à notre ENT.

Chapitre II
Les bases de
données
Réparties

Introduction

L'évolution des techniques informatiques depuis les vingt dernières années ont permis d'adapter les outils informatiques à l'organisation des entreprises. Vu, le grand volume de données manipulées par ces dernières, la puissance des micro-ordinateurs, les performances des réseaux et la baisse considérable des coûts du matériel informatique ont permis l'apparition d'une nouvelle approche afin de remédier aux désagréments causés par la centralisation des données, et ce en répartissant les ressources informatiques tout en préservant leur cohérence.

Pour palier aux problèmes des bases de données centralisées et ces divers inconvénients, l'utilisation d'une base de données répartie est nécessaire, vu qu'elle dispose d'une architecture plus adaptée à l'organisation des entreprises décentralisées.

Avant de concevoir une base de données répartie, il est nécessaire de bien comprendre les étapes de conception, car différentes méthodes de conception existent et chacune d'elles nous offre une approche très différente de l'autre.

Dans le cas d'une base de données répartie, la difficulté réside dans le choix des techniques de conception, un mauvais choix pourrait conduire à la création d'un système inefficace.

La conception d'une base de données répartie peut être le résultat de deux approches totalement distinctes, soit d'une part le regroupement d'une multitude de bases de données déjà existantes dans ce cas on parle des bases de données fédérées, d'autre part, que cette dernière soit construite du zéro.

II.1 Notion générales :

II.1.1 Définition d'une base de données :

Une base de données (*database*) est un ensemble structuré et cohérent de données enregistrées avec le minimum de redondance pour satisfaire simultanément plusieurs utilisateurs de manière sélective et dans un temps opportun. [10]

Selon Wikipédia une **base de données** (anglais **database**) est un conteneur informatique permettant de stocker dans un même endroit l'intégralité des informations en rapport avec une activité. Une base de données permet de stocker et de retrouver un ensemble d'informations de plusieurs natures ainsi que les liens qu'il existe entre les différentes natures.

II.1.2 Système de gestion de base de données :

Un système de gestion de base de données (SGBD) est un ensemble de logiciels permettant la création, la manipulation et la maintenance de bases de données.

Et selon la définition du Wikipédia : Le système de gestion de base de données (abr. SGBD) est une suite de programmes qui manipule la structure de la base de données et dirige l'accès aux données qui y sont stockées. Il sert d'intermédiaire entre la base de données et ses usagers. La base de données est composée d'une collection de fichiers et le seul moyen d'accéder aux données est par l'intermédiaire du SGBD. Celui-ci reçoit des demandes de manipulation du contenu et effectue les opérations nécessaires sur les fichiers. Il cache la complexité des opérations et offre une vue synthétique sur le contenu. Le SGBD permet en outre à plusieurs usagers de manipuler simultanément le contenu, et peut offrir différentes vues sur un même stock de données

II.1.3 Architecture d'un SGBD :

Les groupes de normalisation se sont penchés depuis fort longtemps sur les architectures de SGBD. A la fin des années 70, le groupe ANSI/X3/SPARC DBTG a proposé une architecture intégrant les trois niveaux de schémas : externe, conceptuel et interne. Cette architecture permet de bien comprendre les niveaux de description et transformation de données possible dans un SGBD.

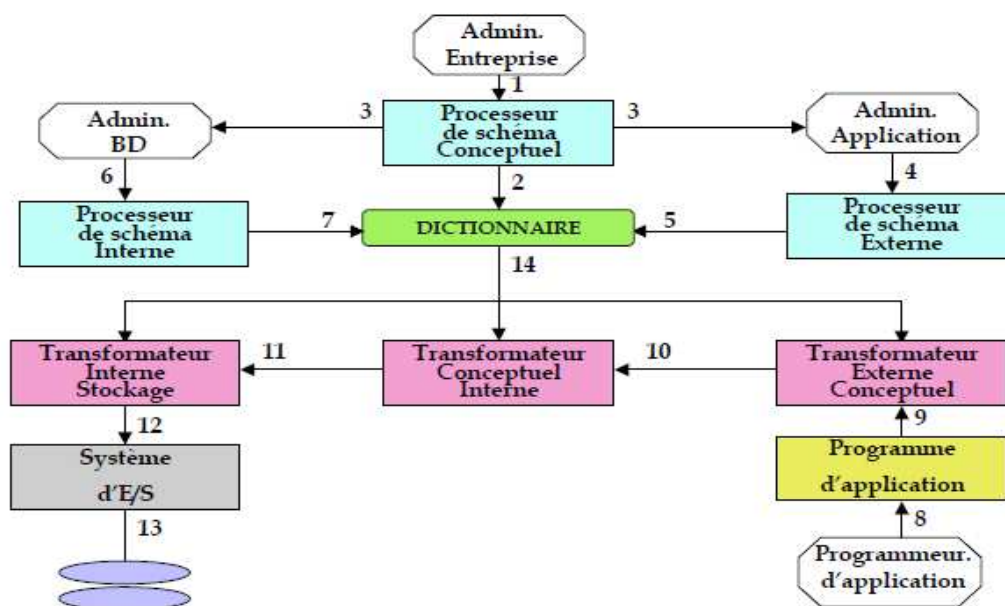


Figure II.1: Architecture typique d'un SGBD

II.2 les bases de données réparties :

II.2.1 Définition : [11]

Une base de données répartie (BDR) est une base de données dont différentes parties sont stockées sur des sites, généralement géographiquement distants, reliés par un réseau. La réunion de ces parties forme la base de données répartie.

Un système de bases de données réparties ne doit donc en aucun cas être confondu avec un système dans lequel les bases de données sont accessibles à distance (selon le principe client serveur). Il ne doit non plus être confondu avec un système multibase. Dans ce dernier cas, chaque utilisateur accède à différentes bases de données en spécifiant leur nom et adresse, et le système se comporte alors simplement comme un serveur de BD et n'apporte aucune fonctionnalité particulière à la répartition.

II.2.2 Buts de la répartition des bases de données :

- ✓ *Plus de fiabilité* : les bases de données réparties ont souvent des données répliquées. La panne d'un site n'est pas très importante pour l'utilisateur, qui s'adressera à autre site.
- ✓ *Meilleures performances* : réduire le trafic sur le réseau est une possibilité d'accroître les performances. Le but de la répartition des données est de les rapprocher de l'endroit où elles sont accédées. Répartir une base de données sur plusieurs sites permet de répartir la charge sur les processeurs et sur les entrées/sorties.
- ✓ *Faciliter l'accroissement*: l'accroissement se fait par l'ajout de machines sur le réseau.

II.2.3 Objectifs des bases de données réparties [12]

Les principaux objectifs définis par C.DATE sont les suivants :

1. Autonomie locale
2. Egalité entre sites (pas de site « central »)
3. Fonctionnement continu (pas d'interruption de service)
4. Localisation transparente
5. Fragmentation transparente
6. Indépendance à la réplication
7. Exécution de requêtes distribuées
8. Gestion de transactions réparties
9. Indépendance vis-à-vis du matériel
10. Indépendance vis-à-vis du Système d'Exploitation
11. Indépendance vis-à-vis du réseau
12. Indépendance vis-à-vis du SGBD

Parmi tout ces objectifs cités nous allons détailler les plus importants :

- **Autonomie locale : [7]**
La BD locale est complète et autonome (intégrité, sécurité, gestion), elle peut évoluer indépendamment des autres.
- **Egalité entre sites**
Un site en panne ne doit pas empêcher le fonctionnement des autres sites (mais perturbations possibles)
- **Transparence de localisation :**
Les utilisateurs accèdent à la BD soit directement par le schéma conceptuel soit indirectement au travers de vues externes. Mais en aucun cas ils n'ont les moyens d'accéder aux schémas locaux ni de préciser le site.
- **transparence de partitionnement :**
Les utilisateurs n'ont pas à connaître les partitionnements de la base de données et ne doivent pas aussi savoir si telle information est fractionnée, et ne doivent donc pas se préoccuper de la réunifier. C'est le système qui gère les partitionnements et les modifie en fonction de ses besoins, et c'est donc lui qui doit rechercher toutes les partitions et les intégrer en une seule information logique présentée à l'utilisateur.
- **Fragmentation transparente :**
Des données d'une même table éparpillées (fragmentées) doivent être vues comme untout
- **Indépendance vis-à-vis du SE :**
Le SGBD fonctionne sur les différents systèmes d'exploitation.(Unix, Linux, Windows ...)

II.3 Conception d'une base de données répartie :

La définition du schéma de répartition est la partie la plus délicate de la phase de conception d'une BDR, car il n'existe pas de méthode miracle pour trouver la solution optimale. L'administrateur doit donc prendre des décisions dont l'objectif est de minimiser le nombre de transferts entre sites, les temps de transfert, le volume de données transférées, les temps moyens de traitement des requêtes, et le nombre de copies de fragments, ... etc.

II.3.1 Conception descendante (top down design) :

On commence par définir un schéma conceptuel global de la base de données répartie, puis on distribue sur les différents sites en des schémas conceptuels locaux. La répartition se fait donc en deux étapes, en première étape la fragmentation, et en deuxième étape l'allocation de ces fragments aux sites. L'approche *top down* est intéressante quand on part du néant.

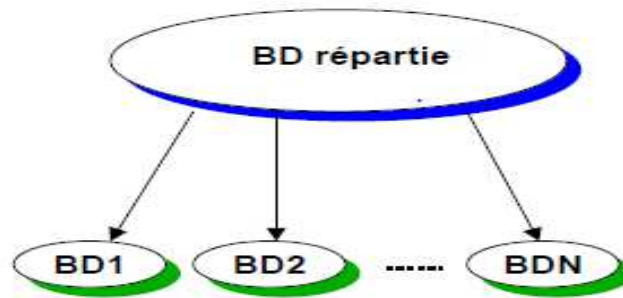


Figure II.2 : Conception descendante

II.3.2 Conception ascendante :

Cette approche se base sur le fait que la répartition est déjà faite, mais il faut réussir à intégrer les différentes BDs existantes en une seule BD globale. En d'autre terme, les schémas conceptuels locaux existent et il faut réussir à les unifier dans un schéma conceptuel global.

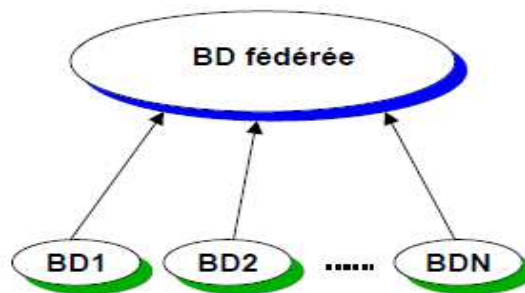


Figure II.3: Conception ascendante

II.4 La fragmentation : [12]

II.4.1 Définition : La fragmentation est le processus de décomposition d'une base de données en un ensemble de sous bases de données. Cette opération de décomposition doit être nécessairement sans perte d'information.

La fragmentation possède deux règles principales qui sont les suivantes :

1. **La complétude:** pour toute donnée d'une relation R , il existe un fragment R_i de la relation R qui possède cette donnée.
2. **La reconstruction:** pour toute relation décomposée en un ensemble de fragments R_i , il existe une opération de reconstruction.

II.4.2 Types de fragmentation [13]

a) La fragmentation horizontale :

C'est un découpage d'une table en **sous tables** par utilisation de prédicats permettant de sélectionner les lignes appartenant à chaque fragment. L'opération de fragmentation est obtenue grâce à la sélection des tuples d'une table selon un ou des critères bien précis.

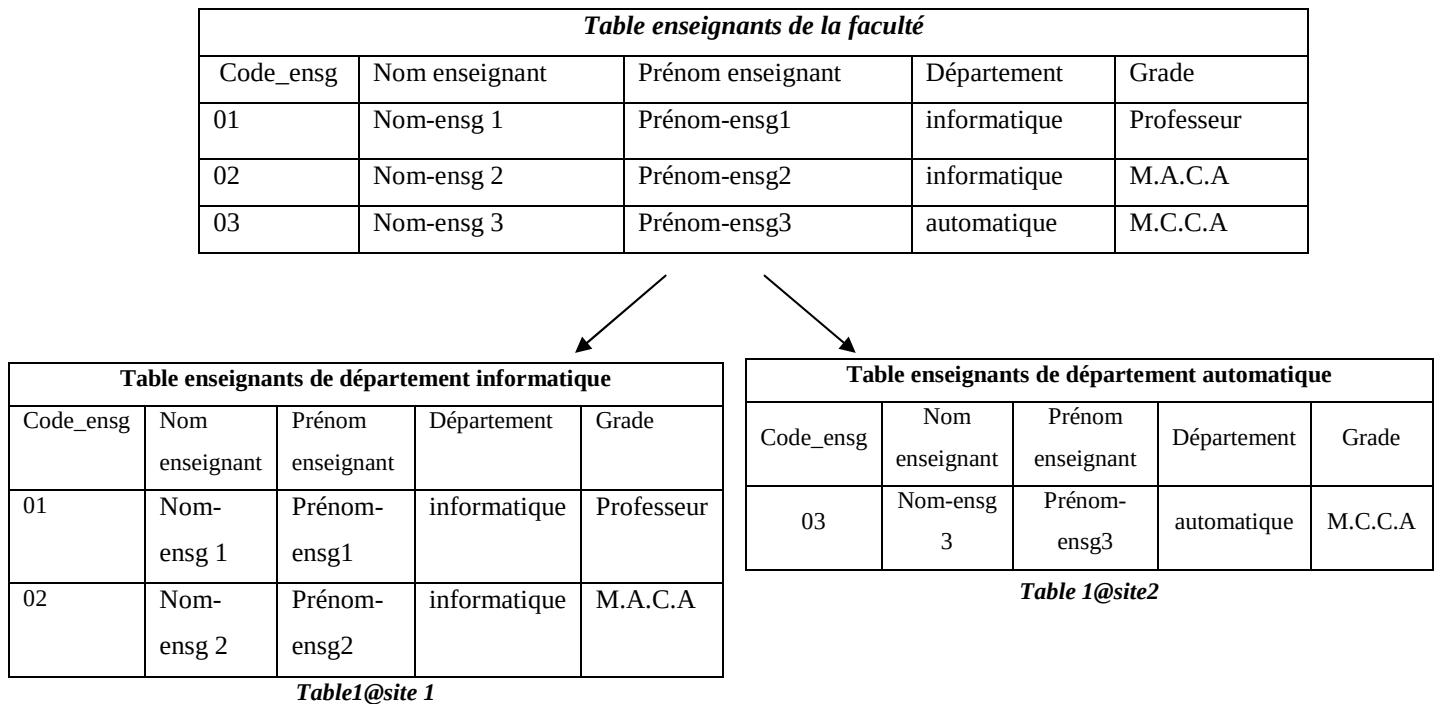


Figure II.4: fragmentation horizontale

Assemblage de fragmentation horizontale sous SQL : la reconstitution de la relation initiale se fait grâce à l'union (U) des sous-relations. la requête de la reconstruction est la suivante.

```
CREATE VIEW V1
AS SELECT Table1.cle, Table1.attr1
FROM Table1@site1
UNION
SELECT Table2.cle, Table2.attr1
FROM Table2@site2
```

b) La fragmentation verticale :

Elle est le découpage d'une table en sous tables par projection permettant de sélectionner les colonnes composant chaque fragment. La relation initiale doit pouvoir être recomposée par la jointure des fragments.

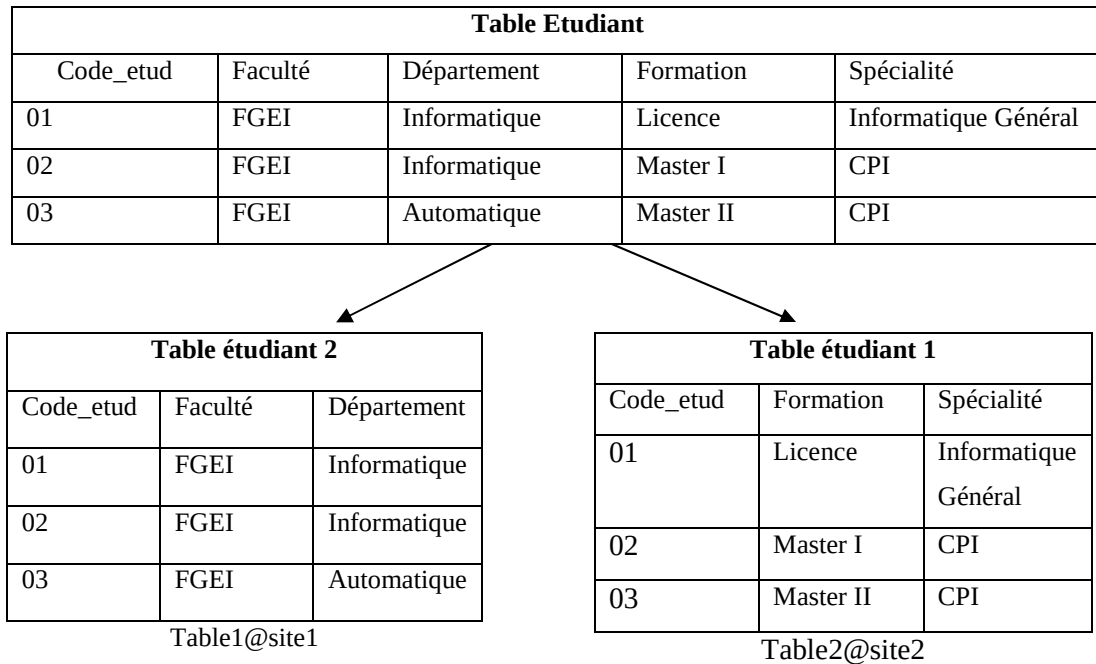


Figure II.5: fragmentation verticale

Assemblage : la reconstruction de la table initiale se fait par la jointure comme nous l'avons cité précédemment on exécutant des requêtes qui ressemblent à celle-ci :

```
CREATE VIEW V1
AS SELECT Table1.cle, Table1.attr1,
Table2.attr2
FROM Table1@site1, Table2@site2
WHERE Table1.cle=Table2.cle
```

c) *Fragmentation hybride (fragmentation mixte)* : [14]

La fragmentation hybride est la répartition des valeurs cette dernière se fait par la combinaison des deux fragmentations précédentes, horizontale et verticale. Les occurrences et les attributs peuvent donc être répartis dans des partitions différentes.

L'opération de partitionnement est une combinaison de projections et de sélections et donc l'opération de recombinaison est une combinaison de jointures et d'unions.

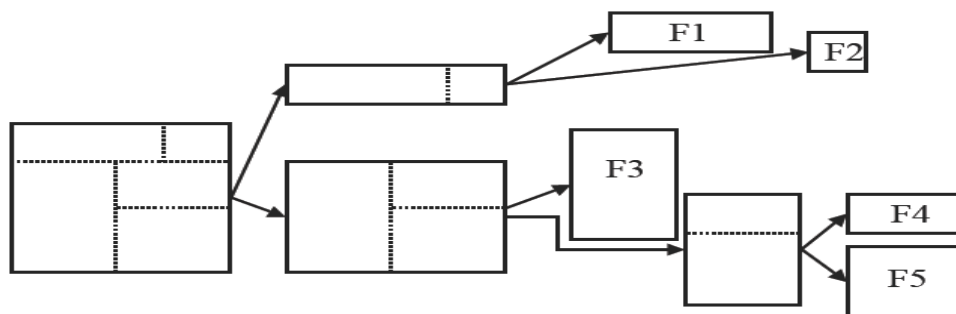


Figure II.6: fragmentation hybride (Mixte)

II.4.3L'allocation des fragments :

Le processus de fragmentation est suivie par un autre processus qui est l'allocation de ces fragments, ce dernier consiste à localiser les fragments sur les sites où ils sont le plus utilisés, l'objectif principale de cette opération est de minimiser les transferts de données entre les sites.

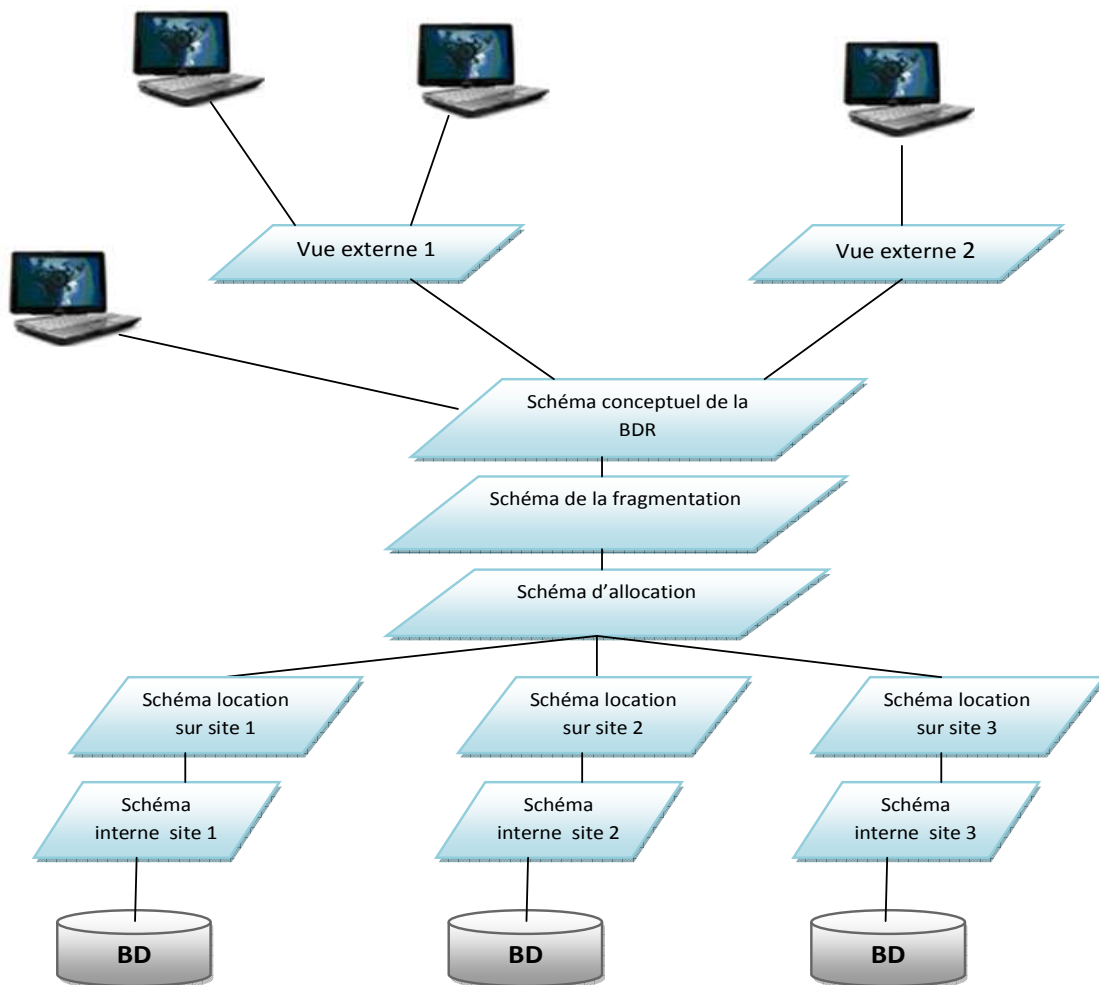
Un schéma doit être élaboré afin de déterminer la localisation de chaque fragment et sa position dans le schéma global.

II.4.4Le schéma de répartition

Pour fragmenter les requêtes, il est nécessaire de connaître les règles de localisation des données. Lors de l'exécution d'une requête, le SGBDR doit décomposer la requête globale en sous requêtes locales en utilisant le schéma de répartition.

II.5Architecture de schéma d'une base de données distribuée :

Après la présentation des techniques de fragmentation et d'allocation, la figure suivante présente l'architecture d'une base de données distribuée :



FigureII.7 : architecture de schéma d'une base de données distribuée

II.6 Les systèmes fédérés : [8]

L'**architecture fédérée** est une forme d'architecture centrée sur les données dont Heimbigner et McLeod ont introduit ce dernier terme par la définition essentielle de ce qu'est une base de données fédérée faiblement couplée en anglais « collection of components to uniteloosecoupledfederation in order to share and exchange information » ainsi une base de données fédérée fortement couplée « en anglais :an organization model based on equal, autonomoustatabases, with sharing controlled by explicit interfaces. » que nous allons détailler dans ce qui suit

Toutes BDD d'une fédération doit avoir trois schémas.

- ES: Schéma d'Exportation: pour les données permises à l'exportation
- IS: Schéma d'Importation: pour les données importées
- PS: Schéma Privé: pour l'ensemble de données privées, dont les ES et IS sont inclus.

Et aussi il doit y avoir un Dictionnaire de Fédération (FD)

II.7 Catégories des systèmes fédérés :[5]

II.7.1 Les fédérations faiblement couplées :Réalisant l'intégration des différentes bases de données à l'aide des mécanismes des bases de données multibases et réparties et donc la tâche de maintenir la fédération est laissée aux administrateurs locaux

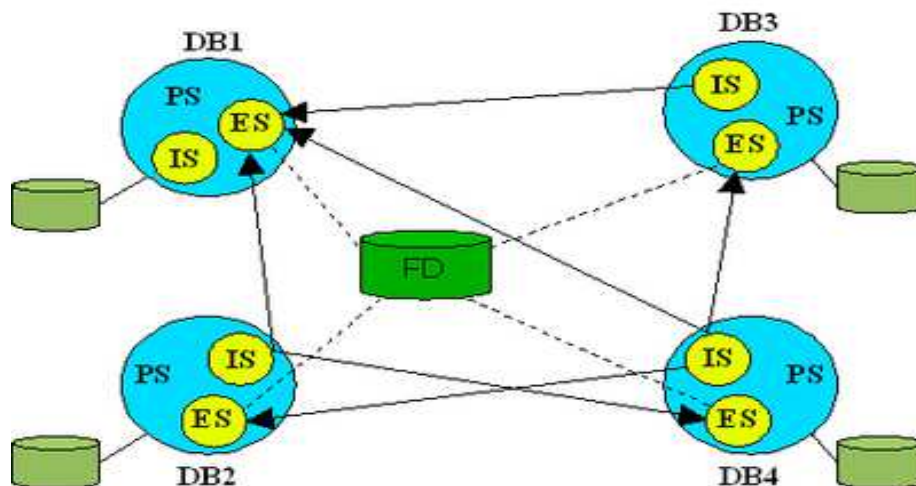


Figure II.8 : La fédération faiblement couplées

Dans la figure précédente nous avons quatre bases de données distinctes, offrant la portion de leurs données qui est nécessaire à la fédération via leurs schémas d'exportations (ES). Le dictionnaire de Fédération (FD) contient Les informations concernant le schéma global et

nous retrouvons également les informations indiquant comment effectuer l'importation des données pour réaliser des vues sur le schéma global

Nous pouvons remarquer aussi l'absence de schéma globale sur la figure qui est nécessaire pour une base de données fédérée ; mais une telle architecture ce dernier existe d'une façon virtuelle, les différents administrateurs de données locaux doivent créer des vues à la main sur les différents schémas d'exportation pour réaliser une vue totale ou partielle (mais virtuelle) du schéma global. Les schémas d'exportation (ES) sont en général des vues sur une portion de la base locale de manière à permettre la correspondance entre cette portion de la base et le schéma global. Nous constaterons que les schémas originaux restent inchangés et que les utilisateurs locaux peuvent continuer à utiliser leurs anciennes bases comme si de rien n'était. La seule nouveauté pour un usager local est l'accès possible aux données de la fédération via son IS local.

II.7.2 Les fédérations fortement couplées :

Les systèmes fortement couplés réalisant l'intégration à l'aide d'un SGBD fédéré (le maintien de la fédération est sous le contrôle de l'administrateur de la fédération).

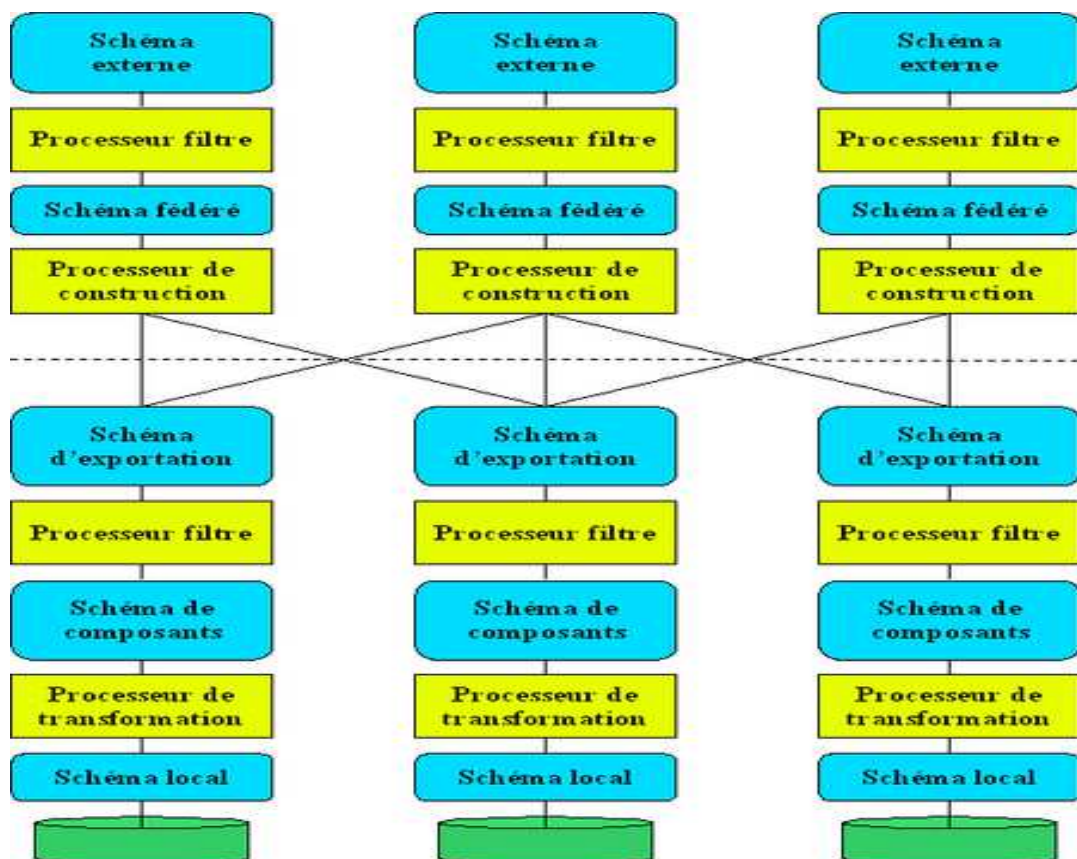


Figure II.9: La fédération fortement couplée

L'architecture fortement couplée est subdivisée en deux parties séparées avec une ligne pointillée (figure :) la partie supérieure responsable de l'intégration des différents BD de la partie inférieure (traducteur) qui permet la traduction d'une BD dans un modèle canonique. Tel que le déploiement de cette architecture avec les traducteurs sont stockés sur les mêmes hôtes que les SGBD locaux et l'intégrateur sur un hôte particulier. Cette architecture est composée essentiellement de processeurs et de schémas

Les processeurs :

➤ **Processeurs de transformation :**

Transforme une commande effectuée dans un langage (ex.: SQL) en une requête dans un autre langage (ex. : CODASYL). Ce processeur offre une vision transparente des modèles de données. Ils convertissent donc un schéma conçu selon un modèle en un autre

➤ **Processeurs filtres :**

Ces processeurs vérifient les commandes envoyées à un processeur ou les résultats retournés par un processeur pour s'assurer qu'ils ne violent pas des règles d'intégrité syntaxique et/ou sémantique. Ces processeurs peuvent s'occuper également de la vérification des droits d'accès aux données. Concrètement, ils sont responsables du maintien de la fédération en offrant des mécanismes explicites permettant de résoudre les contraintes du schéma global.

➤ **Processeurs de construction :**

Ce processeur occupe de la division et/ou l'intégration des commandes ou les données envoyées par un processeur à plusieurs processeurs (répartition) ou inversement c'est-à-dire de plusieurs processeurs (intégration)

Les autres tâches complémentaires exécutées par ces processeurs sont les suivantes :

- ✓ Intégration de schémas
- ✓ Négociation de protocoles entre processeurs
- ✓ Optimisation de requêtes et décomposition des requêtes pour différents processeurs
- ✓ Gestion des transactions (atomicité et concurrence)

Les schémas :

La partie inférieure de l'architecture fortement couplée se compose de trois schémas :

- **Schémas locaux :** c'est le schéma des bases de données participants qui sont exprimés dans leurs modèles (relationnel, réseau, hiérarchique).
- **Schémas des composants :** Ils correspondent aux schémas des bases de données participantes une fois que les schémas sont traduits dans un modèle de données commun (pivot).
- **Schémas d'exportation :** c'est la portion des schémas des composants destinés à l'intégration au sein de la fédération.

Ainsi la partie supérieure se compose de deux schémas :

- **Schéma fédéré** : Ils'agit du schéma de la fédération proprement dit.
- **Schémas externes** : Équivalents aux schémas externes de l'architecture Ansi/Sparc mais au niveau de la fédération.

II.7.3 Les bases de données fédérées :

Une base de donnée fédérée est une collection de bases de données hétérogènes capables d'interopérer via une vue commune [9]

L'objectif de la création d'une base fédérée est de donner aux utilisateurs une vue unique des données présentes sur plusieurs systèmes a priori hétérogènes. Faire cohabiter les différents systèmes tout en leur permettant d'inter opérer d'une manière harmonieuse. Il ne s'agit pas ici de reconstruire une base répartie à partir des anciennes, mais de conserver les anciennes bases et leurs autonomies locales tout en permettant l'intégration de l'ensemble.

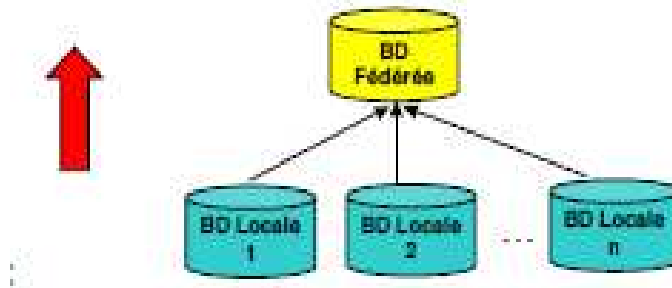


Figure II.10 : Base de données fédérée

II.8 Les SGBD fédérés : [15]

SGBDF pour Système de Gestion de Bases de Données Fédérées et possède de multiples points en commun avec les SGBD traditionnels et répartis. En fait, nous pouvons considérer ces systèmes comme des SGBD répartis pour lesquels les schémas internes sont des SGBD traditionnels. Il est donc naturel de concevoir un SGBDF comme une évolution des SGBD répartis.

Actuellement, la norme qui s'impose en matière de SGBDF est conçue par IBM qui a fait des investissements significatifs en technologies de fédération qui ont eu comme conséquence de principales possibilités du marché à travers la brochure de produit de gestion de l'information. Aujourd'hui, les possibilités de fédération permettent l'accès unifié à n'importe quelle information numérique, dans n'importe quel format structuré et non structuré et dans n'importe quel base de données. Les possibilités de fédération sont disponibles par une variété de produits d'IBM comprenant le serveur de fédération d'InfoSphere DB2 pour Linux, UNIX, et Windows, et le portail de l'information d'entreprise d'IBM (EIP).

II.8.1 Définition de DB2 d'IBM :

DB2 est l'un des systèmes de gestion de base de données propriétaire d'IBM. Il utilise le langage SQL tout comme Oracle, PostgreSQL ou MySQL. Il est déployé sur les Mainframes, systèmes UNIX, Windows, Mac/OS et Linux. Il existe également une version allégée pour les ordinateurs de type Palm.[5]

II.8.2 Les avantages et les inconvénients du DB2 d'IBM [15]

a) Les avantages :

1. SQL proche de la norme
2. Nombreux assistants qui auraient dû permettre une administration plus aisée (mais très gourmande en ressource)
3. Richesse fonctionnelle du langage et des jointures
4. Gestion centralisée de plusieurs instances
5. Ordonnanceur intégré
6. Stockage des données XML, data links
7. Compression des données stockées

b) Les inconvénients :

- Interface client Java lourd instable, peu conviviale avec des menus contextuels à rallonge. Dans les faits, seule la ligne de commandes semble stable.
- Journalisation gourmande en disques.
- Particulièrement instable sous Windows en configuration mémoire dynamique
- faiblesse de DDL, bien que les assistants tentent maladroitement de cacher ces faiblesses en exécutant des traitements lourds
- Prix exorbitant, tant au point de vue des licences que des composants matériels (RAM, CPU) à fournir pour de bonnes performances
- Gestion des utilisateurs extrêmement limitative, dédiée à l'OS
- Complexité du langage procédural.

II.8.3 Les composants d'un système fédéré avec DB2[15]

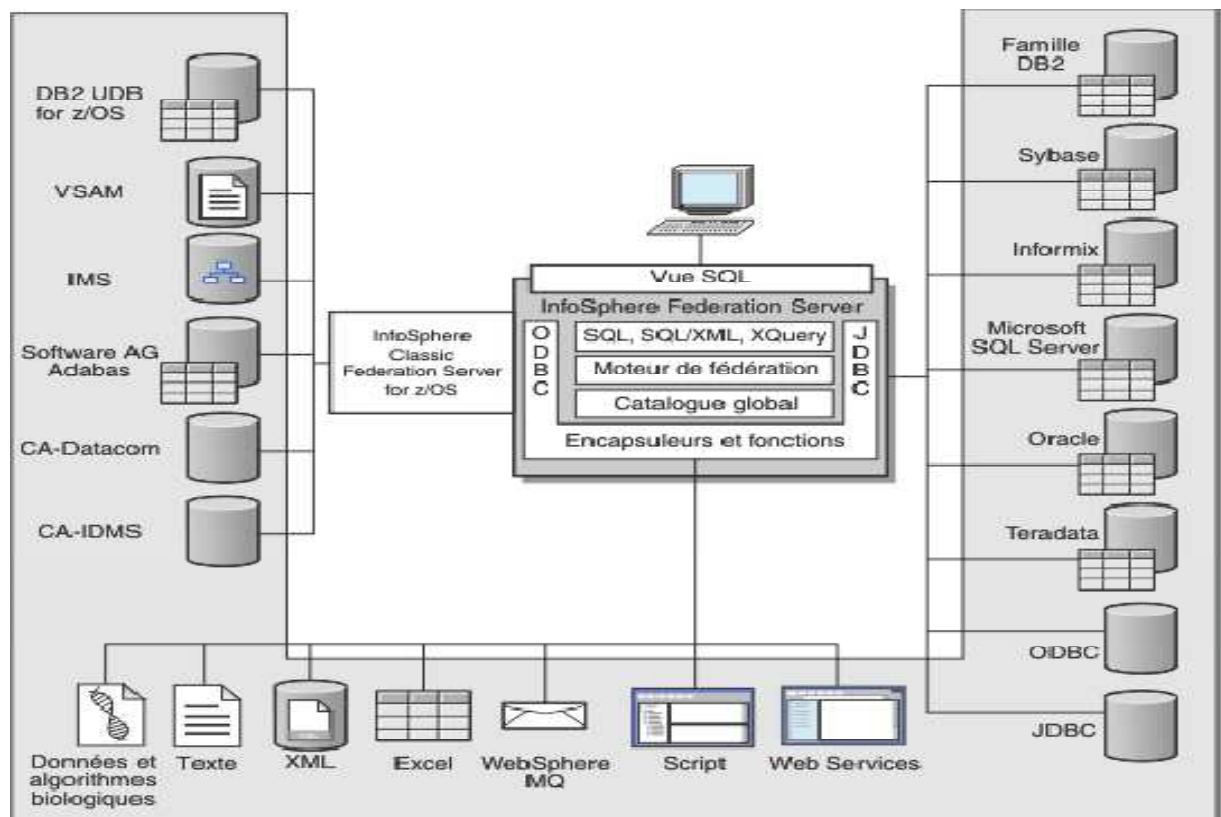


Figure II.11 : SGBD DB2

✓ Les sources de données :

Dans le tableau suivant nous allons résumer les différentes sources de données prise en charge par IBM InfoSphere Federation Server ainsi les logiciel client associés à chacun deux et les systèmes d'exploitations sur les quel ils sont exécutés :

Source de données	Version s prises en charge	Logiciel client	Système d'exploitation et architecture matérielle 32 bits	
			Linux, RedHat Enterprise Linux(RHEL), SUSE	Windows
BioRS	5.2, 5.3	Aucune	Y	y
DB2 for Linux, UNIX et Windows	8.1. x, 8.2.x, 9.1, 9.5, 9.7	Aucune	Y	y

DB2 for z/OS	7.x, 8.x, 9.x	DB2 Connect V9.7	Y	y
DB2 for System i	5.2, 5.3, 5.4, 6.1	DB2 Connect V9.7	Y	y
DB2 Server for VSE et VM	7.2 , 7.4	DB2 Connect V9.7	Y	y
Fichiers à plat		Aucune	Y	y
Informi x	Informi x XPS 8.50, 8.51 et Informi x IDS IDS 7.31, IDS 9.40, IDS 10.0, 11.5, 11.10	Informix Client SDK version 2.81.TC2 ou supérieure ; 3.0 requis pour SLES 10 sous Power	Y	y
JDBC	3.0 ou version supérieure	Pilotes JDBC compatibles avec JDBC 3.0 ou version supérieure	Y	y
Micros oft Excel	2000, 2002, 2003, 2007	Aucune		y
Micros oft SQL Server	Microso ft SQL Server 2000/SP 4, 2005,	Pour Windows, piloteMicrosoft SQL Server Client ODBC 3.0 (ou versionsupérieure). Pour Unix, DataDirect ODBC 5.3	Y	y

	2008			
MQ	MQ7	MQ7	Y	y
ODBC	3.0	Pilotes ODBC compatibles avec ODBC 3.0, ou versionsupérieure	Y	y
OLE DB	2.7, 2.8	OLE DB 2.0 ou versionsupérieure	Y	y
Oracle	10g, 10gR2, 11g, 11gR1	Client Oracle NET 10.0 -10.1, 10.2.0.1 avec module de correction 3807408, 10.1.0.3avec module de correction 3807408, 11.1.0.6.0	Y	y
Sybase	Sybase ASE 12.5, 15.0	Sybase Open Client 12.5 -15.0	y	y
Teradata	Teradata	Composants communs partagés pour l'internationalisation de Teradata (tdicu) version 1.01 ou supérieure, Teradata Generic Security Services (TeraGSS) version 6.01 ou supérieure et le logiciel sur les systèmes d'exploitation suivants : Pour Windows client Teradata TTU 8.0 ou version supérieure ou bibliothèque d'API Teradata API CLIV2 4.8.0 ou version supérieure Pour UNIX et Linux Teradata Call-Level Interface Version 2 CLIV2 Edition 4.8.0 ou supérieure	Y	y
Web Services	WSDL 1.0, 1.1 SOAP 1.0, 1.1	Aucune	Y	y
XML	XML1.0, XML1.1	Aucune	Y	y

Tableau : Sources de données prises en charge, configurations logicielles client requises et prise en charge à partir des systèmes d'exploitation 32 bits

Chaque source de données citée dans le tableau nécessite l'installation d'un logiciel client pour accéder à ces dernières. Ces logiciels doivent être installés sur le même système que IBM InfoSphere Federation Server. D'autre part il est nécessaire de disposer d'une SDK Java appropriée pour utiliser certains outils comme DB2 Control Center ainsi que pour créer et exécuter des applications Java, notamment des procédures mémorisées et des fonctions définies par l'utilisateur.

✓ **Le catalogue système :**

La base de données fédérée contient un catalogue système qui stocke des informations sur les données. Le catalogue système de la base de données fédérée contient des entrées qui identifient les sources de données et leurs caractéristiques. Le serveur fédéré consulte les informations stockées dans le catalogue système de la base de données fédérée et l'encapsuleur de source de données afin de déterminer le meilleur plan pour traiter les instructions SQL.

✓ **Le serveur fédéré :**

Comme tout autre serveur d'applications, le serveur fédéré est une instance du gestionnaire de bases de données. Les processus applicatifs connectent et soumettent les requêtes à la base de données au sein du serveur fédéré. Cependant, deux caractéristiques principales le distinguent des autres serveurs d'applications :

- Le serveur fédéré reçoit des requêtes susceptibles d'être partiellement ou entièrement destinées à des sources de données différentes ; dont le serveur fédéré occupe de la répartition de ces requêtes vers les destinations concernées
- Pour la communication avec les instances de la famille DB2 le serveur fédéré utilise les protocoles de la pile TCP/IP. Cependant, contrairement à d'autres serveurs d'applications, un serveur fédéré utilise le client natif de la source de données pour accéder à la source de données. Par exemple, un serveur fédéré utilise Sybase Open Client pour accéder aux sources de données Sybase et un pilote ODBC Microsoft SQL Server pour accéder aux sources de données Microsoft SQL Server.

✓ **Encapsuleurs et modules d'encapsuleur :**

Les *encapsuleurs* sont des mécanismes qui permettent à la base de données fédérée d'interagir avec des sources de données. Pour l'implémentation de ces derniers la base de données fédérée utilise des routines stockées dans une bibliothèque appelée *module d'encapsuleur*.

Généralement, le propriétaire d'instance fédérée utilise l'instruction CREATE WRAPPER pour enregistrer un encapsuleur dans la base de données fédérée, l'enregistrement pourra être isolé ou sécurisé en utilisant l'option d'encapsuleur DB2_FENCED

Pour chaque type de source de données auquel nous voulons accéder on doit créer un encapsuleur par exemple : *accéder à trois tables de base de données DB2 for z/OS, à une table DB2 for System i, à deux tables Informix et à une vue Informix. Dans ce cas, on doit*

créer un encapsuleur pour les objets de source de données DB2 et un autre pour les objets de source de données Informix.

➤ **Les tâches effectuées par l'encapsuleur :**

De nombreuses tâches sont effectuées par l'encapsuleur, par exemple :

- ❖ Il effectue la connexion à la source de données. Il utilise l'API de connexion standard de la source de données.
- ❖ Il soumet des requêtes à la source de données (si les sources de données prennent en charge le langage SQL, la requête est soumise en SQL. Si non, la requête est convertie en langage de requête natif de la source de données ou en une série d'appels d'API source)
- ❖ Il reçoit les ensembles de résultats provenant de la source de données. L'encapsuleur utilise les API standard source pour la réception de l'ensemble de résultats.
- ❖ Il répond aux requêtes de la base de données fédérée

II.9 Caractéristiques de la solution fédérée d'IBM :

- **Transparence:**

Le système fédéré masque à l'utilisateur les différences, l'hétérogénéité et les mises en œuvre des sources de données sous-jacentes. Idéalement, il rend l'ensemble de sources fédéré en vu de l'utilisateur comme un système unique (simple). c'est-à-dire l'utilisateur ne devrait pas être conscient de l'emplacement où les données sont stockées (transparence d'emplacement), quel langage ou interface de programmation sont supportés par la source de données (transparence d'invocation), comment les données sont physiquement stockées (l'indépendance des données physique, la fragmentation et la transparence de reproduction), et quels protocoles réseau sont utilisés (transparence de réseau). En résumé l'utilisateur devrait voir une interface simple unique et complète.

- **Hétérogénéité :**

Les sources de données peuvent différer de plusieurs façons : ils peuvent fonctionner sur du matériel différent, utiliser des protocoles de réseau différents et avoir des logiciels différents pour gérer leurs bases de données, ils peuvent aussi avoir des langages d'interrogation différents, et avoir des schémas de données différents et fournir une sémantique de transaction différent. Mais un système fédéré prend en charge le problème d'hétérogénéité des sources quelque soit le degré de différenciation grâce au système d'encapsulation et ces modules d'encapsuleurs.

- **Extensibilité et ouverture de la fédération :**

Au fil de temps tout système doit se développer dont un système fédéré de nouvelles sources peuvent être nécessaire pour respecter les besoins des utilisateurs, IBM facilite l'ajout de nouvelles sources. Ces dernières pouvant accéder à la base de données fédérée via un composant logiciel qui est l'encapsuleur, cette architecture d'emballage permet aux sources d'être dynamiquement ajoutée à la fédération sans arrêter son fonctionnement en cours ou

des transactions.

- *Autonomie des sources de donnée :*

Chaque source de données a des applications locales et ces propres utilisateurs, la fédération de la base de données ne dérange pas le fonctionnement des applications locales existantes c'est-à-dire que les données restent inchangées, elles sont ni déplacées ni modifiées et les interfaces restent les mêmes d'autre part l'exécution des requêtes globale (les requêtes du système fédéré) l'influencent pas sur l'exécution des requêtes locales. Donc en résumé il n'y a aucun impact sur la cohérence du système locale lorsqu'une source entre ou quitte une fédération.

II.10 SGBD Open source MySQL: [16]

II.10.1Présentation de MySQL

MySQL est donc un Système de Gestion de Bases de Données Relationnelles, qui utilise le langage SQL. C'est un des SGBDR les plus utilisés. Sa popularité est due en grande partie au fait qu'il s'agit d'un logiciel Open Source, ce qui signifie que son code source est librement disponible et que quiconque qui en ressent l'envie et/ou le besoin peut modifier MySQL pour l'améliorer ou l'adapter à ses besoins. Une version gratuite de MySQL est par conséquent disponible. À noter qu'une version commerciale payante existe également.

II.10.2Avantages et inconvénients de MySQL :

- ✓ *Avantage :*

- Solution très courante en hébergement public
- Très bonne intégration dans l'environnement Apache/PHP
- Open Source, bien que les critères de licence soient de plus en plus difficiles à supporter
- Version cluster depuis la version 4
- ordonnanceur dès la version 5.1
- Partitionnement dès la version 5.1
- Facilité de déploiement et de prise en main.
- Plusieurs moteurs de stockage adaptés aux différentes problématiques, configurable au niveau table.

- ✓ *Inconvénients :*

- Ne supporte qu'une faible partie des standards SQL-92
- Support incomplet des triggers et procédures stockées
- Gestion des transactions avec les moteurs Falcon ou InnoDB uniquement
- Assez peu de richesse fonctionnelle
- Manque de robustesse avec de fortes volumétries
- Pas d'héritage de table
- Pas de vue matérialisée

II.10.3MySQL et la répartition des bases de données :

La fonctionnalité de partitionnement MySQL permet aux développeurs et aux administrateurs de bases de données d'améliorer les performances des bases de données et de simplifier la gestion de très grandes bases de données. MySQL autorise un partitionnement horizontal dans lequel les lignes d'une base de données sont divisées en plus petits ensembles de données, puis distribuées dans plusieurs répertoires et sur plusieurs disques.

Le partitionnement augmente les performances des requêtes puisque des ensembles de données plus petits ont uniquement besoin d'un accès à des opérations spécifiques, et non à une table unique de grande taille. Il est également possible de répartir une table partitionnée sur des lecteurs physiques différents, ce qui permet de réduire la contention sur les interfaces d'E/S physiques en cas d'accès simultané à plusieurs partitions.

Commençons par voir le partitionnement horizontal, qui pour rappel, consiste à répartir les enregistrements d'une table dans différentes partitions. Pour déterminer à quelle partition appartient chaque enregistrement, il faut définir un critère de partitionnement. Et selon ce dernier, MySQL propose quatre types de partitionnement horizontal.

1. **Partitionnement par intervalles** (*Range partitioning*) : les enregistrements sont répartis en fonction de la valeur d'une colonne, par rapport à un ensemble d'intervalles définissant le domaine d'appartenance de chaque partition.

Exemple de partitionnement par intervalle

Soit la table *Employe* (*id, nom, prenom, service, salaire*) et un partitionnement en fonction de la valeur du salaire. Nous décidons de définir trois partitions :

Partition 1 : salaire < 1500

Partition 2 : 1500 <= salaire < 3000

Partition 3 : salaire >= 3000

donc les enregistrements seront répartis ainsi :

Partition 3 : (1, 'Dupond', 'Benard', 'Administratif', 3100)

Partition 1 : (2, 'Smith', 'John', 'Technique', 1400)

Partition 2 : (3, 'Lucaza', 'Sophie', 'RHC', 1800)

2. **Partitionnement par listes** (*List partitioning*) : même principe que le partitionnement par intervalles, mais la partition d'un enregistrement est déterminée à partir de listes de valeurs.
3. **Partitionnement par hachage** (*Hash partitioning*) : la partition à laquelle appartient un enregistrement est déterminée à partir de la valeur de retour d'une fonction définie par l'utilisateur. Cette fonction de hachage doit donc opérer un traitement sur la valeur d'une ou plusieurs colonnes des enregistrements.

4. **Partitionnement par clé** (*Key partitioning*) : cette méthode de partitionnement est similaire au partitionnement par hachage, à l'exception que la fonction de hachage est fournie par le serveur MySQL.

Nous venons de voir les techniques et les types de partitionnement horizontal. En gardant à l'esprit que plus les tables sont petites, plus les requêtes s'exécutent rapidement, nous pouvons en déduire qu'il est également possible d'optimiser nos requêtes grâce à un partitionnement vertical.

Malheureusement, MySQL 5.1 ne gère pas automatiquement le partitionnement vertical, comme il le fait pour l'horizontal. Cependant nous pouvons nous-mêmes simuler des partitions à partir de tables et pour les gérer au mieux, nous pouvons utiliser deux fonctionnalités apparues dans MySQL 5 : les **vues** et les **procédures stockées** se que nous allons détailler dans le chapitre suivant en présentant les techniques d'intégration de données.

Conclusion :

Les bases de données distribuées constituent un domaine important pour la gestion des informations stockées sur différents sites.

Dans ce chapitre nous avons présenté les principes des bases de données distribuées et ces différentes techniques .on a vu on premier lieu les techniques de répartition des BDD qui peut se faire selon différentes scénarios choisi par le concepteur, et on deuxième lieu on a vu les techniques de fédération de différents base de données stockées sur plusieurs sites.

Nous avons présenté également quelque SGBD qui réalisent la distribution des bases de données citons DB2 d'IBM pour la fédération et le Open source MySQL.

Chapitre III
Intégration de
sources de
données

Introduction :

L'intégration de sources de données devient un défi pour la communauté des bases de données, dont aujourd'hui, il est devenu un domaine de recherche très important. Les solutions proposées à cette problématique visent à faciliter l'accès et l'interrogation de ces multiples sources qui sont autonomes et réparties sur plusieurs sites géographiquement éloignées, dans la majorité des cas. A cet effet un système d'intégration de données doit gérer et donner aux utilisateurs une vue uniforme de ces sources, et permettre de les interroger d'une manière transparente, alors qu'il s'agit de données distribuées sur plusieurs sites.

Un système d'intégration de données est composé d'un triplet $I : \langle G, S, M \rangle$, où :

- « **G** » : représente le schéma global (défini sur un alphabet A_G) qui modélise le schéma intégré
- « **S** » : est l'ensemble des schémas des sources (définis sur un alphabet A_S) décrivant la structure des sources participantes au processus d'intégration.
- « **M** » : est une correspondance entre G et S qui établit la connexion entre les éléments du schéma globale et ceux des sources.

Pour interroger le système intégré, les requêtes sont exprimées en termes de constructions du schéma globale G . par raisonnement, tout système d'intégration de données doit considérer l'intégration à deux niveaux :

(1) : le niveau schéma, qui consiste à consolider tous les schémas des sources en un seul schéma global, ou schéma de médiation, qui sera utilisé pour supporter les requêtes

(2) : le niveau donné (population du schéma intégré)

Dans ce qui suit de ce chapitre nous allons présenter les concepts de base des vues, et aussi des vues matérialisées qui ont un rôle important dans le processus d'intégration, par la suite nous allons voir les différentes techniques d'intégration de données.

III.1 Les vues classiques : [19]

La confidentialité des informations dans une base de données est un aspect important qu'un SGBD doit prendre en compte. Plusieurs SGBD tel que MySQL depuis sa version 5, Oracle, etc.... renforcent la confidentialité par l'utilisation des vues (*views*) qui agissent comme des fenêtres sur la base de données.

Les vues correspondent à ce qu'on appelle « le niveau externe » qui reflète la partie visible de la base de données pour chaque utilisateur. Seules les tables contiennent des données et, pourtant, pour l'utilisateur, une vue apparaît comme une table. En théorie, les utilisateurs ne devraient accéder aux informations qu'en questionnant des vues. Ces dernières

masquant la structure des tables interrogées. En pratique, la plupart des applications se passent de ce concept en manipulant directement les tables. La figure suivante illustre ce qui a été dit.

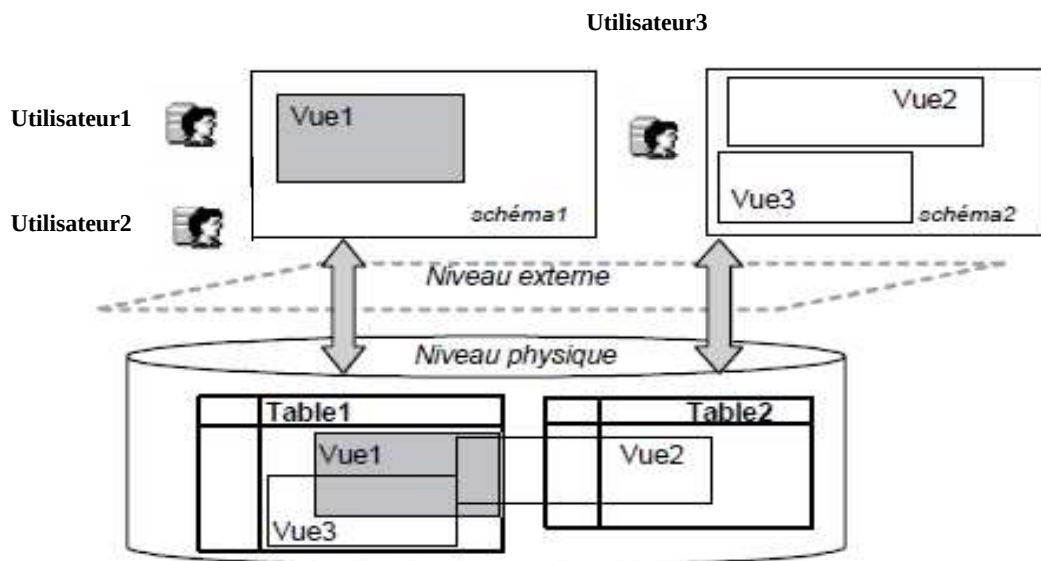


Figure III.1 : Utilisation des vues

III.2 Création d'une vue :

Une vue est créée à l'aide d'une instruction SELECT appelée « requête de définition ». Cette requête interroge une (ou plusieurs) table(s) ou vue(s). Une vue se recharge chaque fois qu'elle est interrogée, donc elle est considérée comme une table virtuelle car elle n'a pas d'existence propre, c'est-à-dire seule sa structure est stockée dans le dictionnaire. Ses données seront extraites de la mémoire à partir des tables source, à la demande.

La figure suivante présente le processus de création et d'utilisation d'une vue, dont le premier schéma montre la création d'une vue à travers une requête SQL qui permet de sélectionner des données d'une ou plusieurs tables.

Le second schéma montre comment un utilisateur final accède à une vue par l'intermédiaire d'une requête. L'utilisateur demande la vue : la requête SQL correspondante est exécutée, les données puisées dans la table de base. La requête renvoie les données à l'utilisateur. Pour celui-ci, la demande d'une vue est identique à celle d'une table.

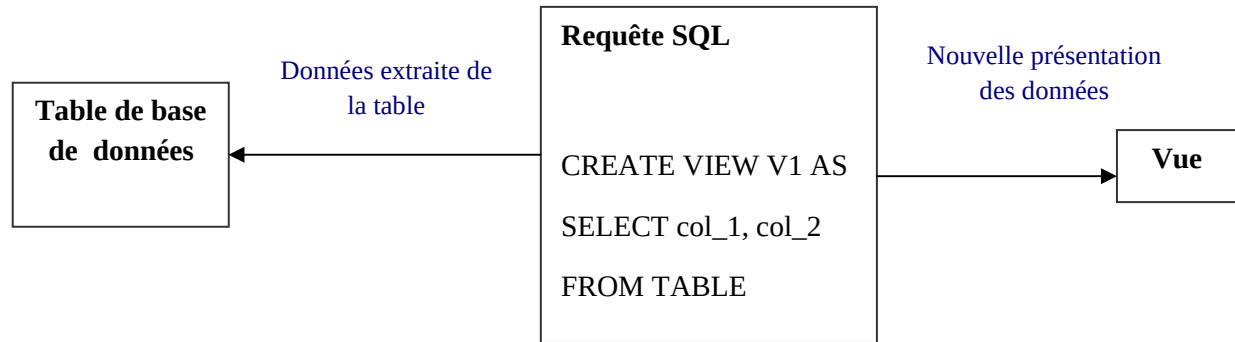


Schéma 1

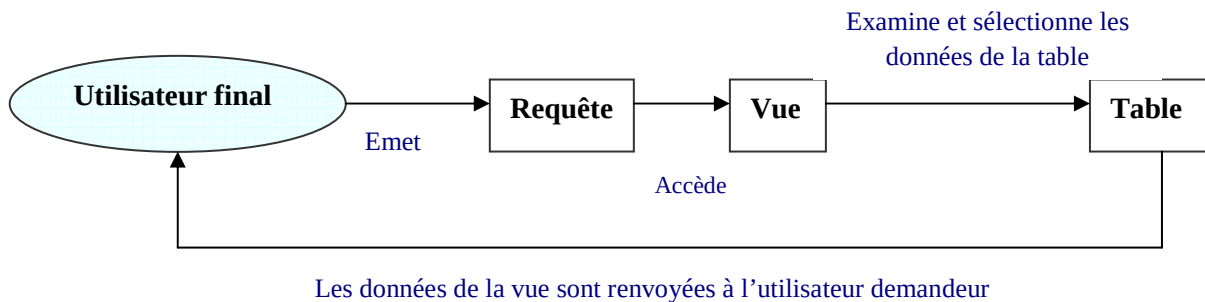


Schéma 2

Figure III.2 : Processus de création et d'utilisation des vues

III.3 Les vues matérialisées : [17]

Afin de combiner les avantages des vues et les avantages des tables normales, il fallait donc créer une nouvelle présentation de données, c'est ainsi que le concept des vues matérialisées est né.

Une vue matérialisée est un objet qui permet de stocker le résultat d'une **requête SELECT**. Là où une vue se contente de stocker la requête, la vue matérialisée va stocker directement les résultats (elle va donc les matérialiser), plutôt que la requête. Lorsque l'on fait une requête sur une vue matérialisée, on va donc chercher directement des données dans celle-ci, sans passer par les tables d'origine et/ou une table temporaire intermédiaire.

III.3.1 Manipulation des vues matérialisées :

III.3.1.1 Oracle :

Oracle Permet de créer directement des vues matérialisées, avec des outils et des options dédiés. Il suffit alors de créer les vues matérialisées que l'on désire et de les paramétrer correctement, et tout se fait automatiquement.

Remarque : Les vues matérialisées peuvent porter sur des tables, mais aussi des vues ou des vues matérialisées.

Exemple :

```
CREATE MATERIALIZED VIEW mz-relation REFRESH COMPLETE
```

```
ON DEMAND
```

```
AS
```

```
SELECT col1, col2 ... FROM ma-table;
```

➤ **Mise à jour des vues matérialisées sous Oracle :[13]**

Afin d'assurer une certaine cohérence des données, il faut mettre à jour les vues matérialisées et les tables périodiquement. Il existe trois façons de mises à jour qui sont la régénération complète, rapide et forcée.

- **Rafraîchissement complet :**

Il va ré-exécuter la requête basée sur la table de base et remplace l'ensemble des données de la VM par les données obtenues et ceci même si la table de base n'a pas été modifiée

- **Rafraîchissement incrémental (rapide) :**

Son principe est de propager uniquement les données modifiées depuis le dernier rafraîchissement

- **Rafraîchissement forcé :**

Dans ce type de rafraîchissement, lorsqu'une régénération rapide n'est pas possible, alors une régénération complète est exécutée.

III.3.1.2 MySQL :

Malheureusement, MySQL n'implémente pas les vues matérialisées en tant que telles. Cependant, on peut utiliser des moyens, c'est-à-dire les objets et concepts disponibles avec MySQL, il est tout à fait possible de construire des vues matérialisées manuellement.

Le principe de la vue matérialisée est de stocker les données d'une requête SELECT la structure qui permet de stocker ces données semblent évidentes est de créer une table.

En résumé Pour créer une vue matérialisée avec MySQL, on utilise donc tout simplement une table, dans laquelle on stocke les résultats d'une requête SELECT.

Exemple :

```
CREATE TABLE VM
ENGINE = InnoDB
SELECT attribut1 attribut2 ...
FROM Table
WHERE condition de selection
```

➤ **Mise à jour des vues matérialisées sous MySQL : [17]**

Pour assurer l'efficacité d'utilisation des vues matérialisées sous MySQL, il ne suffit pas de les créer, mais il faut aussi de les tenir à jour, lorsque les données des tables d'origines changent.

Il y a deux possibilités de mise à jour :

- Une mise à jour sur demande ;
- Une mise à jour automatique chaque fois qu'un changement est fait.

a) Mise à jour sur demande :

Pour mettre la vue matérialisée à jour ponctuellement, on peut utiliser une procédure stockée. Le plus simple sera de supprimer les données de la vue matérialisée, puis d'insérer les données à jour grâce à la même requête de sélection ayant servi à créer ou initialiser la vue matérialisée.

✓ **Exemple de procédure :**

```
CREATE PROCEDURE maj_vm()
BEGIN
  TRUNCATE VM
  INSERT INTO VM
  SELECT attribut1, attribut2...
  FROM table1, table2 WHERE "condition de selection"
END
```

b) Mise à jour automatique :

Pour que les données soient toujours à jour par rapport aux derniers changements de la base de données, on utilisera plutôt les triggers (déclencheurs) pour mettre à jour la vue matérialisée.

Les triggers (ou déclencheurs) : sont des objets de la base de données. Attachés à une table, ils vont déclencher l'exécution d'une instruction, ou d'un bloc d'instructions, lorsqu'une, ou plusieurs lignes sont insérées, supprimées ou modifiées dans la table à laquelle ils sont attachés.

Remarque : les instructions du trigger peuvent être exécutées soit juste avant l'exécution de l'événement déclencheur, soit juste après.

Exemple de trigger:

```
CREATE TRIGGER nom trigger moment trigger evenement_trigger
ON nom table FOR EACH ROW
Corps trigger
```

Après avoir présenté et expliqué les notions de bases des vues et des vues matérialisées ainsi leurs mise à jour, nous allons voir dans ce qui suit, leur utilisation dans le système d'intégration des données, en présentant ces différentes approches.

III.3.2 Avantage des vues matérialisées :

- ✓ Permettent de gagner en performances
- ✓ Permettre ou interdire à un groupe d'utilisateurs de voir certaines données.
- ✓ Eviter au serveur de faire certains traitements lourds et répétitifs ce qui implique un gain de temps.
- ✓ Disponibilité de données.
- ✓ Intégrité de données.

III.4 Système d'intégration de données : [13]

Un système d'intégration de données fournit une vue unifiée de données provenant de sources multiples. Il permet d'accéder à ces données au travers d'une interface, sans se soucier de leur structure ni de leur localisation. La figure suivante illustre les composants d'un système d'intégration.

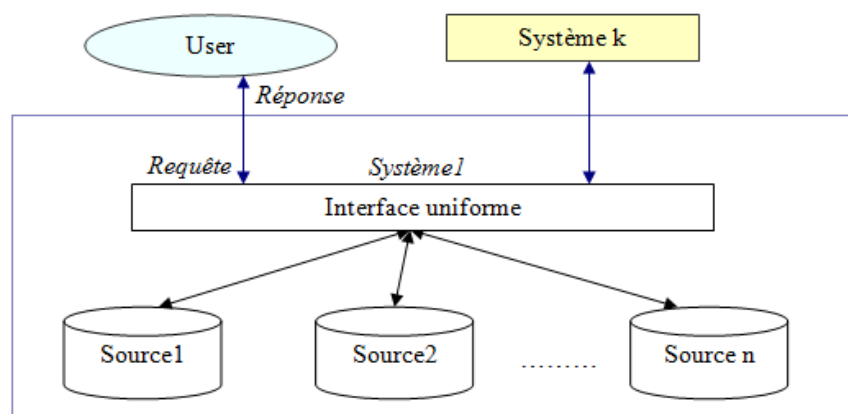


Figure III.3 : Système d'intégration de données

Comme nous l'avons vu dans la figure ci-dessous le système se compose de trois parties :

- ✓ Une partie externe qui correspond aux utilisateurs du système intégré, ou autres systèmes.
- ✓ Une deuxième partie qui comprend des sources d'information participantes au processus d'intégration
- ✓ une interface uniforme qui permet aux utilisateurs ou autres systèmes (de partie externe) d'interroger d'une manière transparente avec les sources de données, comme s'il n'y avait qu'une source unique.

L'interface peut être définie par l'une des deux façons suivantes :

Soit une couche sans données propres (les données sont virtuelles), soit par une couche qui contient des données c'est-à-dire une duplication matérialisée des données des sources (approche entrepôt).

III.4.1 Classification des systèmes d'intégration de données :

Il existe principalement deux architectures physiques possibles pour les systèmes d'intégration de données, lorsque les données des sources sont stockées dans les systèmes d'intégration on parle d'approche matérialisée (entrepôt de données) en utilisant des vues matérialisées, à l'inverse, lorsque les données intégrées ne sont pas répliquées, on parle d'approche virtuelle (système médiateur).

III.4.1.1 Technique du médiateur : [18]

Cette technique se base sur les vues traditionnelles, dont les données ne sont pas stockées au niveau du médiateur. Elles restent dans les sources de données et ne sont accessibles qu'à ce niveau. C'est pour cette raison que cette approche est appelée approche «virtuelle ».

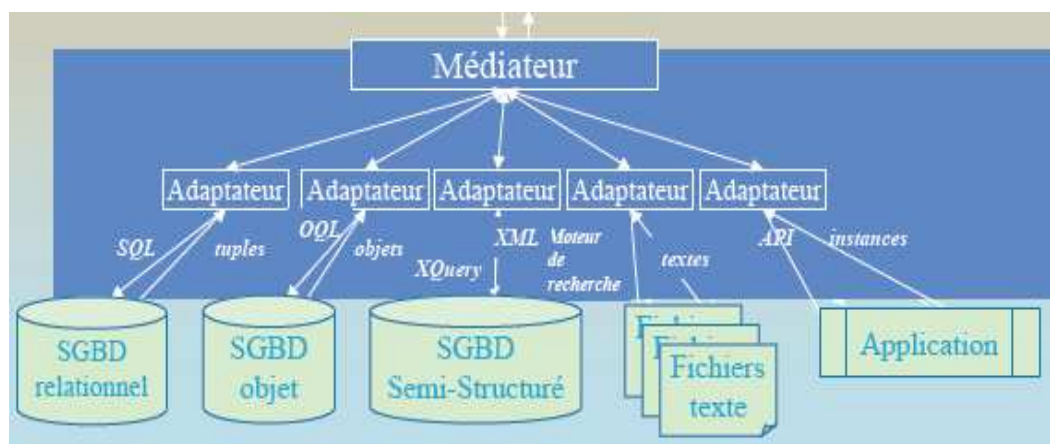


Figure III.4 : Architecture de médiation

L'architecture de médiateur se base sur deux composants essentiels :

- **Le médiateur** : s'occupe de la distribution des données et de leurs localisations, et la décomposition des requêtes en sous requêtes adaptée par chacune des sources(c'est l'approche de Local as View abrégé par LaV), ainsi le traitement réciproque c'est-à-dire la recombinaison des différents résultats provenant de chacune des sources (approcheGaVou Global as View)
- **L'adaptateur** :l'adaptateur (Wrapper) s'occupe de l'hétérogénéité des sources. C'est un traducteur qui fait la traduction de données dans un modèle commun de données dans un sens médiateur /source ou source /médiateur.

III.4.1.2Technique d'entrepôt

L'approche matérialisée consiste à stocker localement des données issues des différentes sources de données dans un **entrepôt** de données en utilisant des vues matérialisées. Pour donner une vue globale des sources à intégrer, les données provenant de différentes sources distribuées doivent être organisées, intégrées et stockées dans l'entrepôt.

➤ ***Qu'est ce qu'un entrepôt de données (Data Warehouse) :[20]***

C'est un nouveau concept de bases de données séparées des systèmes de production, pour le stockage et la consultation des données. La firme IBM a nommé ce nouveau concept de base de données séparée « le dépôt des informations ». Plus tard, Ce concept a été baptisé par *Bill Inmon*« data warehouse »

Cette technique est utilisée par plusieurs entreprises pour avoir et maintenir un avantage compétitif. En rassemblant les informations des sources internes et externes, l'entrepôt moderne est une collection d'informations stockées d'une manière qui améliore l'accès aux données. Une fois les informations rassemblées, quelques types de logiciel de recherche seront utilisés sur les PC pour retirer les données de l'entrepôt où elles sont analysées, manipulées et reportées.

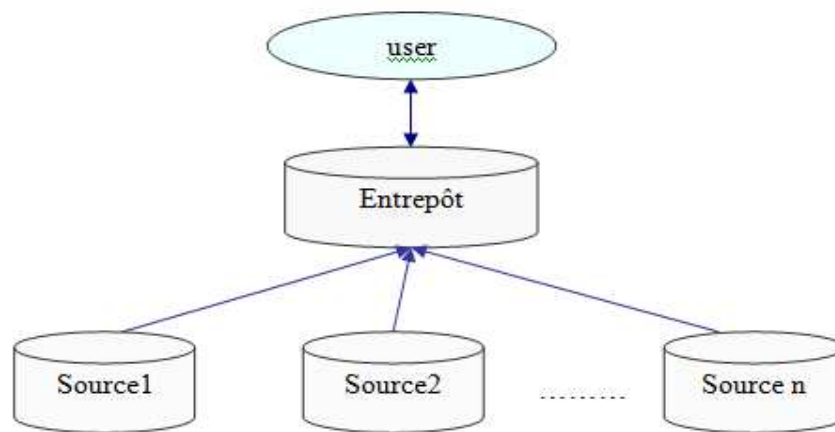


Figure III.5 : Intégration par l'approche de l'entrepôt

III.5 Technique implémenté dans notre application :

Le système d'intégration qu'on va utiliser est un système à base d'entrepôt dans le quel nous allons implémenter les vue matérialisées sous MySQL.

Notre application contient trois base de données sources (BDD de département informatique, BDD de département automatique et la BDD du service budget et comptabilité « service commun de la faculté ») que nous allons les intégrés vers une base de données globale (entrepôt) « base de données de la faculté », dont l'objectif est de donner aux utilisateurs une vue globale sur les données stockée sur les sites locaux comme s'ils accèdent à une base de ENT centralisée.

Le schéma suivant illustre la technique d'intégration que nous allons implémenter dans notre application

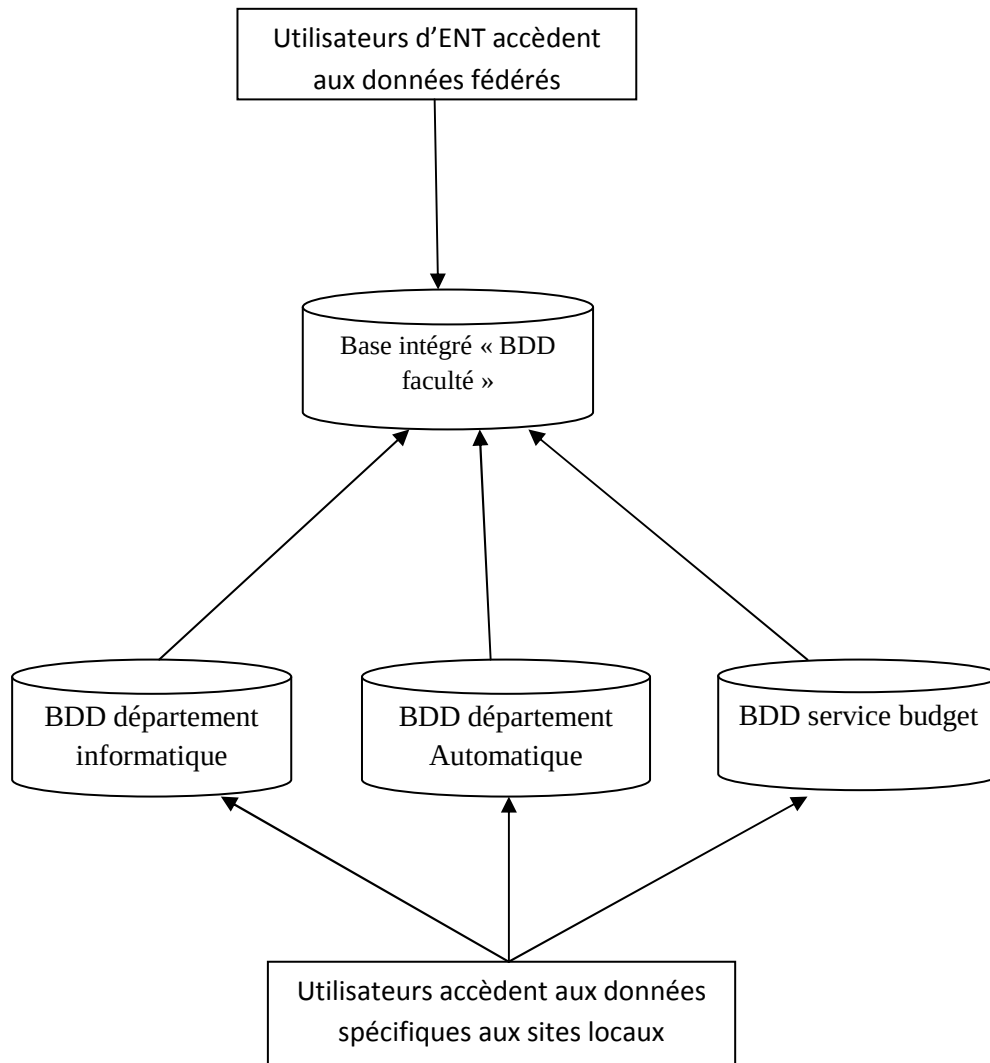


Figure III.6 : Système d'intégration retenu pour notre application

III.5.1 Les utilisateurs :

Nous commençons par la description des utilisateurs de notre système, qui sont classés en deux catégories :

- Les utilisateurs accèdent à la BDD fédérée (c'est-à-dire les données intégrés à partir des sites locaux) qui sont les suivant :
 - Administrateur du campus
 - Administrateur de la faculté
 - Chef de section budget
- Les utilisateurs accèdent aux données spécifiques aux sites locaux :
 - Les enseignants
 - Etudiants

- Le chef de section budget (les données spécifique au service budget)
- Administrateur de département

III.5.2 Les bases de données :

Chaque site local à sa propre base de données local. La base de données fédérée stocke les données intégrées à partir des sites locaux selon les besoins de notre application, c'est-à-dire la base de la faculté contient des tables qui stockent des données sélectionnées à partir des tables des bases des sites locaux où la requête de sélection est exécutée selon nous besoins.

III.5.3 Exemple :

A partir des tables des deux bases des départements informatique et automatique nous créant une vue matérialisé sur une table de la base de la faculté, plus précisément nous sélectionnons des attributs qu'on aura besoin à partir des tables enseignant et Etat de vacation et les stocker dans la table etat_enseignant au niveau de la base faculté (entrepôt).

A l'application de cette procédure le chef de section budget pourra consulter tout les enseignants de la faculté ainsi leurs états de vacation comme si ces données sont stockées sur une base centralisé.

Nous pouvons créer aussi une vue matérialisée qui rassemble tout les étudiants de la faculté à partir des bases des départements informatique et automatique qui sera consulté par l'administrateur de la faculté.

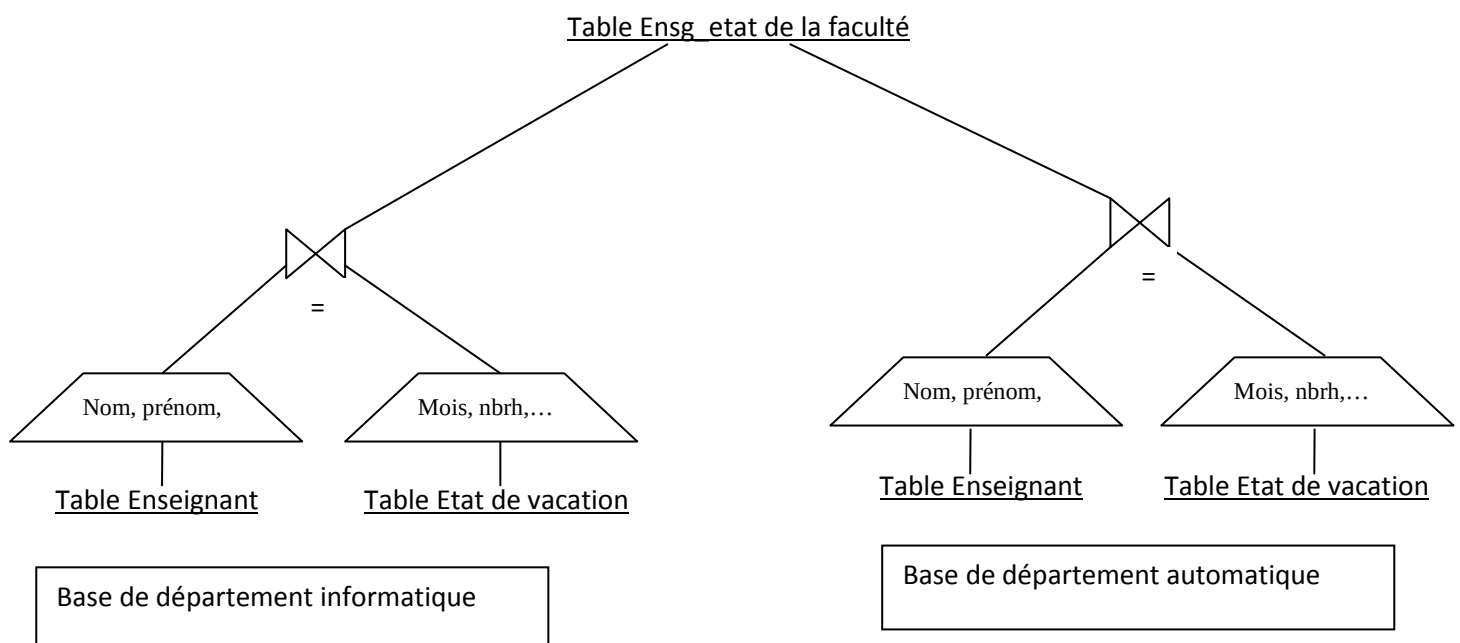


Figure III.7 : Exemple d'intégration de données de notre application

Conclusion :

Dans ce chapitre, nous avons étudié et présenter les concepts de base des vues et la matérialisation de ces derniers ainsi leurs implémentation par différent SGBD. Ensuite nous avons présenté les différents systèmes d'intégration (*entrepôt, médiateur*), et nous avons terminé par la présentation de la solution d'intégration retenue.

Dans le chapitre qui suit nous allons détailler les bases de données que nous allons implémenter par l'analyse et la conception de ces dernières

Chapitre IV :
Analyse et
conception

Introduction :

La phase de conception de toute solution informatique est d'une grande importance et elle doit être traitée avec rigueur et précision, car elle constitue la base du système à développer.

Avant d'entamer la conception, il est impératif de passer par la phase d'analyse qui permet d'identifier les différents acteurs qui interagissent avec le système ainsi que leurs besoins. Puis on passe à la conception qui, en s'appuyant sur les résultats de la phase d'analyse, donnera la description détaillée du système cible et des objectifs à atteindre.

Pour ce faire, notre démarche va s'appuyer sur le langage UML étendu pour leweb, qui permet une bonne représentation des aspects, statique et dynamique,d'une application.

IV.1 Proposition organisationnelle :**IV.1.1 Proposition organisationnelle de la plateforme du campus :**

L'ENT est destinée à la communauté universitaire UMMTO (étudiants, enseignants, personnels administratifs...), donc il présentera un ensemble d'informations de divers domaines que ce soit pédagogique ou administratif. Pour cela l'organisation du campus doit être efficace pour répondre à toute demande d'information et de facilité sa transmission.

L'organisation que nous allons proposer suit l'organisation hiérarchisée de l'université dont elle est structurée comme suit :

- **Rectorat** : recteur, les vices recteurs, le secrétariat général, la bibliothèque centrale.
- **Les facultés** : faculté des sciences, faculté de génie électrique et informatique, faculté de génie de construction, faculté de droit...etc.

Chaque faculté contient des départements, des services communs (service personnel, service budget et comptabilité, service moyens et maintenance..), du personnels administratifs (chef de département, chef de service..) personnels techniques, des enseignants et des étudiants.

La figure suivante présente l'organigramme et la structure de navigation dans le campus :

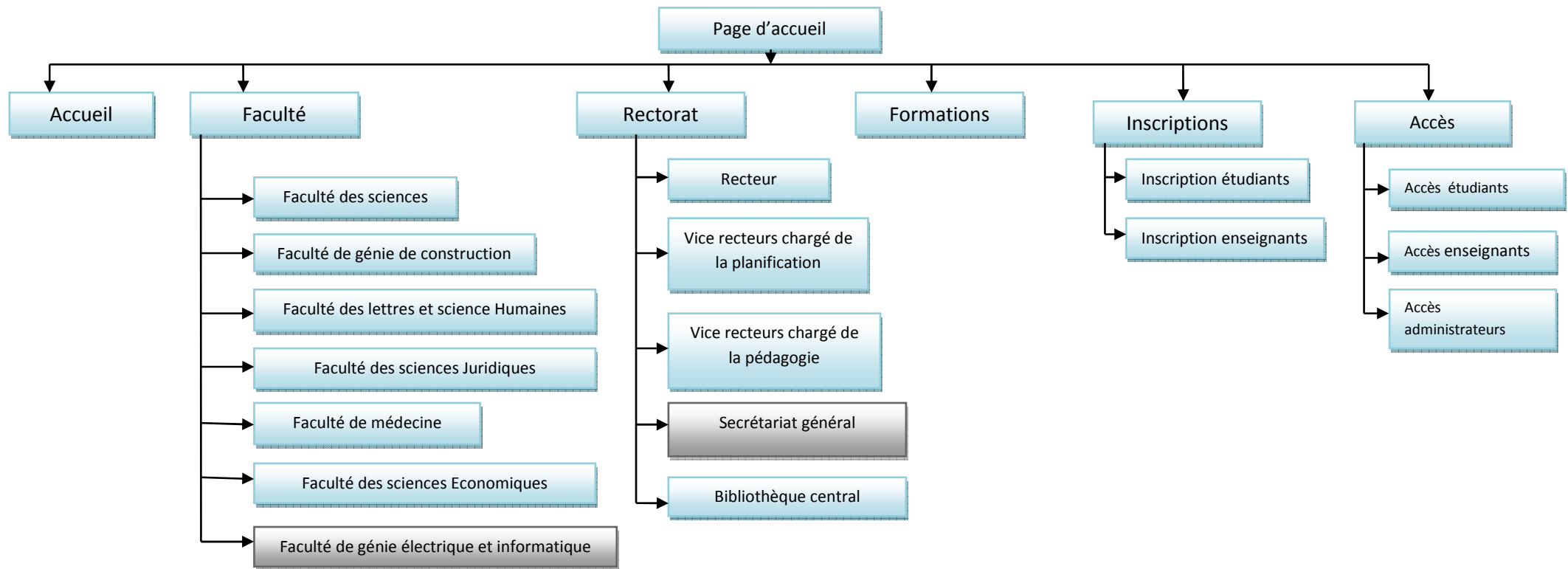


Figure IV .1: Organigramme de la plateforme d'ENT d'UMMTO

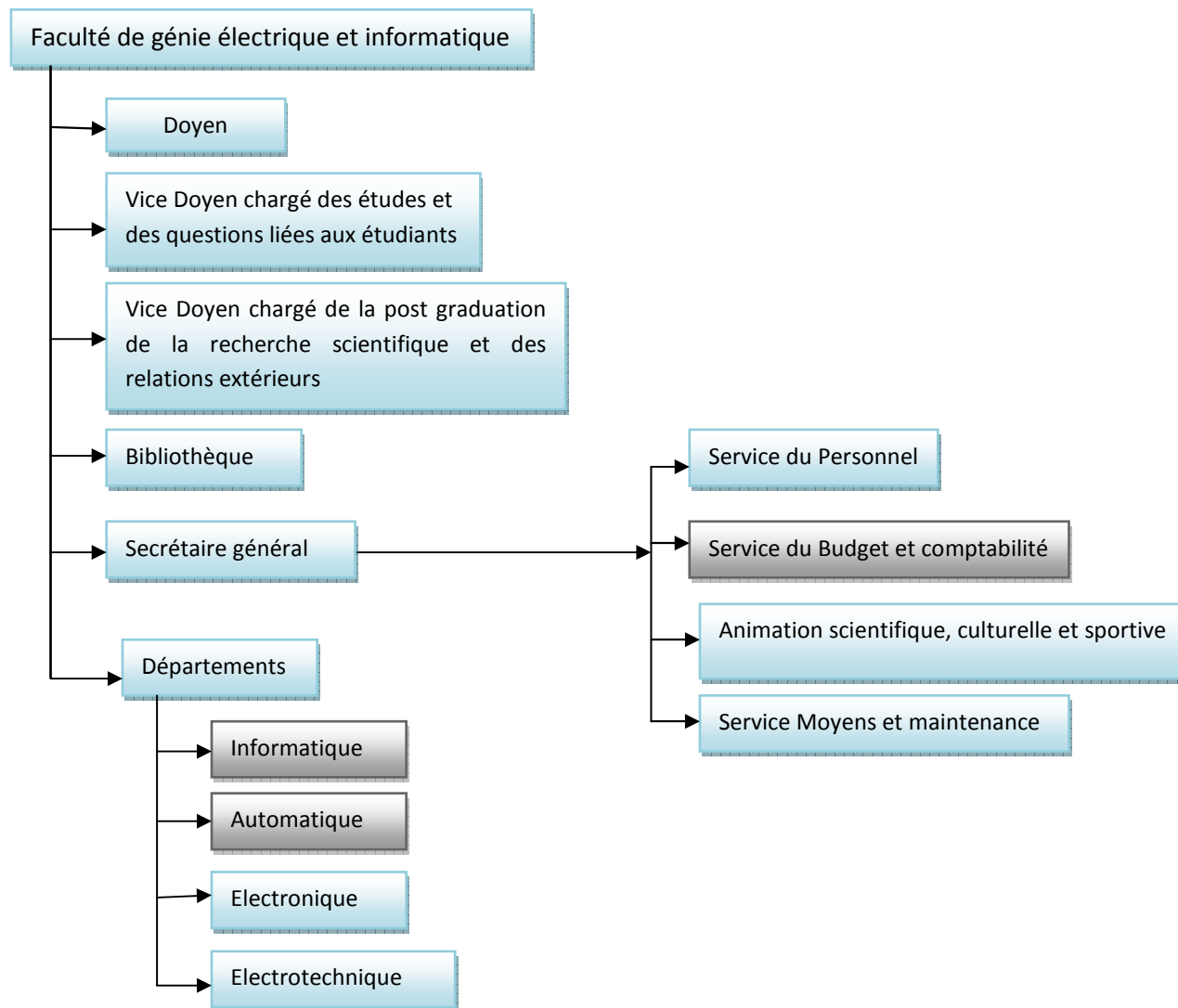


Figure IV.2 : Organigramme du module faculté « cas faculté de génie électrique et informatique »

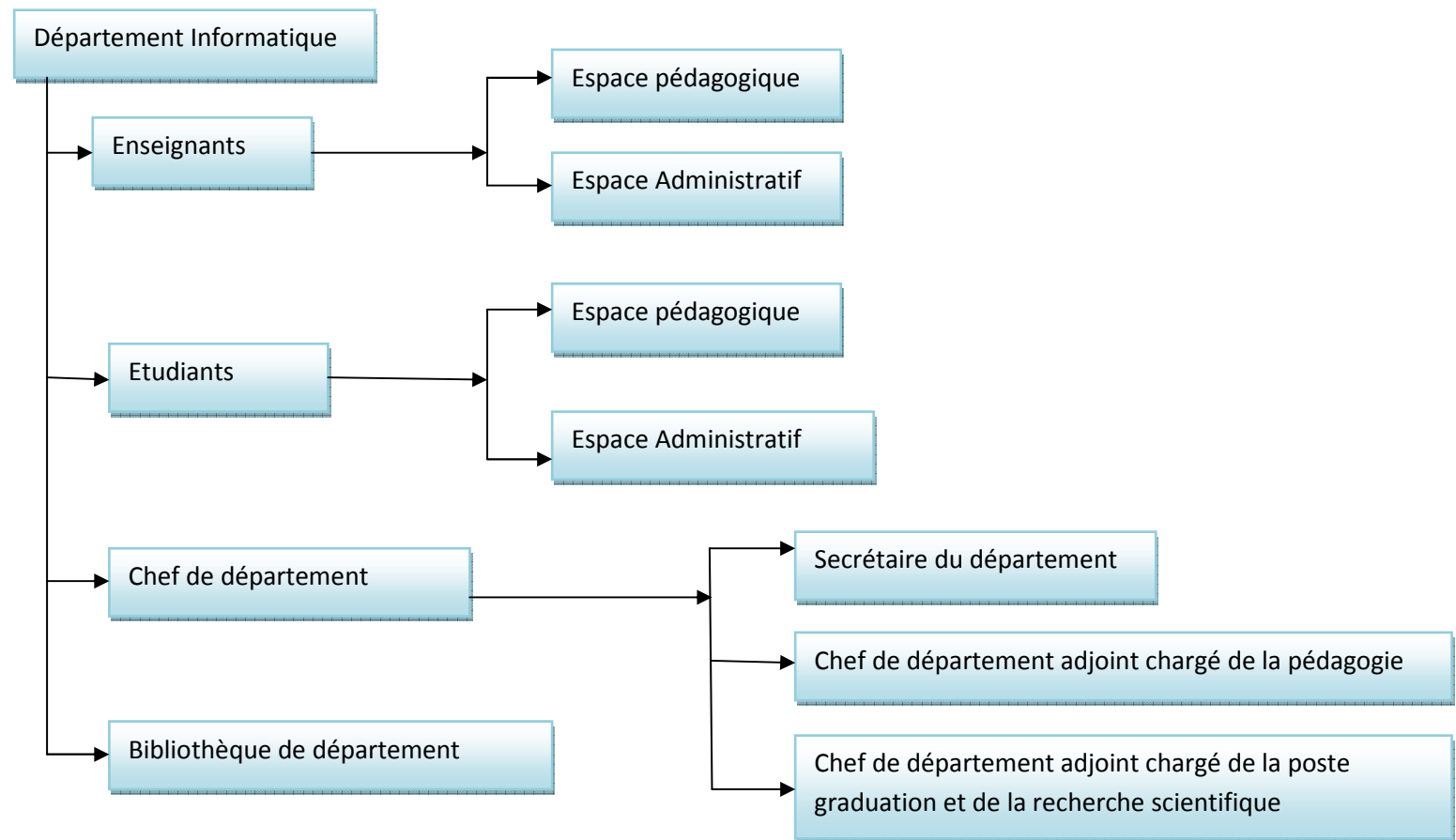


Figure IV.3: Organigramme du module département « Cas département d'informatique »

IV.1.2 Proposition organisationnelle des données :

Les données manipulées par les différents acteurs du campus peuvent être présentées par la formule suivante :

$$\sum \mathbf{DC} = \sum \mathbf{DSD} + \sum \mathbf{DSF} + \sum \mathbf{DSSC} + \sum \mathbf{DP}$$

DC : les données du campus

DSD : les données spécifiques à chaque département

DSF : les données spécifiques à chaque faculté

DSSC : les données spécifiques aux services communs

DP : les données partagées

Dans notre cas d'étude, nous nous limitons à l'étude d'une seule faculté qui est la faculté de génie électrique et informatique, ou nous allons nous baser sur deux départements «département informatique » et « département d'automatique » et un service commun à la faculté « service budget et comptabilité », donc notre formule sera présentée comme suit :

$$\sum \mathbf{DF} = \sum \mathbf{DSD.Info} + \sum \mathbf{DSD.Aut} + \sum \mathbf{DS.FGEI} + \sum \mathbf{DSSC.budg} + \sum \mathbf{DP}$$

DF : Les données de la faculté.

DSD.Info : Les données spécifiques au département informatique.

DSD.Aut : Les données spécifiques au département automatique

DS.FGEI : Les données spécifiques à la faculté génie électrique et d'informatique.

DSSC.budg : Les données spécifiques au service budget et comptabilité.

DP : c'est les données partagées entre les unités cités précédemment.

➤ Localisation des données :

Chacune des données spécifiques sont localisées sur des sites locaux comme suit :

- ✓ **DSD.Info** : Sont localisées dans la base de données du site département informatique.
- ✓ **DSD.Aut** : Sont localisées dans la base de données du site département d'automatique.

- ✓ **DSSC.budg** : Sont localisées dans la base de données du site service budget et comptabilité
- ✓ **DS.FGEI** : sont localisées dans la base de données faculté.

Remarque : Le symbole Σ désigne l'ensemble des données.

Par contre les données partagées sont rassemblées dans la base faculté (entrepôt) sous forme de vues matérialisées comme nous l'avons détaillé dans les chapitres précédents.

La figure ci-dessous, illustre la structure générale de la solution proposée pour la faculté de génie éclectique et informatique, dont nous avons englobé tous les départements et services de la faculté.

Mais notre cas d'étude se limite sur deux départements seulement qui sont (informatique et automatique) et un service commun qui est le service de budget et comptabilité

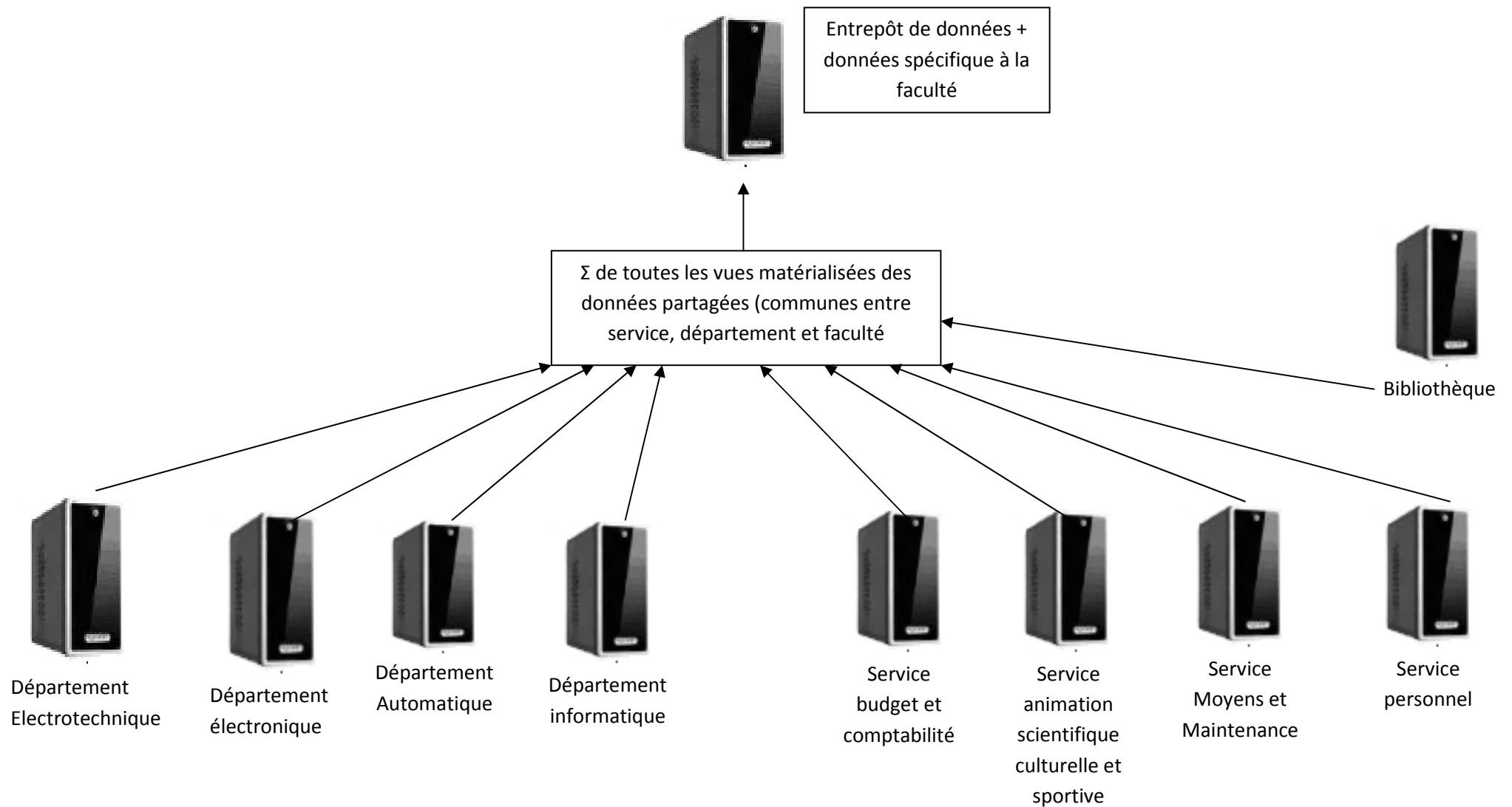


Figure IV.4 Architecture globale de la proposition organisationnelle des données au niveau de la faculté GEI

IV.2Analyse :

Cette phase comprend l'identification des besoins du système, Les acteurs participent, leurs interactions avec le système ainsi que les cas d'utilisation.

IV.2.1 Spécification des besoins :

Notre projet consiste à concevoir et améliorer la base de données du prototype du campus numérique réalisé par les promotions précédentes (« *conception générale et réalisation d'un prototype d'un campus numérique pour UMMTO* » promotion 2011/2012) par son implémentation et son extension d'une base de données centralisée vers une base de données distribuées sur plusieurs sites, autrement dit sur les différents départements incluant les facultés de l'université ou plus précisément la fédération des bases de données existantes des départements et facultés (sites locaux) pour avoir une base de données globale de l'université.

Notre cas d'étude s'appuie sur un exemple de faculté qui est dans notre cas la faculté de génie électrique et d'informatique, ou nous avons choisie d'intégrer les données de deux départements qui sont le département d'informatique et automatique et un service commun de la faculté.

La problématique réside au niveau des données partagées entre les départements et le service commun et comment faire circuler ce flux d'information d'une manière transparente en vue des acteurs de ENT et comment gérer les données spécifiques à chacun deux.

L'objectif de notre projet est de proposer une bonne solution à la problématique posée précédemment par une bonne conception et intégration d'une base de données fédérée, qui va être utilisée par l'espace numérique de travail d'UMMTO.

Donc pour la conception de la BDD de ENT nous allons procéder d'une façon ascendante on suivant les étapes suivantes :

- Etudier les besoins et les acteurs des sites locaux et établissement des schémas conceptuels locaux.
- Etablissement du schéma d'intégration on utilisant l'approche que nous avons détaillé dans les chapitres précédents.
- Réalisation du schéma globale après intégration (fédération)

- Traduction des schémas conceptuels obtenus en schémas relationnels logiques en appliquant les règles de passage UML-relationnel.

IV.2.2 Identification des acteurs :

Comme nous avons cité précédemment notre système comporte plusieurs sites (sites locaux et un site global) donc nous allons identifier les acteurs de chaque site comme le montre cette figure :

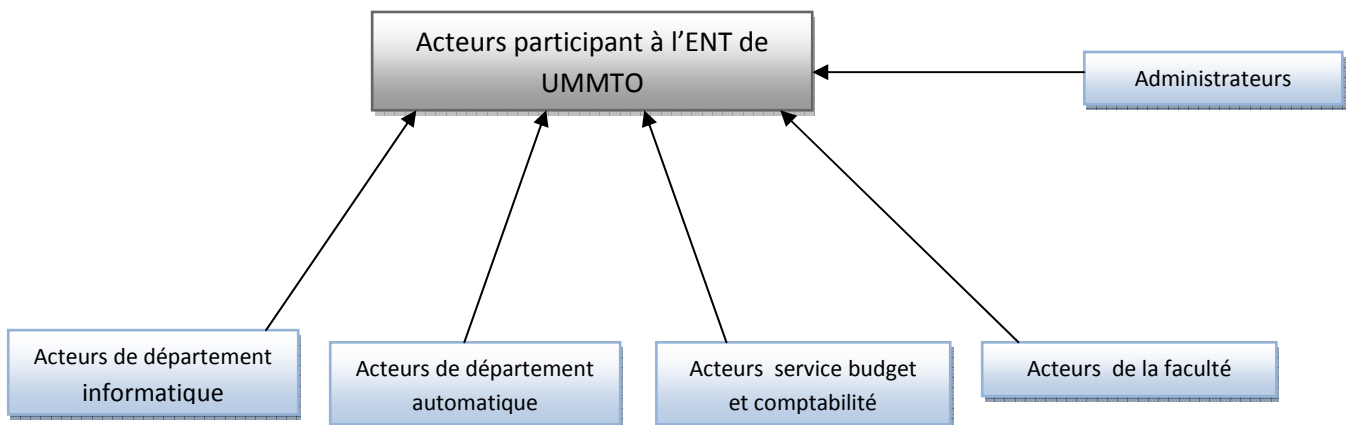


Figure IV.5 : Les acteurs participant à notre système

IV.2.2.1 Identification des acteurs de département informatique :

Les acteurs d'un département sont nombreux et ils sont tous concernés par l'espace numérique mais dans notre cas on se limite aux acteurs suivant :

- **Enseignant** : est un acteur qui peut accéder à l'espace campus pour avoir les informations et les applications spécifiques au département informatique et aussi à l'espace processus métier.
- **Etudiant** : est un acteur concerné juste par les applications et les données spécifiques au département
- **L'administrateur d'e-département** : c'est l'administrateur de site département.

Remarque : les acteurs de département automatique sont identiques aux acteurs de département informatique.

IV.2.2.2 Identification des acteurs du service budget et comptabilité :

Dans ce site nous avons aussi deux acteurs principaux qui sont les suivants :

- Chef de section budget
- L'administrateur d'e-service communs

Et un acteur secondaire :

- Agent administratif du service budget et comptabilité.

IV.2.2.3 Administrateurs :

- E-faculté admin
- E-campus-admin

Pour l'administration du système fédéré l'administrateur de l'espace numérique s'occupe de la gestion de la BDD fédérée.

IV.3 Diagramme de contexte :

Le diagramme suivant illustre le contexte de notre travail :

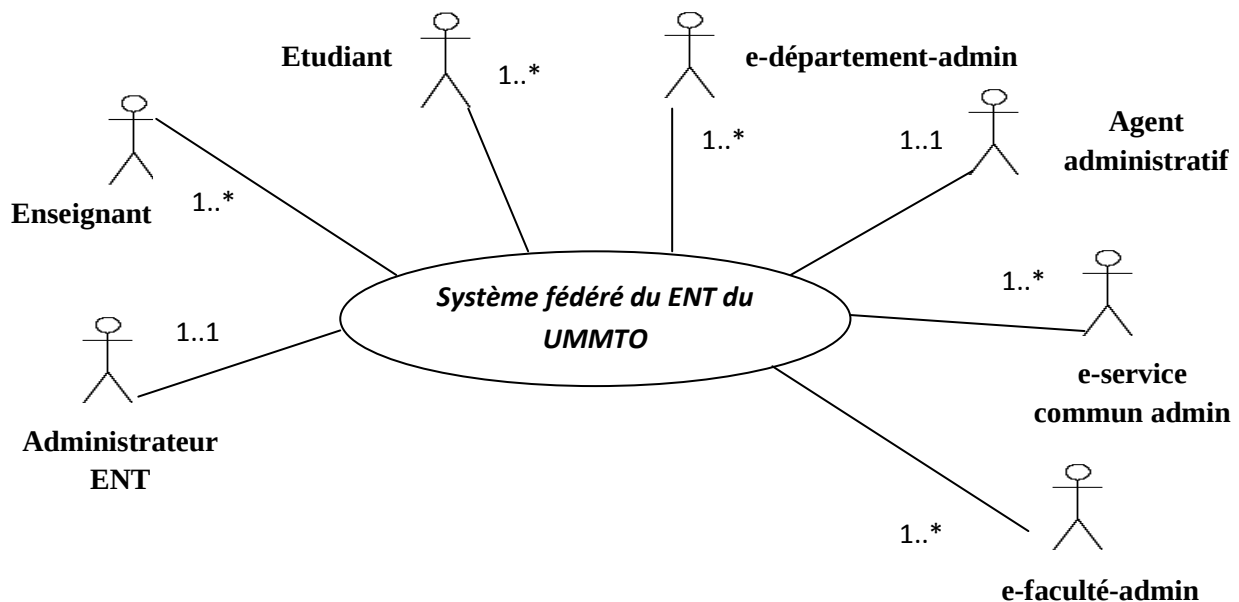


Figure IV.6 : Diagramme de contexte

IV.4 Identification et représentation des cas d'utilisation :

Définition : Un cas d'utilisation désigne une fonctionnalité visible de l'extérieur du système dont on désire décrire le fonctionnement. Un cas d'utilisation doit répondre à un besoin en définissant une partie du comportement attendu du système sans révéler sa structure interne.

Le tableau récapitulatif suivant donne une vue détaillée sur les cas d'utilisation de chaque acteur, classé par site et le type de la base de données, à laquelle peut accéder (base locale ou intégrée), ainsi dans quel espace va naviguer (espace pédagogique ou espace administratif) :

site	Acteur	Cas d'utilisation		Base locale	Base fédéré
		Espace pédagogique	Espace administratif		
	Etudiant	T1 :s'inscrire à l'ENT T2 :s'authentifier T3 : Accéder à son espace T4 : modifier son profil T5 : consulter sa messagerie T6 : Envoyer un message à un enseignant T7 : Envoyer un message à l'administration T8 : consulter son emploi de temps T9: voir les filières et spécialités T10 : voir les modules d'une spécialité T11 : consulter le site faculté T12 : consulter le site département. T13: consulter la bibliothèque centrale T14 : consulter la bibliothèque faculté T15 : consulter la bibliothèque département. T16 : Participer aux forums	T 17:Consultation de ces informations administratif T 18: Consulter le dossier de bourse. T19 : Consulté le module des résidences universitaires.	OUI	
	Enseignant	T1, T2, T3, T4, T5, T7, T8, T10, T11, T12, T13,T14, T15, T16 idem qu'étudiant	T24 : remplir un état de vacation. T25 : Modifier un état de	Oui	Oui

		T20 : Envoyer un message aux étudiants. T21 : Afficher la liste des étudiants. T 22: déposer des cours T 23 : Publier dans un forum.	vacation. T26 : supprimer un état de vacation. T27 : Afficher ces états de vacations. T28 : consulter les avis de virement. T29 : Envoyer un message au service budget. T30 : Afficher les taux de paiement des vacations par grade. T31 : Modifier son numéro de compte. T32 : Modifier son mode de paiement. T33 : mise à jour de ces coordonnées.		
	E-Département admin	T34 : se connecter T35 : Gestion des enseignants (valider inscription, modifier, supprimer). T36 : Gestion des étudiants(valider inscription, modifier, supprimer). T37 : lire et répondre aux messages reçus. T38 : Gestion des formations du département (filières, spécialités, modules) T39 : Mise à jour du site département.	T40 : consulter ces informations administratives personnelles	Oui	
	Chef de section budget		T41 : T21, T22, T26 idem qu'enseignant T42 : Valider les états de vacations. T43 : établir une dépense. T44 : Mandater une dépense (établir un mandat)	Oui	Oui

			T45 : Contacter le comptable. T46 : Consulter les mandats payés (mandats engager par le comptable) T47 : Contacter le CEF (contrôleur financier). T48 : Consulter la liste des mandats régularisés. T49 : afficher la liste de tous les enseignants de la faculté T50 : répondre aux messages reçus. T51 : Envoyer des messages aux enseignants.		
	Agent administratif		T52 : se connecter T53 : gestion des imprimés T54 : répondre aux messages reçus.	Oui	
	Admin e-service commun		T55 : se connecter. T56 : créer un espace pour un chef de section T57 : répondre aux messages reçus. T58 : Mise à jour du site services communs.	Oui	
	e-faculté_admin	T59 : Gestion des départements. T60 : Gestion des services communs. T61 : Gestion des E-département_admin. T62 : Gestion des E-service communs_admin T63 : Mise à jour du site faculté. T64 répondre aux messages reçus.		Oui	Oui
	ENT_admin	T65 : Gestion des E-faculté_admin. T66 : Mise à jour de l'ENT. T67 : Répondre aux messages reçus. T68 : Gestion des facultés.			

Tableau IV.1 : Identification des cas d'utilisations

IV.4.1 Les diagrammes de cas d'utilisation :

Après l'identification des différents acteurs ainsi que les cas d'utilisation qui sont mis en œuvre par ces acteurs, nous terminons la phase d'analyse par l'élaboration des différents diagrammes des cas d'utilisation associés à chaque acteur définie précédemment.

IV.4.1.1 Définition :

Un diagramme de cas d'utilisation permet de représenter graphiquement les cas d'utilisation, le fait qu'un acteur déclenche un cas d'utilisation est représenté par une flèche entre ces deux derniers.

IV.4.1.2 Relations entre les cas d'utilisation :

En plus de la relation de communication, qui consiste au déclenchement d'un cas d'utilisation par un acteur, nous pouvons citer deux types de liens ou relation qui sont les plus utilisés : le lien d'inclusion « include » et le lien d'exclusion « Extend ».

- ***La relation d'inclusion « include »***: Ce lien d'inclusion (ou « include » en anglais) indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination.
- ***La relation d'exclusion « extend »*** : Indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination.

Dans ce qui suit nous allons tracer les différents diagrammes de cas d'utilisation de chaque acteur de notre application :

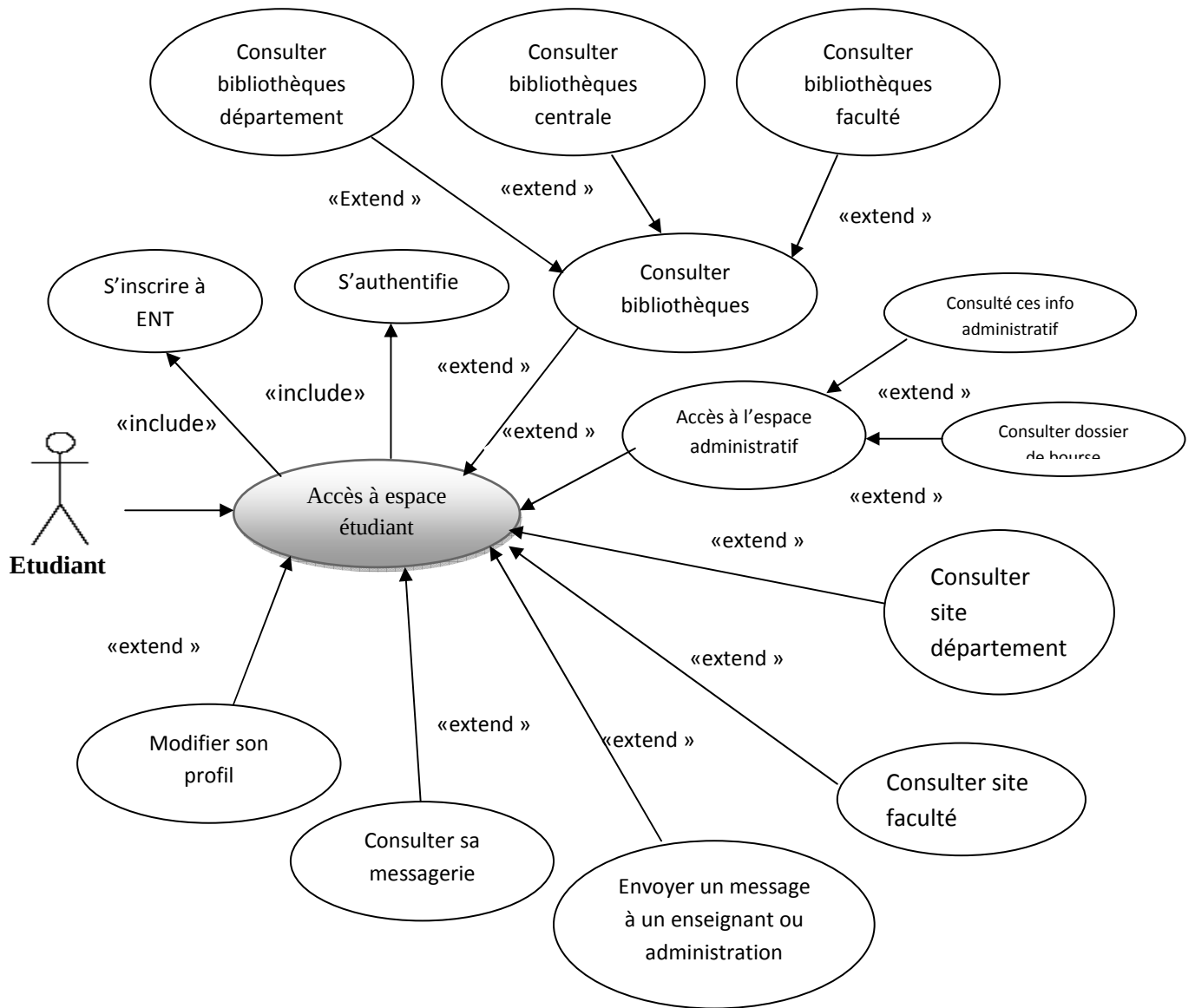


Figure IV.7 : Diagramme de cas d'utilisation Etudiant

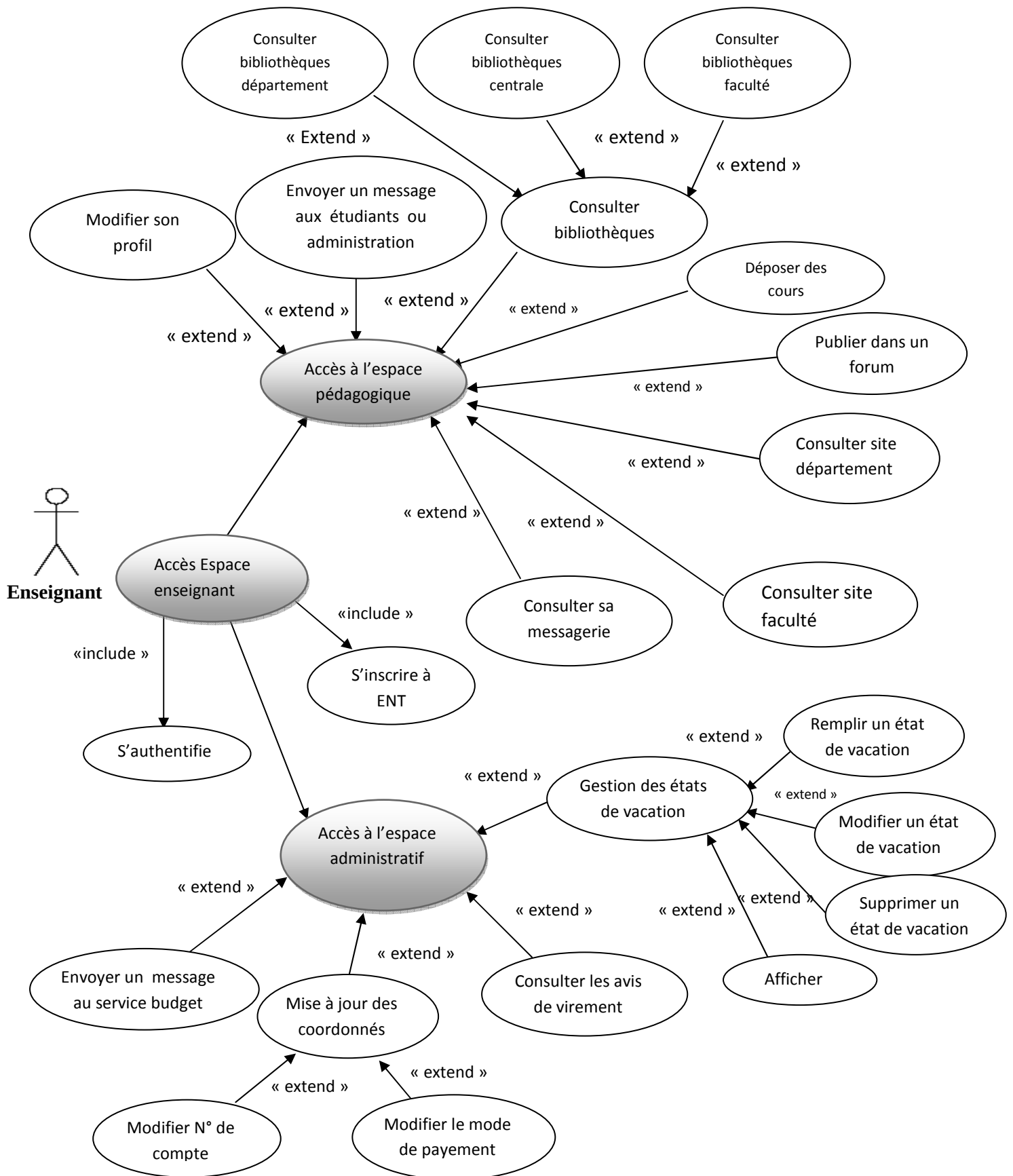


Figure IV.8 : Diagramme de cas d'utilisation « Enseignant »

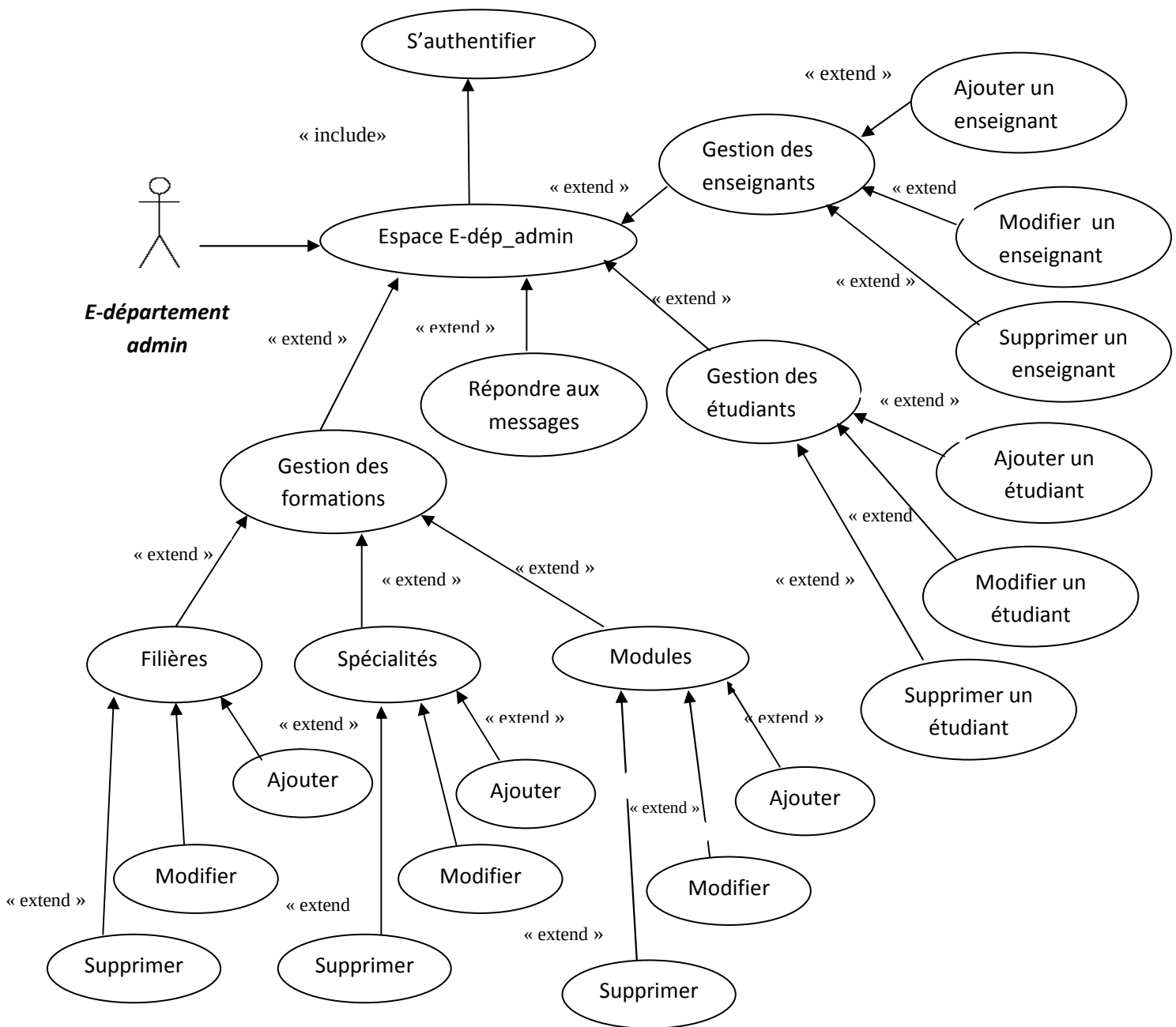


Figure IV.9 : Diagramme de cas d'utilisation « E-département_admin »

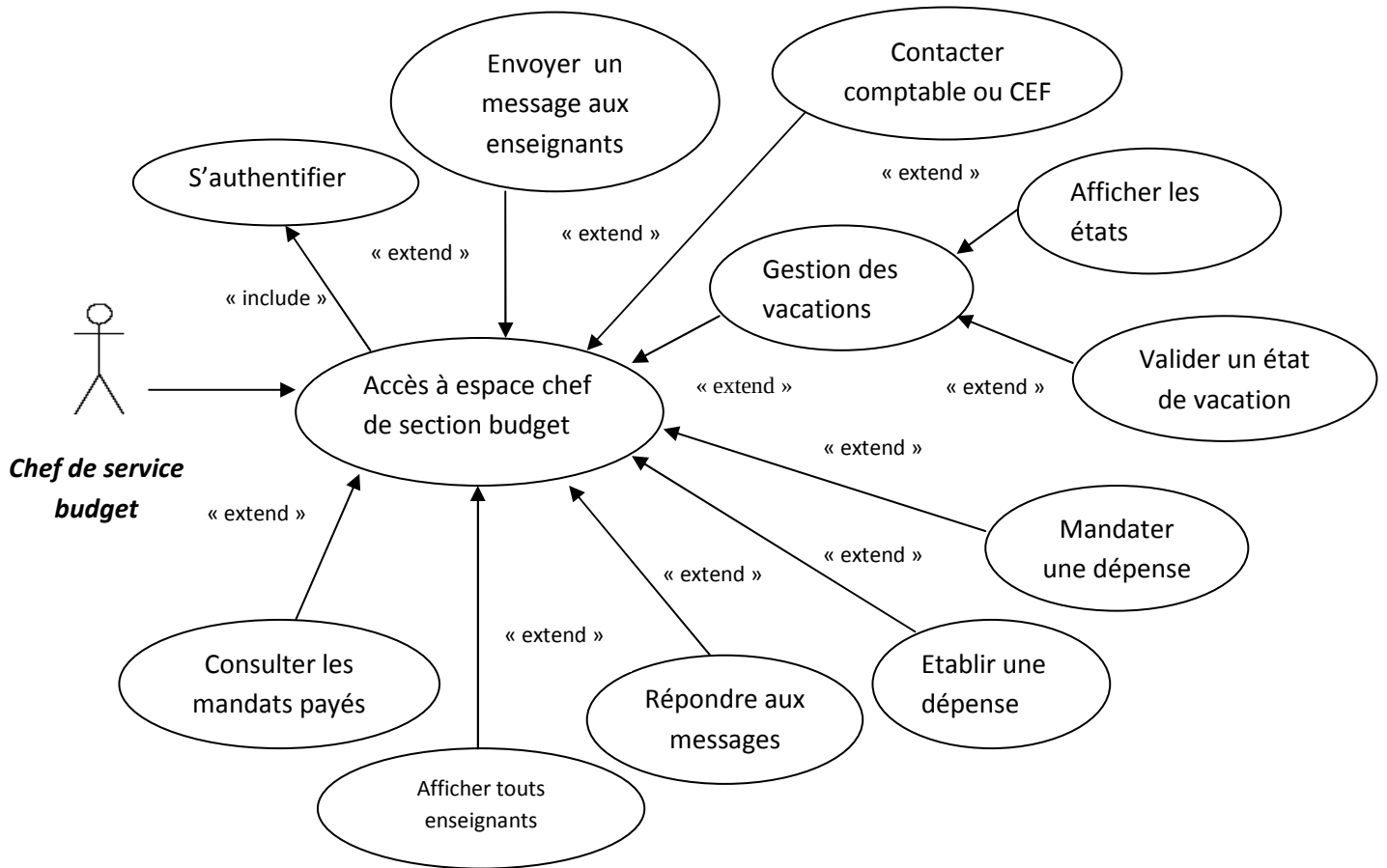


Figure IV.10 : diagramme de cas d'utilisation « Chef de section budget »

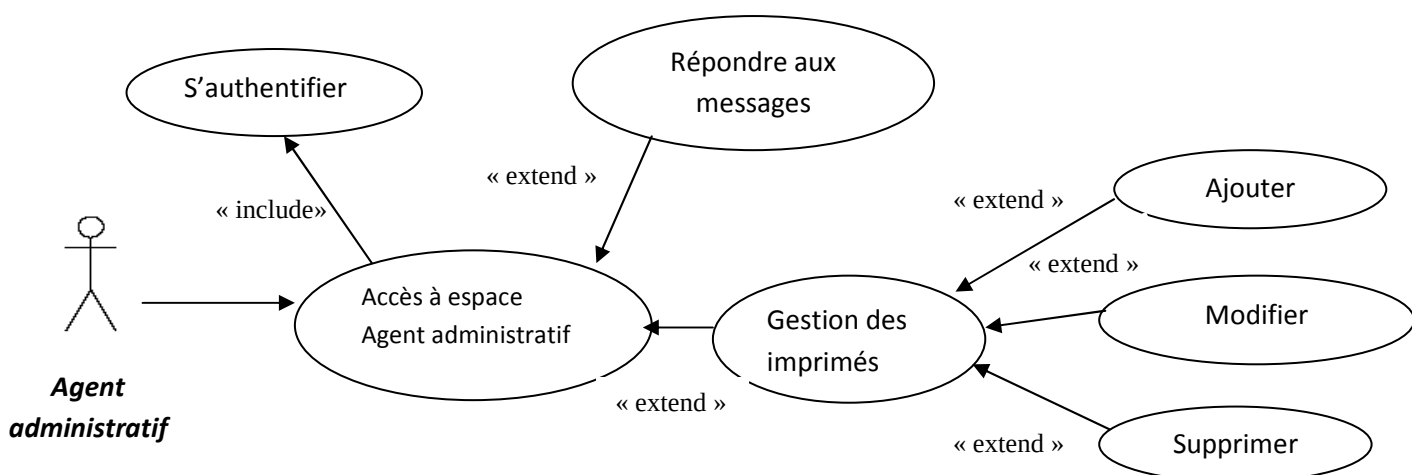


Figure IV.11 : Diagramme de cas d'utilisation « Agent administratif »

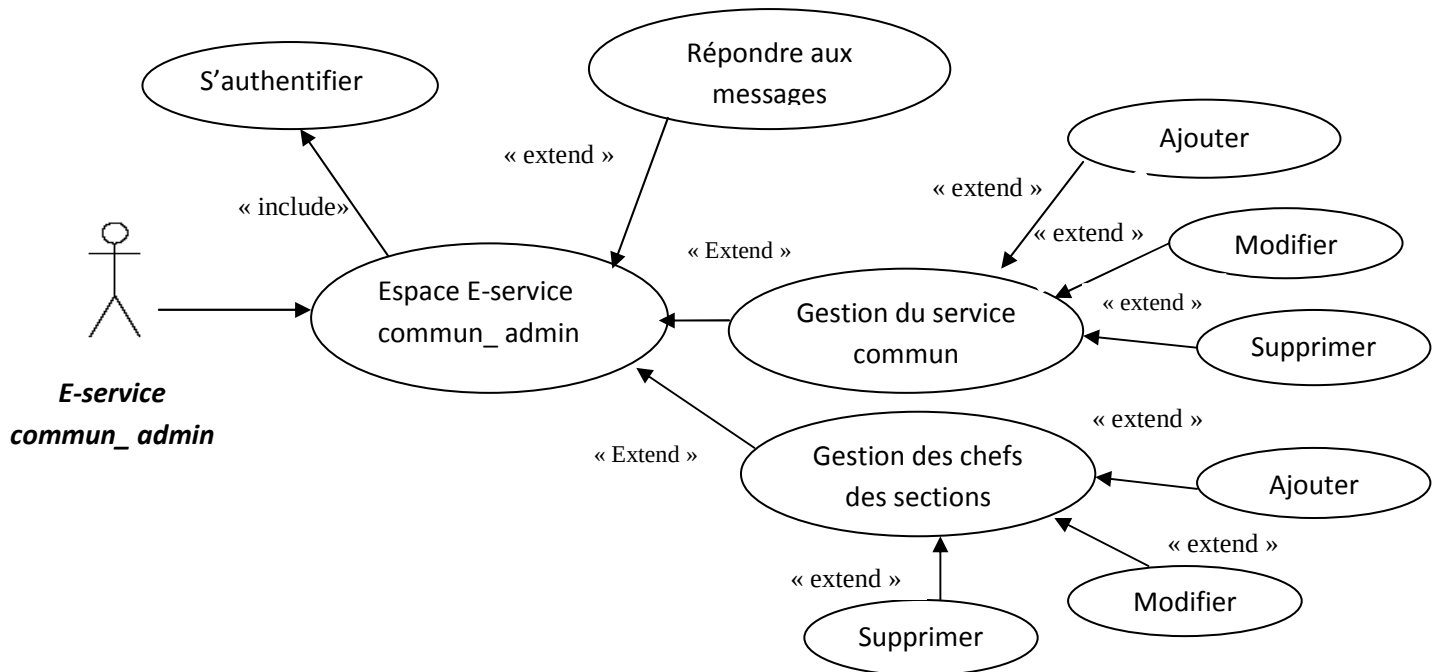


Figure IV.12 : Diagramme de cas d'utilisation «E-service commun_admin»

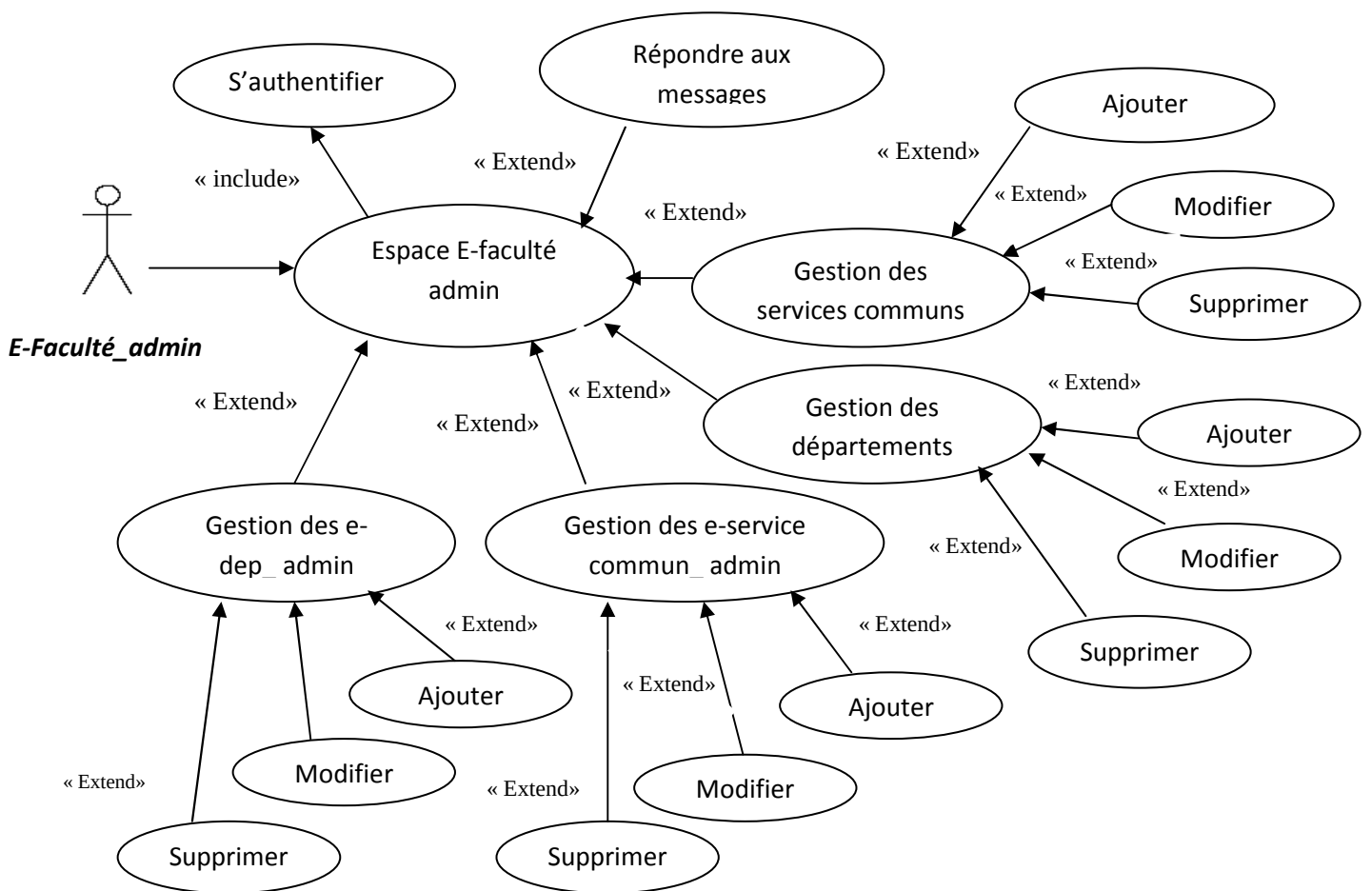


Figure I.V.13 : Diagramme de cas d'utilisation « E-faculté_admin »

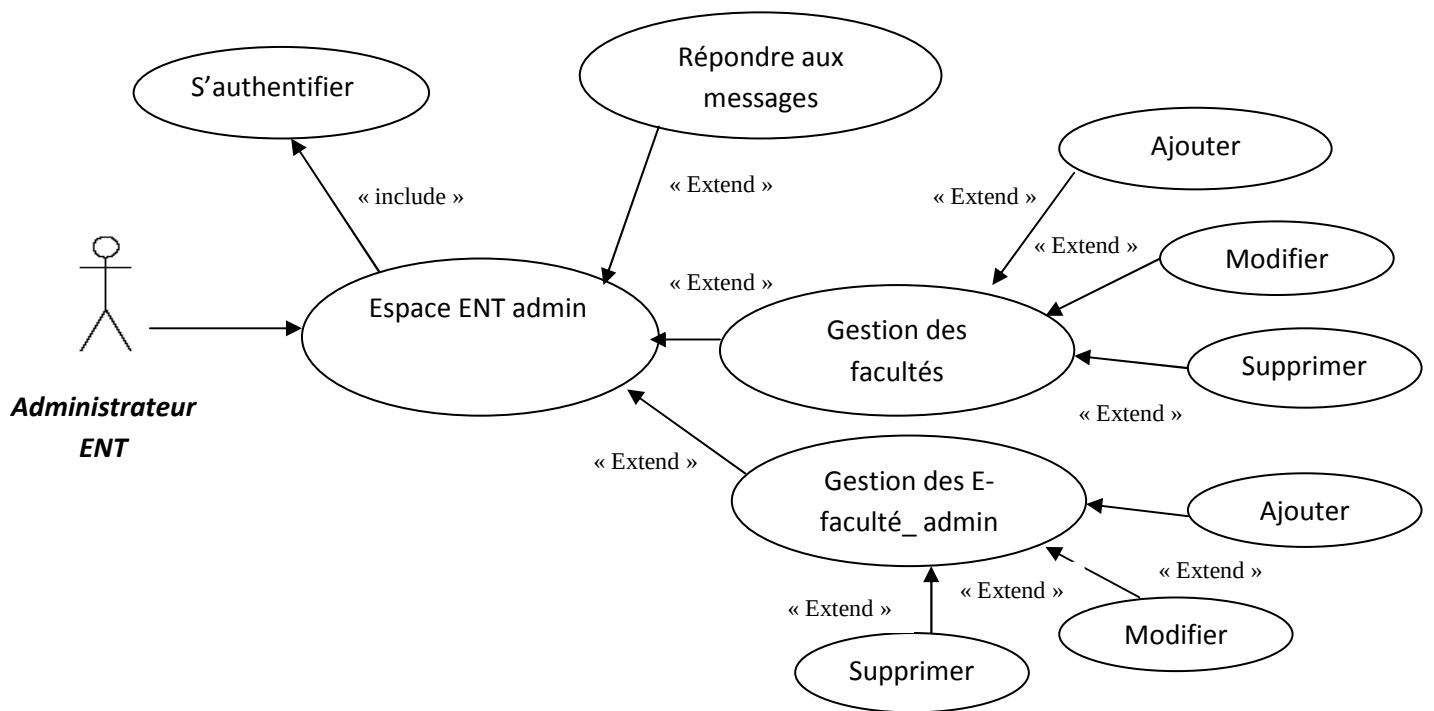


Figure IV.14 : Diagramme de cas d'utilisation « administrateur ENT »

IV.5 Conception :

Après avoir analysés les différents acteurs participant, ainsi leurs cas d'utilisation, nous avons pu terminer cette phase par l'élaboration de leurs différents diagrammes de cas d'utilisation.

Nous allons passer dans ce qui suit à la phase conception qui est une vue qui exprime les modules et exécutables physique sans passer à la partie réalisation proprement dite du système. Cette phase se base essentiellement sur les diagrammes de séquence et diagrammes de classe que nous allons présenter dans ce qui suit :

IV.5.1 Diagrammes de séquence :

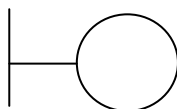
Les diagrammes de séquence décrivant le déroulement de chaque cas d'utilisation, en montrant la façon dont les diverses entités mises en œuvre dans le cas d'utilisation interagissent et collaborent dans le temps (chronologie) afin de réaliser les fonctionnalités attendues.

Typiquement, chaque cas d'utilisation déterminé dans le diagramme des cas d'utilisation fait l'objet d'une étude temporelle des interactions en utilisant un diagramme de séquence par conséquent, on doit avoir autant de diagramme de séquence que de cas d'utilisation, occasionnellement, un cas d'utilisation peut être décrit par plusieurs diagramme de séquence afin de clarifier l'ensemble.

➤ Les objets de type interface :

Ils représentent l'interface entre l'acteur et le système tels des écrans de saisie ou des pages Web complètes.

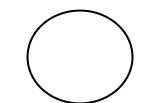
L'icône :



➤ Les objets de type entités :

Ce sont des objets décrits dans le cas d'utilisation mais qui lui survivront. Les opérateurs, les produits, les enregistrements ... sont des objets entité dont les instances peuvent apparaître dans de nombreux cas d'utilisation.

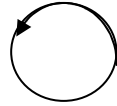
L'icône :



➤ **Les objets de type contrôle :**

Ils représentent des processus, des activités du système tel qu'un calcul ou une recherche; ils dirigent les activités des objets entités et interfaces. Ces types d'objets sont obtenus à partir des verbes des cas d'utilisations.

L'icône :



IV.5.2 Présentation de quelques diagrammes de séquence correspondant aux cas d'utilisation déjà décrits :

Nous avons choisie de présenter quelques exemples des diagrammes de séquence correspondant aux cas d'utilisation présentés précédemment.

Comme nous l'avons cité précédemment notre démarche procède d'une façon ascendante donc nous allons présenter quelques diagrammes de séquence de quelques scénarios déroulent au niveau des sites locaux ainsi dans quels espace vont s'exécutés (Espace pédagogique, espace administratif).

Site locale	Acteurs	Diagrammes de séquence de cas d'utilisation	EP EA
Site département « informatique »	Etudiant et enseignant	- S'inscrire à ENT - S'authentifier - Modifier profil - Envoyer un message	EP
	Enseignant	- Remplir un état de vacation - Modifier le mode de paiement - Modifier le N° de compte	EA
	E-dép_ admin	- Valider inscription (enseignant, étudiant). - Consulter les messages reçus	EA
Site service budget	Chef de section budget	- Valider un état de vacation. - Etablir une dépense. - Mandater une dépense.	EA

Tableau IV .2 : Liste des diagrammes de séquence choisis

IV.5.2.1 Cas d'utilisation « S'inscrire à l'ENT » :

Il contient les objets suivants :

➤ **Objet interface :**

- Page principale
- Formulaire d'inscription
- Page de confirmation

➤ **Objet contrôle :**

- Enregistrer dans la base de données

➤ **Objet entité :**

- Inscription

Remarque : Dans ce diagramme de séquence, l'utilisateur c'est soit étudiant soit enseignant.

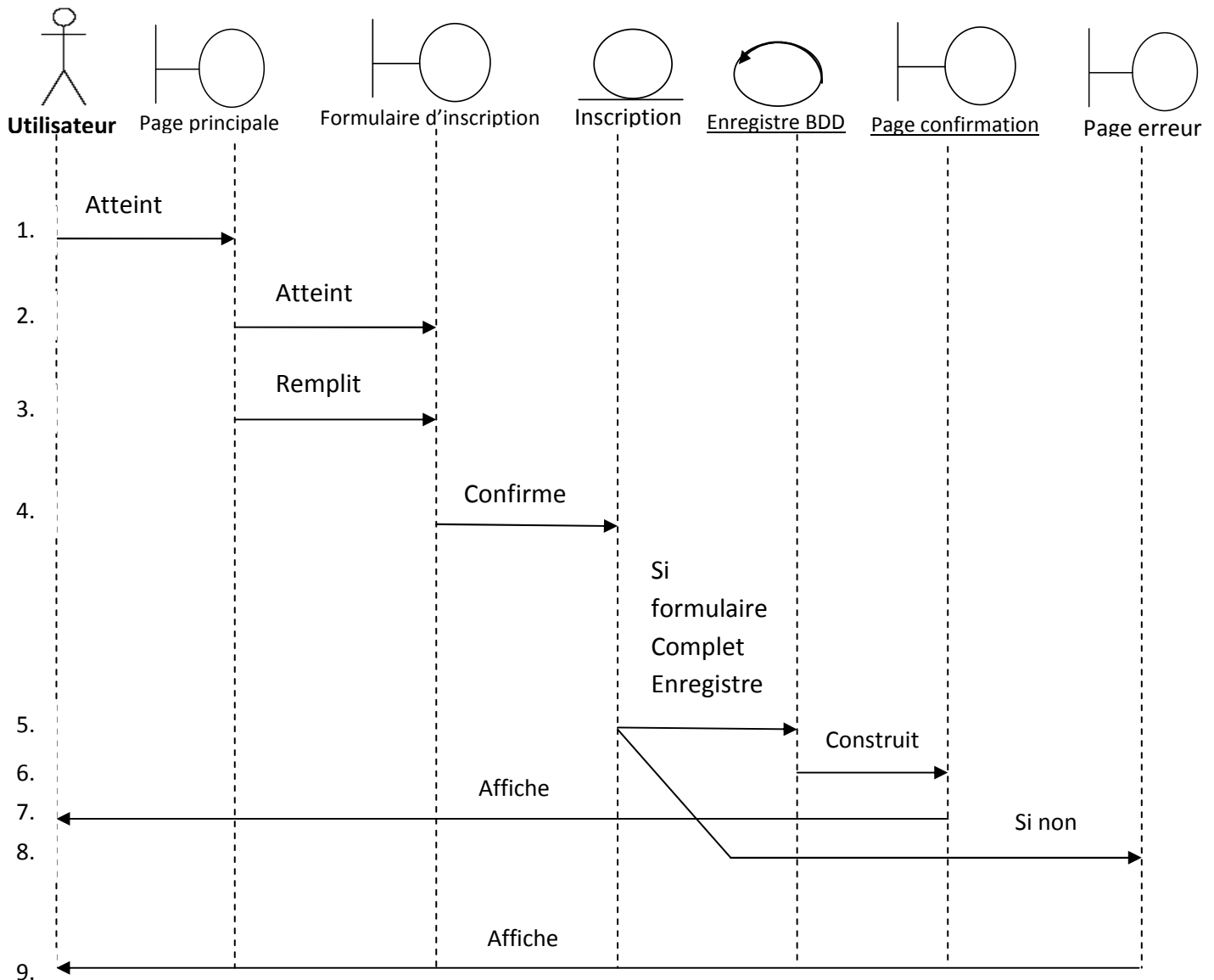


Figure IV.15: Diagramme de séquence de cas d'utilisation « S'inscrire à ENT »

1. L'utilisateur atteint la page principale (page d'accueil) de l'ENT en saisit l'URL.
2. L'utilisateur clique sur le lien inscription et le système lui affiche le formulaire d'inscription.
3. L'utilisateur remplit le formulaire d'inscription.
4. L'utilisateur confirme son inscription avec le bouton valider.
5. Le système vérifie si le formulaire est complet, il enregistre l'inscription dans la base de données.
6. Le système construit la page de confirmation.
7. Le système affiche la page de confirmation
8. Si non il redirige vers la page d'erreur.
9. Le système affiche un message d'erreur (page d'erreur)

IV.5.2.2Cas d'utilisation « s'authentifier » :

Il contient les objets suivants :

➤ ***Objet interface :***

- Page principale
- Page d'authentification
- Espace personnel

➤ ***Objet contrôle :***

- La recherche dans la base de données

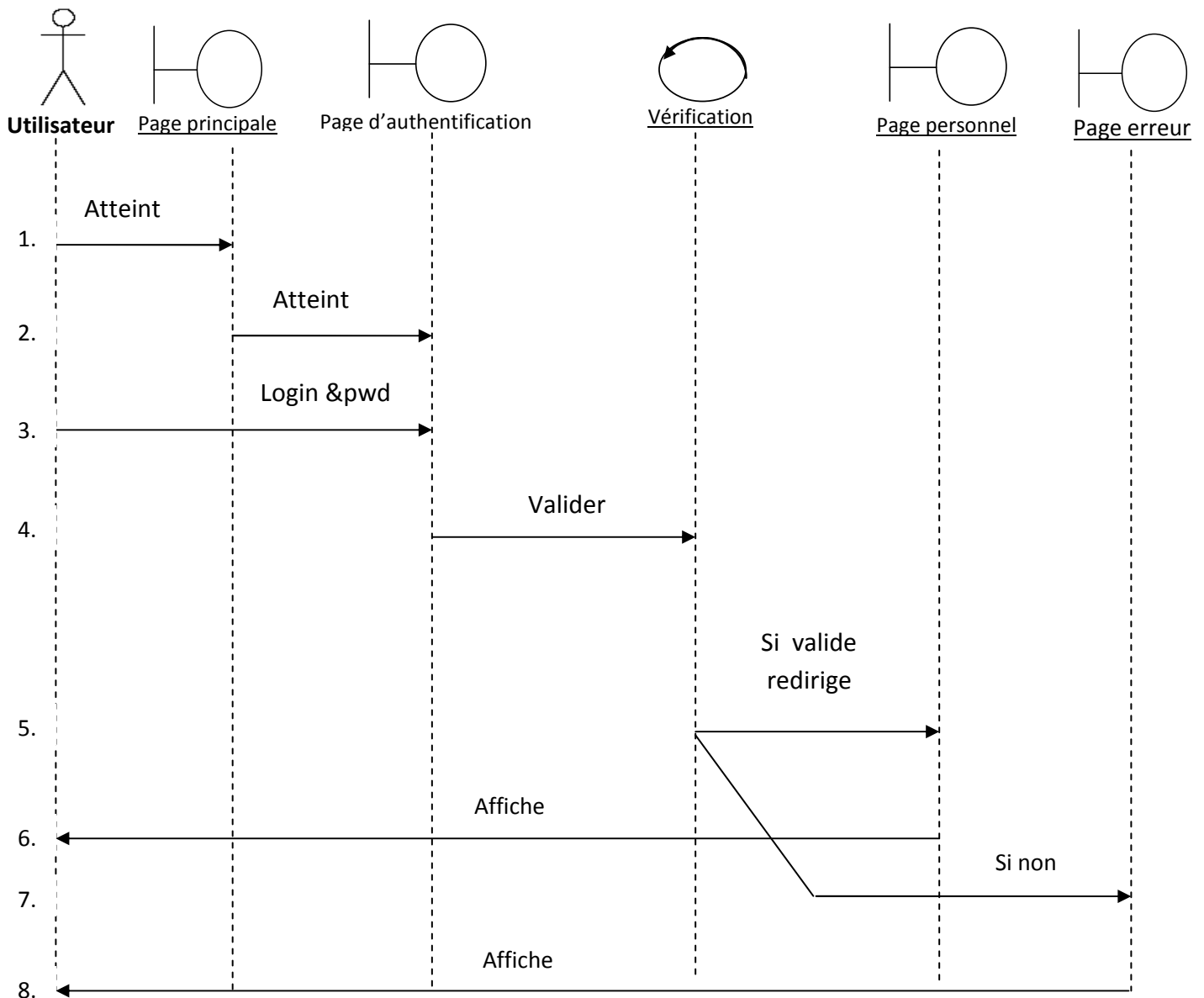


Figure IV.16: Diagramme de séquence de cas d'utilisation « S'authentifier »

1. L'utilisateur atteint la page principale (page d'accueil) de l'ENT en saisit l'URL.
2. L'utilisateur clique sur le lien « authentifier » et le système le redirige vers la page d'authentification.
3. L'utilisateur saisit le login et le mot de passe.
4. L'utilisateur valide avec le bouton valider.
5. Le système vérifie si l'utilisateur existe dans la base de données, s'il existe il le redirige vers son espace personnel
6. Le système affiche la page de l'espace personnel pour l'utilisateur.
7. S'il n'existe pas le système le redirige vers la page d'erreur
8. Le système affiche un message d'erreur (page d'erreur)

IV.5.2.3 Cas d'utilisation « Modifier profil » :

Il contient les objets suivants :

➤ **Objet interface :**

- Page personnel
- Page profil
- Page de confirmation

➤ **Objet contrôle :**

- Enregistrer dans la base de données

Remarque : dans ce diagramme de séquence « Modifier profil » l'utilisateur c'est soit étudiant soit enseignant

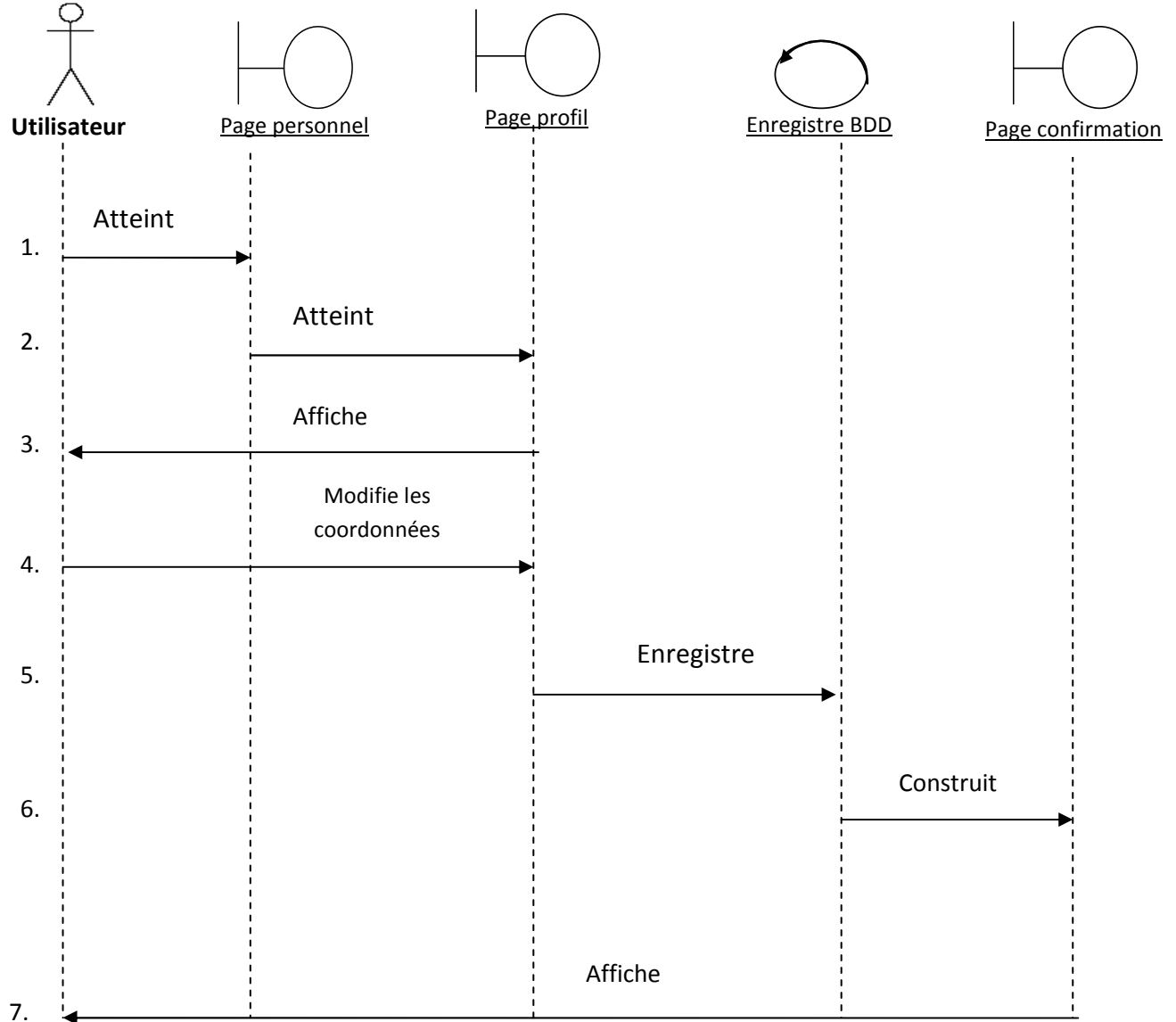


Figure IV.17: Diagramme de séquence pour cas d'utilisation « Modifier profil »

1. L'utilisateur atteint sa page personnel après son authentification.
2. L'utilisateur clique sur le lien mon profil.
3. Le système lui affiche la page de son profil.
4. L'utilisateur modifie ces coordonnées dans le formulaire de profil et valide avec le bouton valider.
5. Le système enregistre les modifications et met à jour le profil dans la base de données.
6. Le système construit la page de confirmation.
7. Le système affiche la page de confirmation.

IV.5.2.4 Cas d'utilisation « Envoyer un message à un étudiant ou enseignant ou administration » :

Il contient les objets suivants :

➤ **Objet interface :**

- Page messagerie
- Page nouveau message
- Page de confirmation

➤ **Objet contrôle :**

- Enregistrer dans la base de données

➤ **Objet entité :**

- Message

Remarque : dans ce diagramme l'utilisateur c'est étudiant et enseignant.

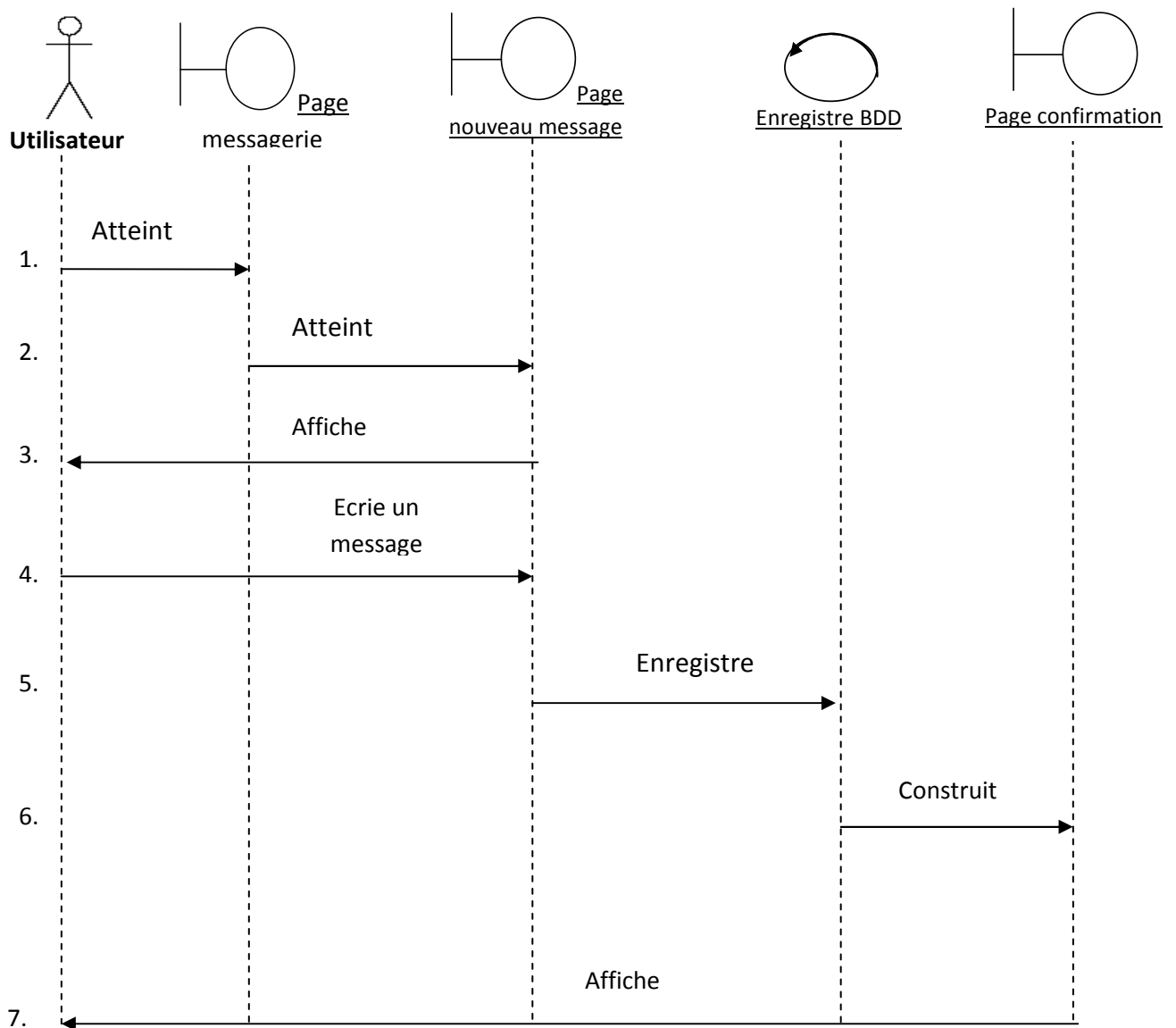


Figure IV.18 : Diagramme de séquence de cas d'utilisation « envoyer un message à un étudiant ou enseignant ou administration »

1. L'utilisateur atteint la page de messagerie après l'étape d'authentification
2. L'utilisateur clique sur le lien nouveau message et le système lui affiche la page de nouveau message
3. L'utilisateur saisit le message et l'adresse de destinataire et clique sur le bouton envoyer
4. Le système enregistre le message dans la base de données
5. Le système construit la page de confirmation
6. Le système affiche la page de confirmation de l'envoi de message.

IV.5.2.5 Cas d'utilisation « Remplir un état de vacation » :

Il contient les objets suivants :

- ***Objet interface :***
 - Page personnel
 - Page formulaire d'état de vacation
 - Page de confirmation
- ***Objet contrôle :***
 - Enregistrer dans la base de données
- ***Objet entité :***
 - Etat de vacation

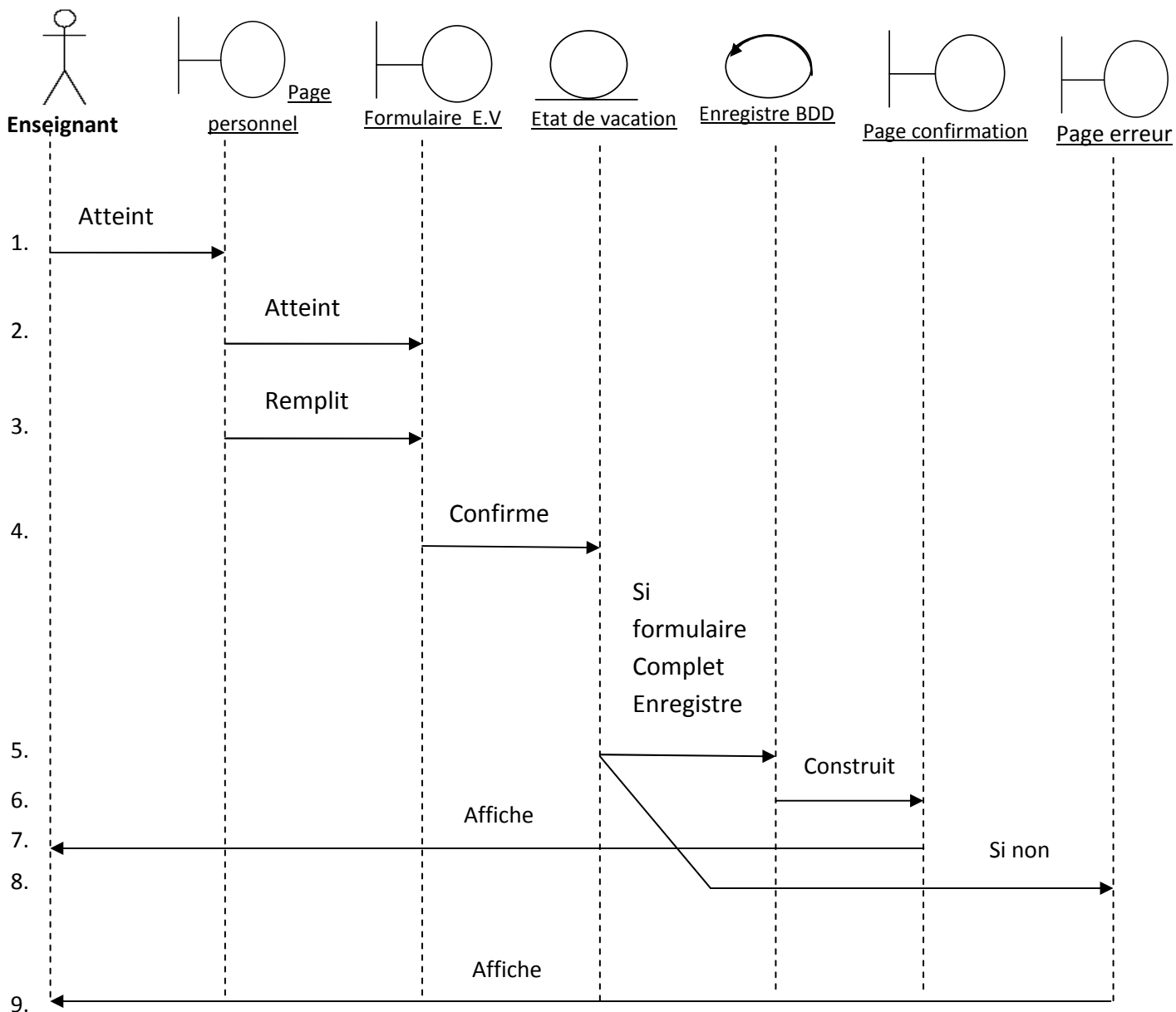


Figure IV.19: Diagramme de séquence de cas d'utilisation « Remplir un état de vacation »

1. L'enseignant atteint la page personnel après l'authentification.
2. L'enseignant clique sur le lien remplir un état de vacation et le système lui affiche le formulaire à remplir.
3. L'enseignant remplit les informations du formulaire.
4. L'enseignant confirme les informations avec le bouton valider.
5. Le système vérifie si le formulaire est complet, il enregistre l'état de vacation dans la base de données.
6. Le système construit la page de confirmation.
7. Le système affiche la page de confirmation
8. Si non il le redirige vers la page d'erreur.
9. Le système affiche un message d'erreur (page d'erreur)

IV.5.2.6 Cas d'utilisation « Modifier le mode de paiement (par banque ou CCP) » :
 Il contient les objets suivants :

➤ **Objet interface :**

- Page personnel
- Page formulaire de mode de paiement
- Page de confirmation

➤ **Objet contrôle :**

- Enregistrer dans la base de données

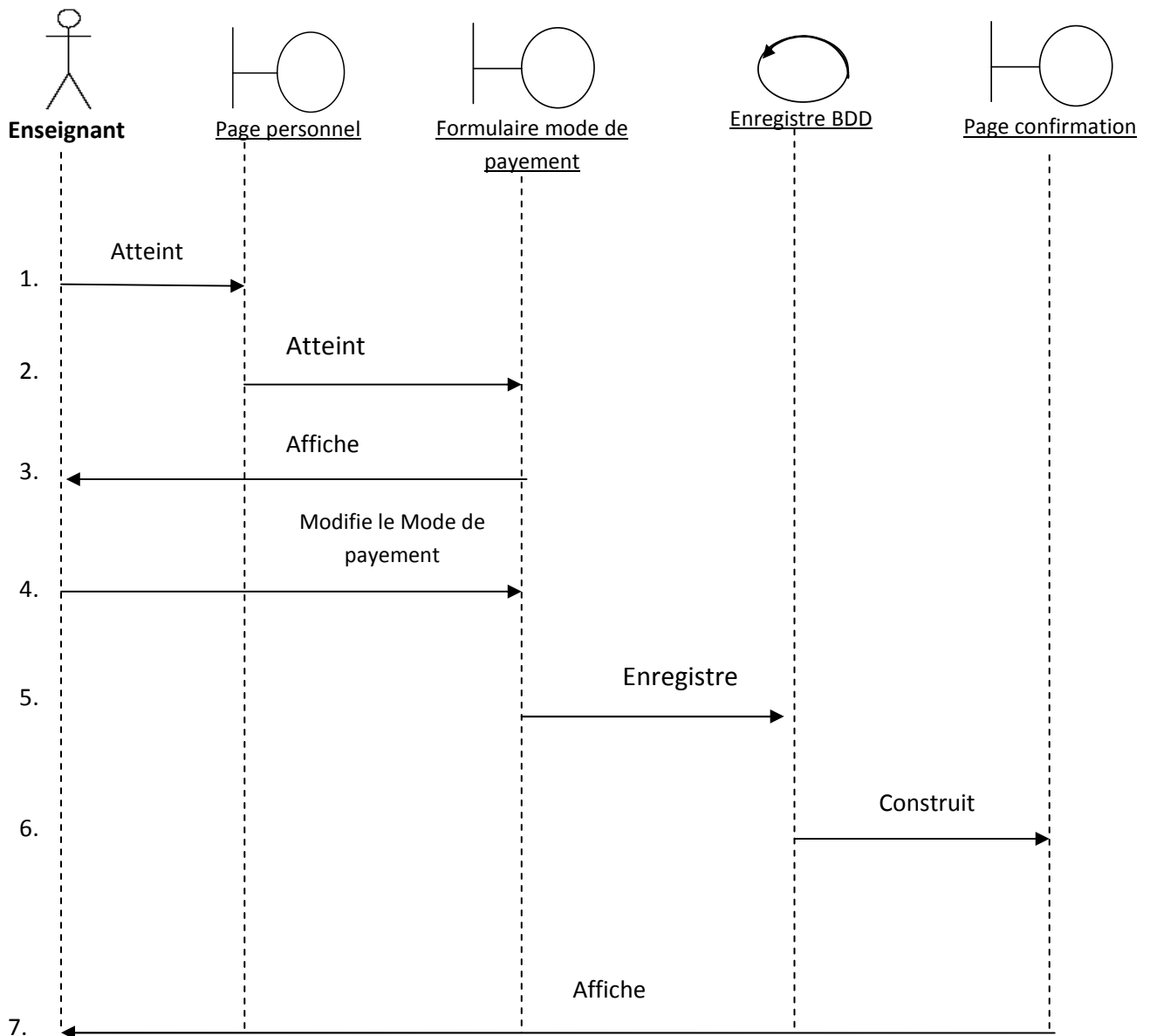


Figure IV.20 : Diagramme de séquence de cas d'utilisation « Modifier le mode de paiement »

1. L'enseignant atteint sa page personnel après son authentification.
2. L'enseignant clique sur le lien « modifier Mode de paiement ».
3. Le système lui affiche la page de Modification.
4. L'enseignant modifie son mode de paiement dans le formulaire et valide avec le bouton valider.
5. Le système enregistre les modifications et met à jour la base de données.
6. Le système construit la page de confirmation.
7. Le système affiche la page de confirmation.

IV.5.2.7 Cas d'utilisation « Modifier le numéro de compte » :

Il contient les objets suivants :

- ***Objet interface :***
 - Page personnel
 - Page Modifier numéro de compte
 - Page de confirmation
- ***Objet contrôle :***
 - Enregistrer dans la base de données
- ***Objet entité :***
 - Numéro de compte.

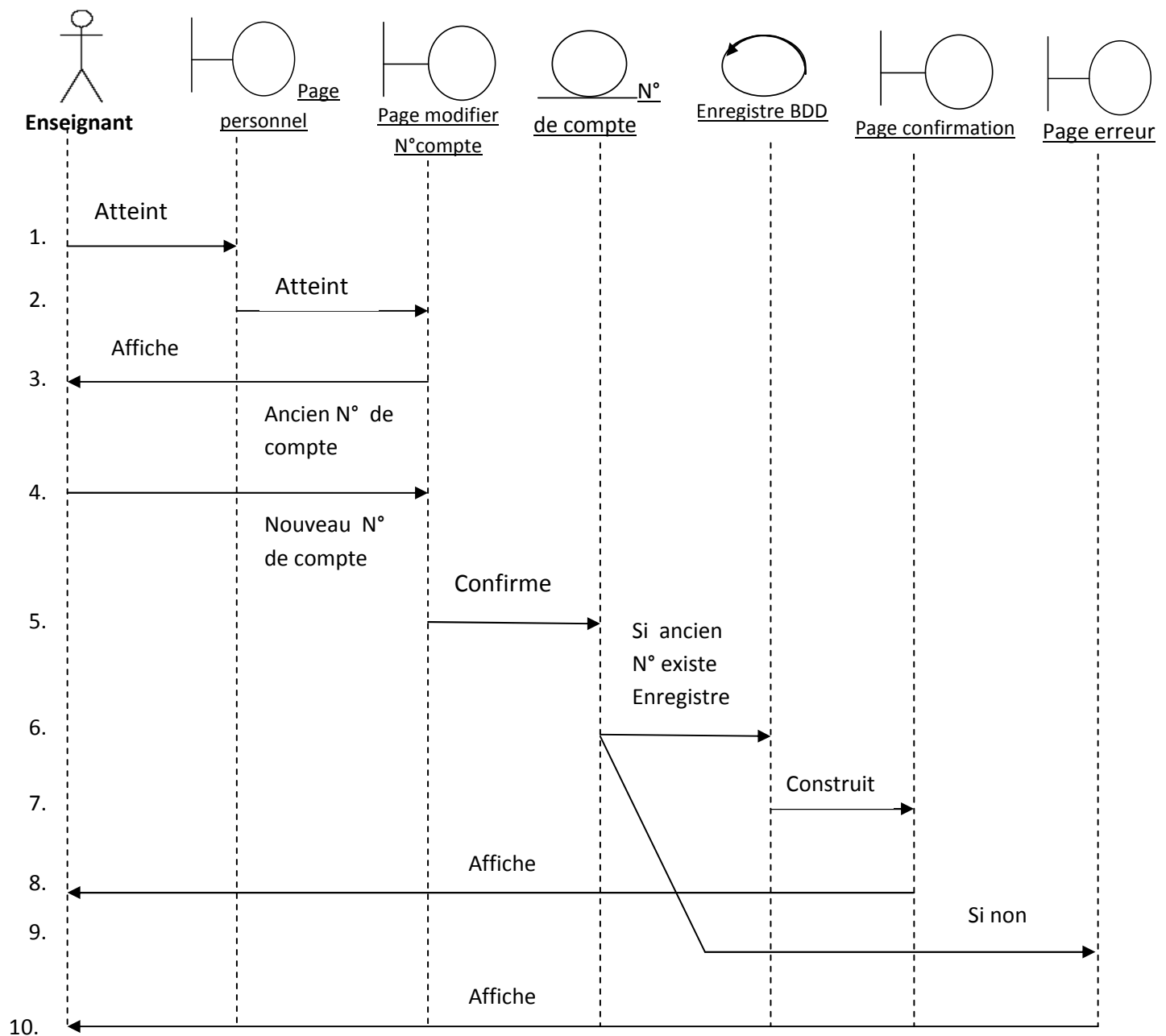


Figure IV.21: Diagramme de séquence de cas d'utilisation «Modifier le numéro de compte »

1. L'enseignant atteint sa page personnel après son authentification.
2. L'enseignant clique sur le lien « modifier mon numéro de compte ».
3. Le système lui affiche la page de Modification du numéro de compte.
4. L'enseignant saisit son ancien numéro de compte et le nouveau numéro
5. L'enseignant valide les informations saisi avec le bouton valider.

6. Si l'ancien numéro de compte saisi correspond bien à celui qui existe déjà dans la base de données le système enregistre les modifications et met à jour la base de données.
7. Le système construit la page de confirmation.
8. Le système affiche la page de confirmation.
9. Si non « s'il y a un conflit ou erreur de saisi » le système le redirige vers la page d'erreur.
10. Le système lui affiche le message d'erreur « la page d'erreur ».

IV.5.2.8 Cas d'utilisation « Valider inscription enseignant » :

Il contient les objets suivants :

- ***Objet interface :***
 - Page personnel
 - Page inscription des enseignants
 - Page de confirmation
- ***Objet contrôle :***
 - Enregistrer dans la base de données
- ***Objet entité :***
 - Inscription

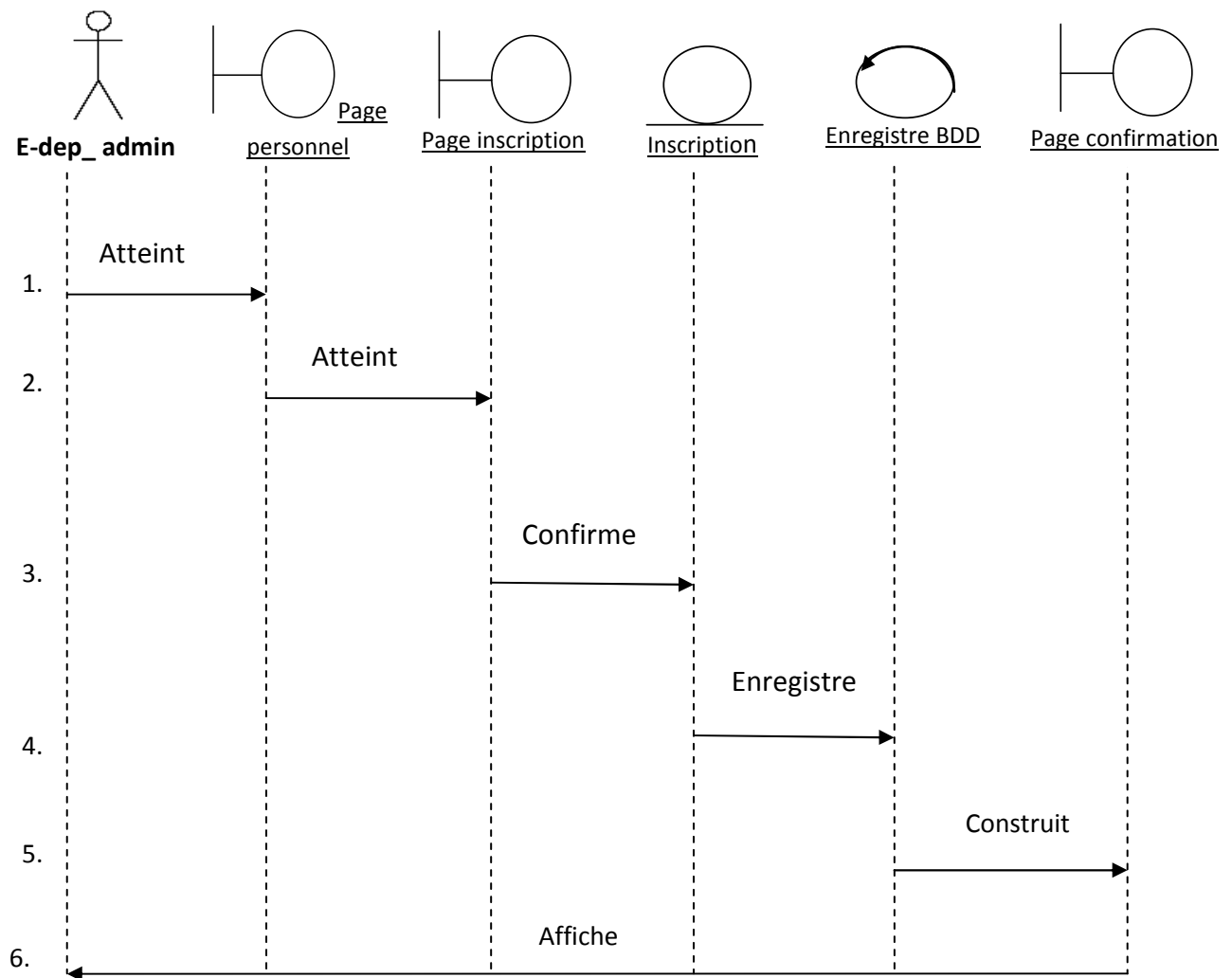


Figure IV.22: Diagramme de séquence de cas d'utilisation « Valider inscription de l'enseignant »

1. l'administrateur de département atteint son espace personnel en s'authentifiant à l'e-département.
2. Il clique sur le lien inscription enseignant qui sera affiché par le système.
3. Il clique sur l'inscription d'enseignant et clique sur le lien valider inscription.
4. Le système enregistre l'inscription de l'enseignant dans la base de données.
5. Le système construit la page de confirmation qui confirme l'enregistrement d'inscription
6. Le système lui affiche la page de confirmation.

IV.5.2.9 Cas d'utilisation « Consulter messagerie » :

Il contient les objets suivants :

➤ **Objet interface :**

- Page personnel
- Page messagerie
- Page des messages reçus

➤ **Objet contrôle :**

- Chercher dans la base de données

➤ **Objet entité :**

- Message

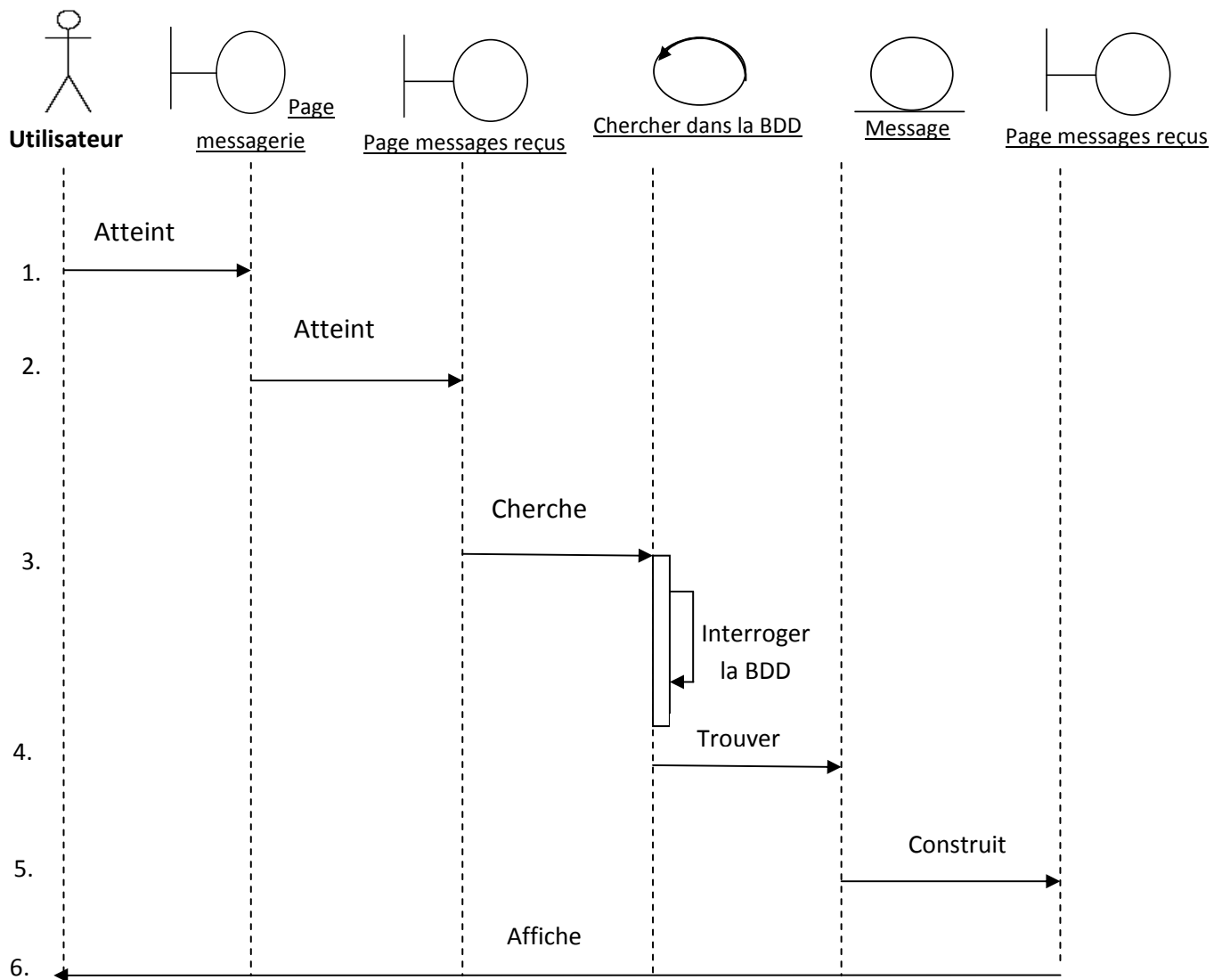


Figure IV.23: Diagramme de séquence de cas d'utilisation « Consulter les messages reçus »

Remarque :

Dans le diagramme de séquence « consulter les messages reçus » l'utilisateur représente un enseignant, étudiant et administrateur (soit un administrateur d'un département ou administrateur de faculté ou d'ENT).

1. L'utilisateur atteint la page de messagerie après son authentification.
2. L'utilisateur clique sur le lien mes messages reçus.
3. Le système cherche tous les messages qui existent dans la base de données
4. Le système obtient les messages reçus après sa recherche.
5. Le système construit la page de messages reçus.
6. Le système affiche la page de messages reçus.

IV.5.2.10 Cas d'utilisation « Valider un état de vacation » :

Il contient les objets suivants :

- **Objet interface :**
 - Page personnel
 - Page des états de vacation.
 - Page de confirmation.
- **Objet contrôle :**
 - Enregistrer dans la base de données
- **Objet entité :**
 - Etat de vacation.

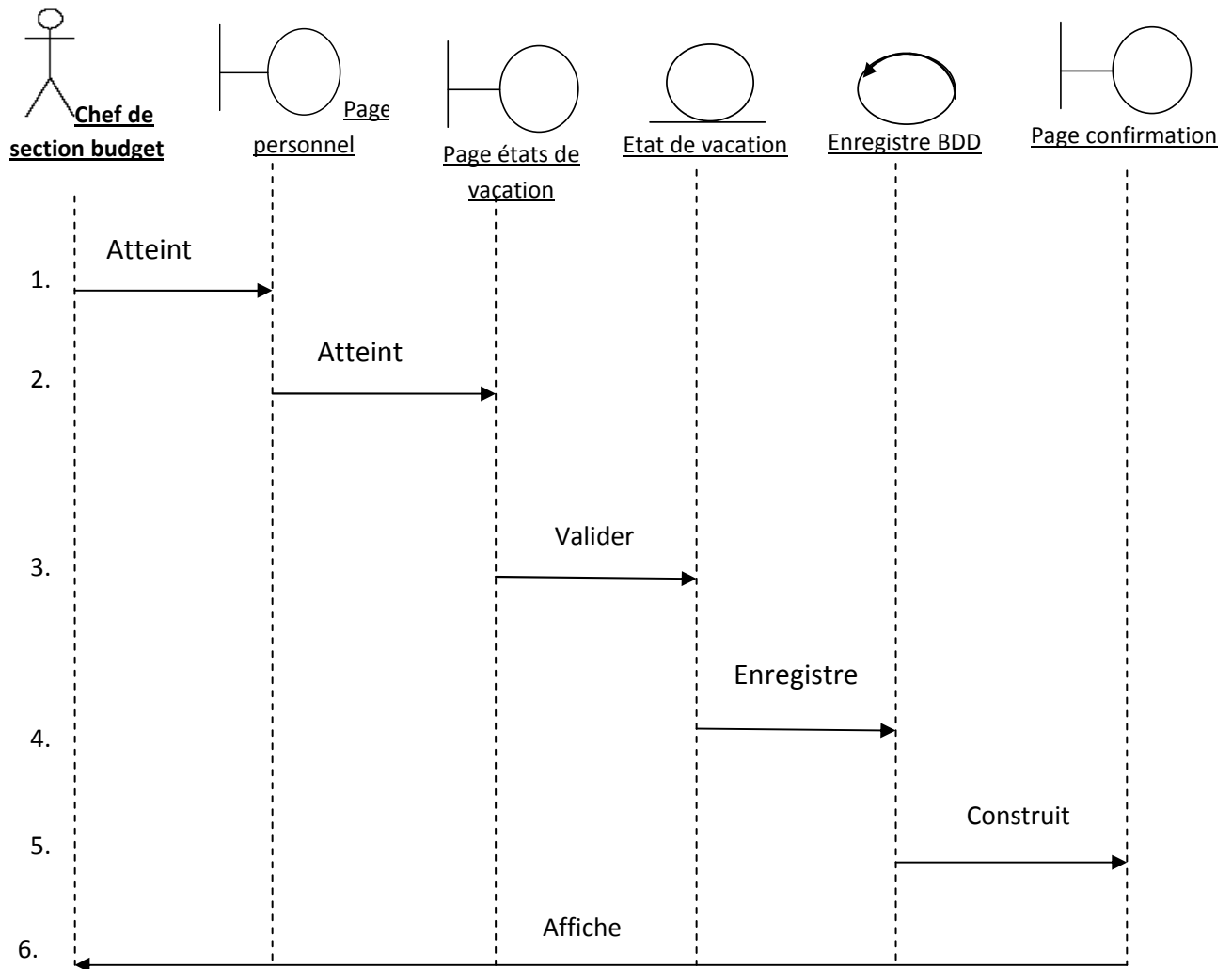


Figure IV.24: Diagramme de séquence de cas d'utilisation « valider un état de vacation »

1. Le chef de section budget atteint son espace personnel après son authentification.
2. Il clique sur le lien des états de vacation des enseignants qui sera affiché par le système
3. Il clique sur état de vacation d'enseignant et clique sur le lien valider Etat de vacation.
4. Le système enregistre la validation de l'état de vacation de l'enseignant dans la base de données.
5. Le système construit la page de confirmation.
6. Le système lui affiche la page de confirmation.

IV.5.2.11 Cas d'utilisation «Etablir une dépense» :

Il contient les objets suivants :

➤ **Objet interface :**

- Page personnel
- Page des dépenses.
- Page de confirmation.

➤ **Objet contrôle :**

- Enregistrer dans la base de données
- Calcul de dépense.

➤ **Objet entité :**

- Dépense.

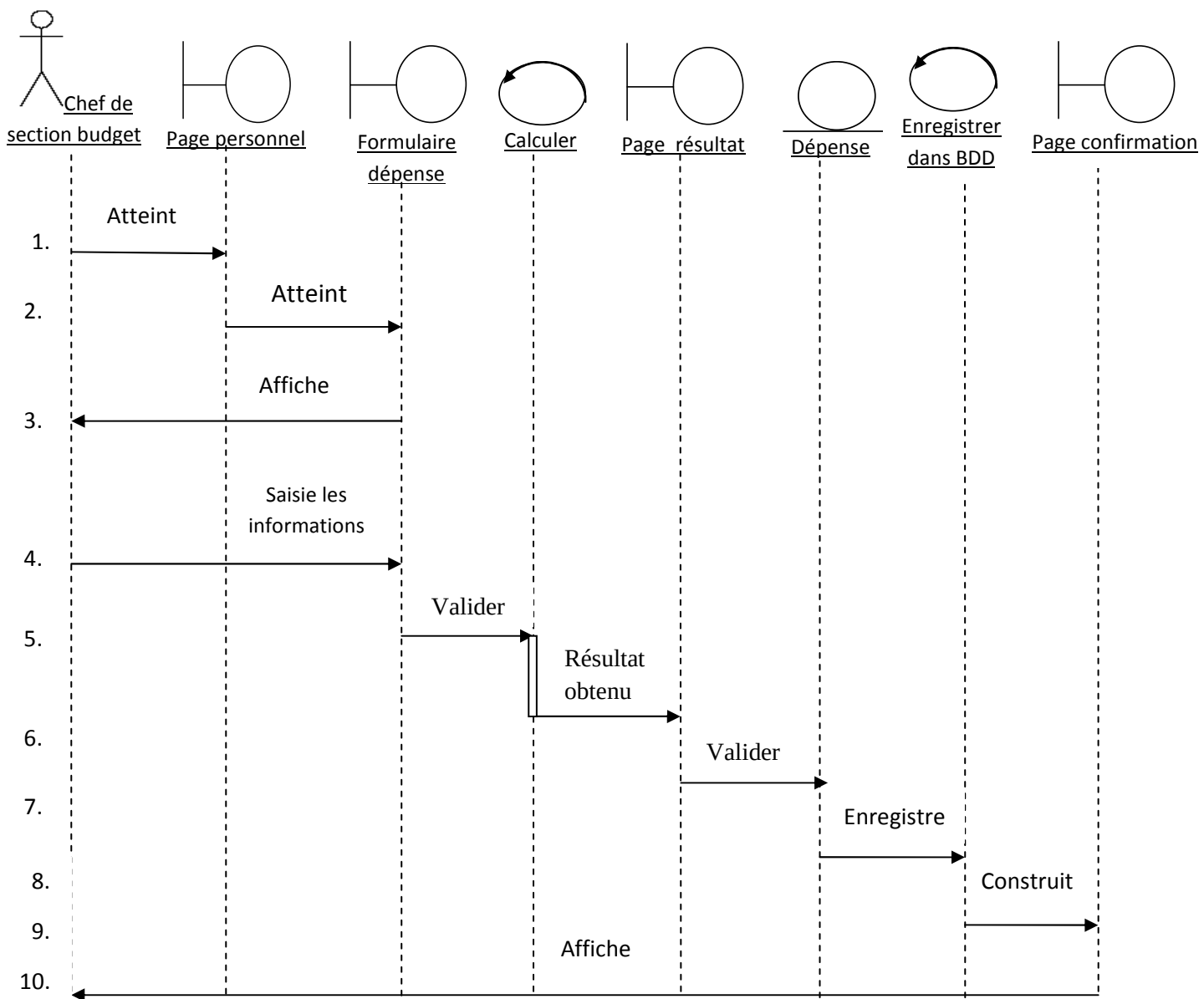


Figure IV.25: Diagramme de séquence de cas d'utilisation « Etablir une dépense »

1. Le chef de section budget atteint sa page personnelle après son authentification.
2. Le chef de section clique sur le lien « Etablir une dépense ».
3. Le système lui affiche le formulaire des dépenses.
4. Le chef de section budget saisit les informations de la dépense (N°, date, les nets à payer ...)
5. Le chef de section valide les informations saisies avec le bouton valider.
6. Le système calcul le montant et il le redirige vers une page résultat.
7. Le chef de section valide les résultats obtenus.
8. le système enregistre la dépense et met à jour la base de données.
9. Le système construit la page de confirmation.
10. Le système affiche la page de confirmation.

IV.5.2.12 Cas d'utilisation «Mandater une dépense» :

Il contient les objets suivants :

- **Objet interface :**
 - Page personnel
 - Page des mandats.
 - Page de confirmation.
- **Objet contrôle :**
 - Enregistrer dans la base de données
- **Objet entité :**
 - Dépense.

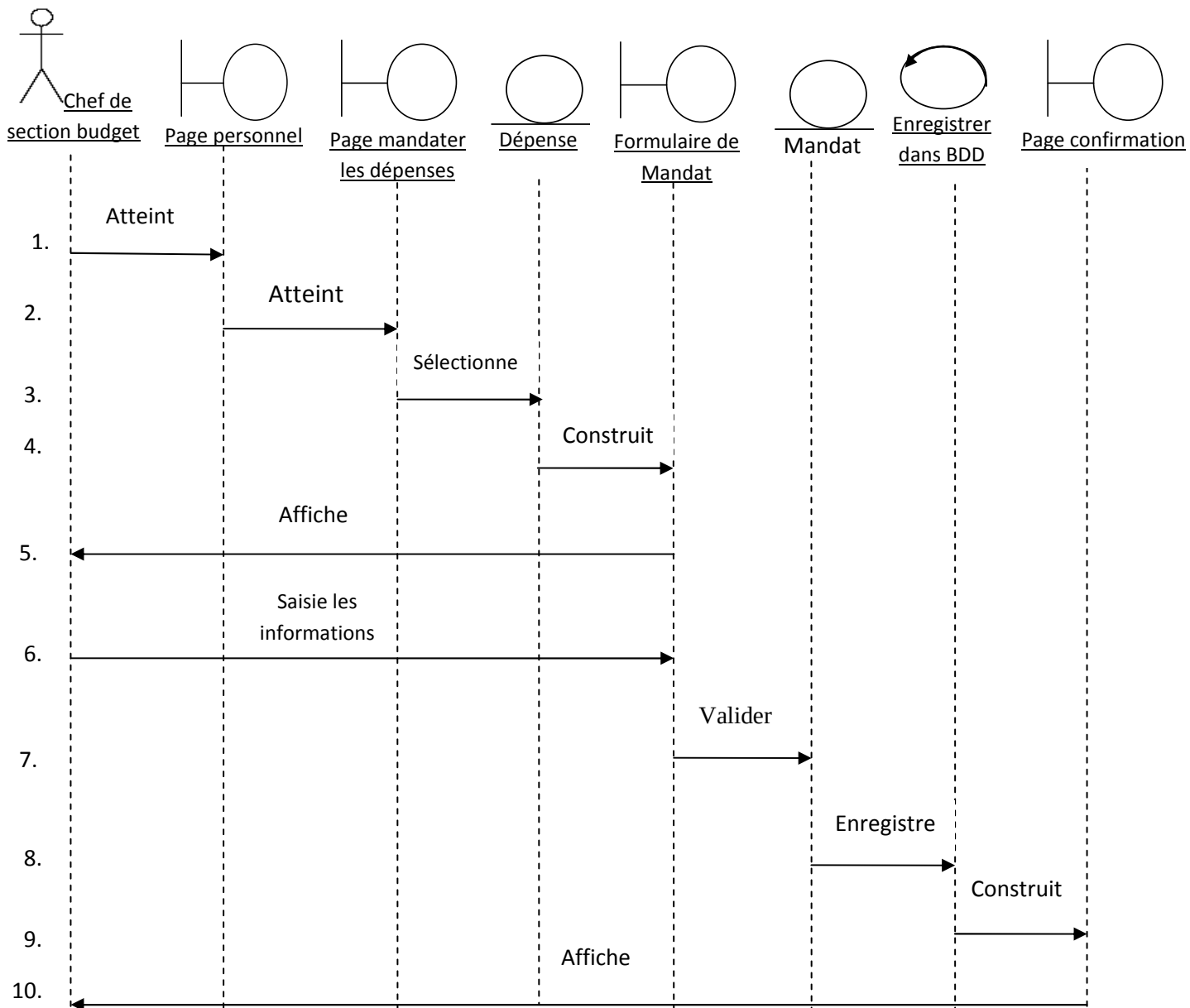


Figure IV.26 : Diagramme de séquence de cas d'utilisation « Mandater une dépense »

1. Le chef de section budget atteint sa page personnel après son authentification.
2. Le chef de section clique sur le lien « Mandater une dépense ».
3. Le système lui affiche la page des dépenses.
4. Le chef de section budget sélectionne la dépense qui sera mandaté.
5. Le système construit la page du formulaire des Mandats.
6. Le chef de section budget saisit les informations du mandat.
7. Le chef de section valide les informations saisies avec le bouton valider.

8. le système enregistre la dépense et met à jour la base de données.
9. Le système construit la page de confirmation.
10. Le système affiche la page de confirmation.

IV.6 Diagramme de classe :

Définition : un diagramme de classe est la vue statique des objets qui donnent naissance à la base de données, leur intérêt majeur est de modéliser les entités d'un système. Autrement dit, ils expriment les relations existantes entre les pages client et serveur.

Nous allons présenter les schémas conceptuels locaux des bases locales, par la suite on présente les vues créées pour compléter le schéma conceptuel global de la base de données fédérée comme le montre la figure suivante :

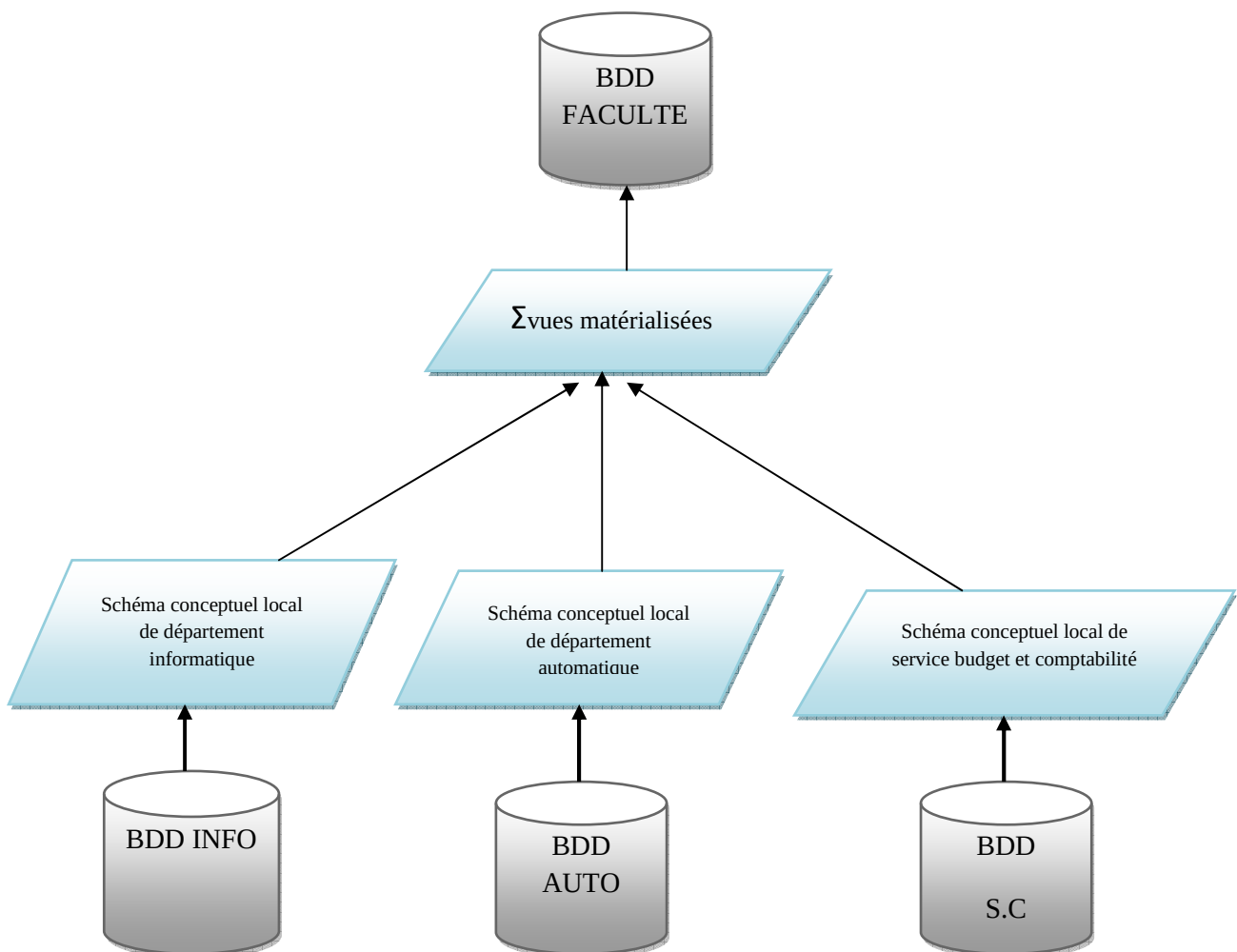
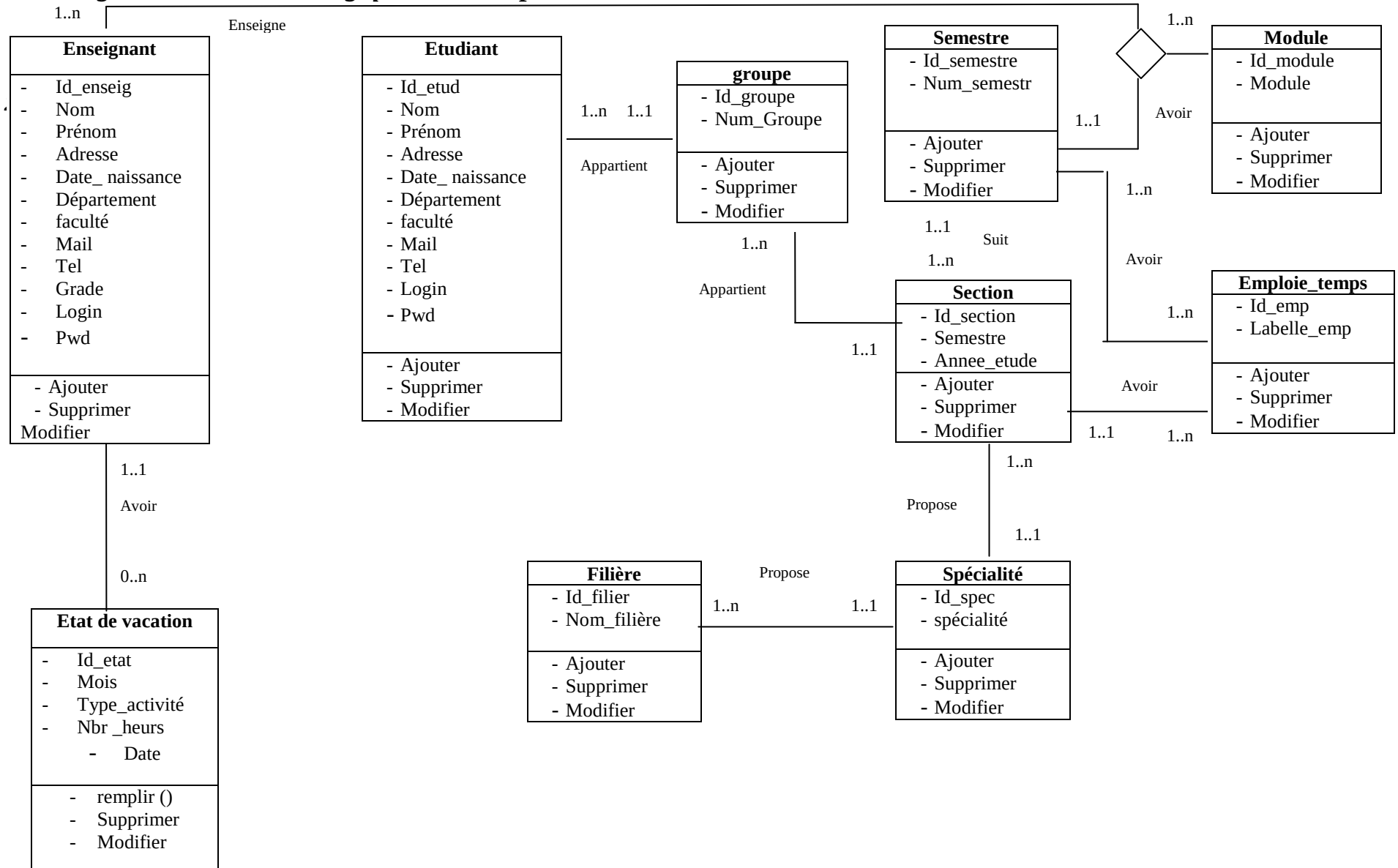
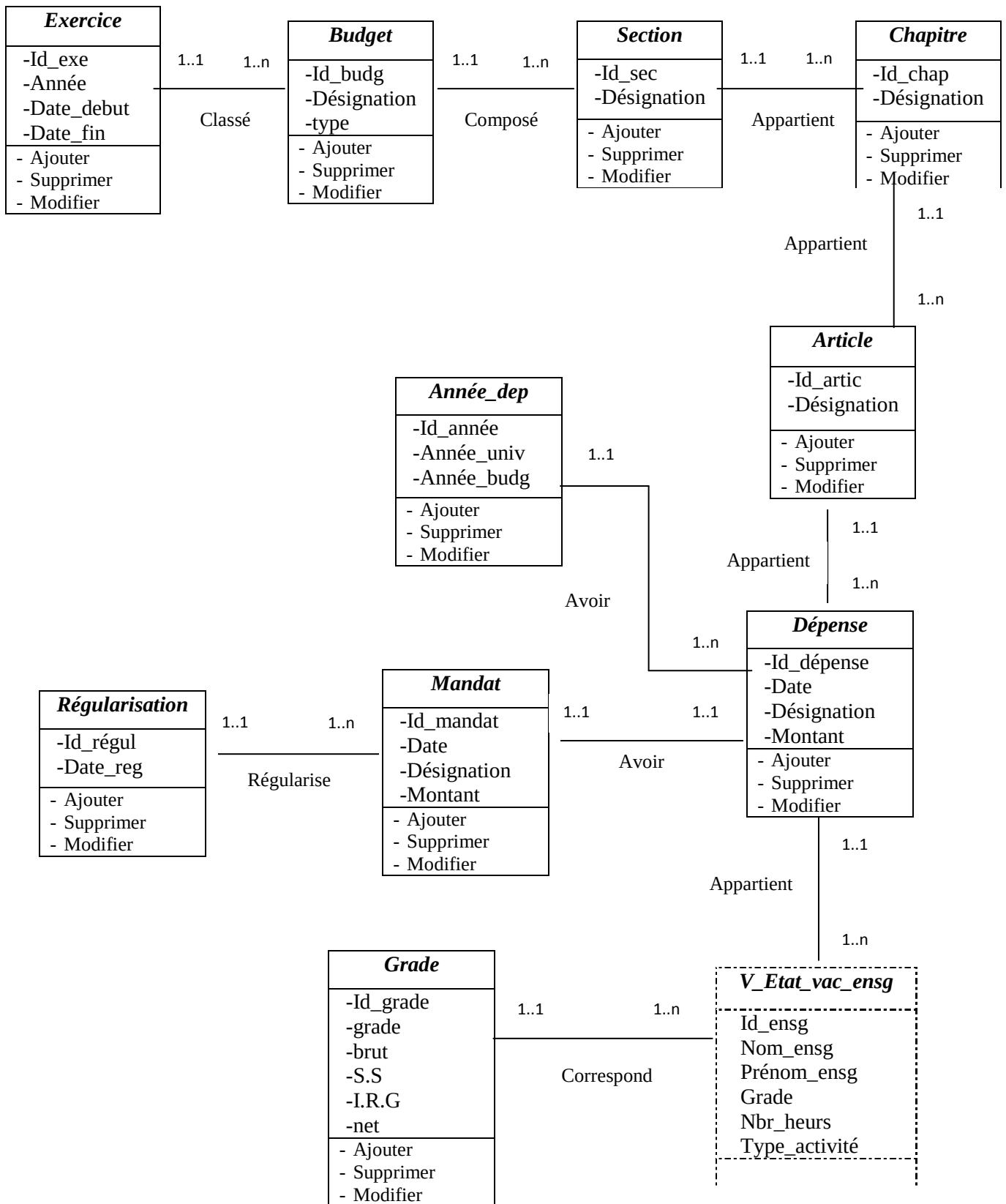


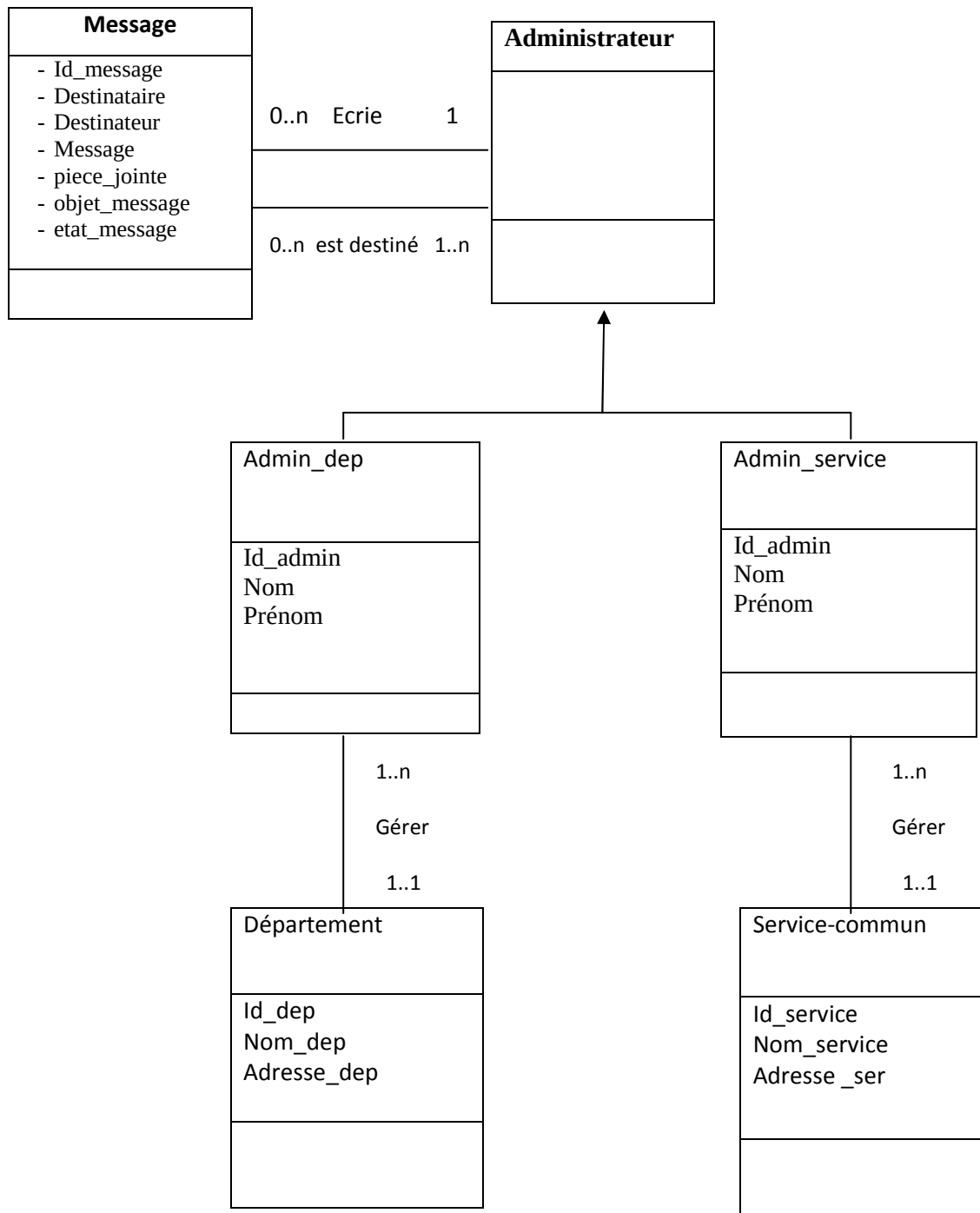
Figure IV.27: Plan de représentation des schémas conceptuels

IV.6.1 Diagramme du schéma logique du site département :



IV.6.2 Diagramme du schéma logique du site service budget :



IV.6.3 Diagramme du schéma logique de la base faculté :

IV.6.4 Les vues matérialisées créer:

La fédération des données se fait en utilisant des vues matérialisées, c'est-à-dire que les données spécifiques à chaque site restent au niveau de la BDD de ce site, par contre les données communes entre les sites sont représentées par des vues matérialisées.

Dans notre cas les données communes entre le département et le service budget est l'état de vacation de l'enseignant qui est stocké au niveau de la base de données département « informatique ou automatique », mais qui va être utilisée par le service budget pour réaliser son processus, d'où le chef de section budget a besoin de voir les enseignants ainsi leurs états de vacation, donc on doit récupérer ces données de la base département sous forme de vue, qui sera manipulée par le service budget dans son processus métier.

Les tables suivantes présentent des exemples de vues matérialisées créés :

<i>Etudiants_fac</i>
-Id_etud -Nom_etud -Prénom_etud -Département -email -tel
- Supprimer - Modifier

↑
Vue sur tous les
étudiants de la
faculté GEI

<i>Enseignants_fac</i>
-Id_etud -Nom_eng -Prénom_eng -Département -Grade_eng -email -tel
- Supprimer - Modifier

↑
Vue sur tous les
étudiants de la
faculté GEI

Vue sur tous les états de
vacations de tous les
enseignants de la faculté

<i>Etat_vac_eng_fac</i>
-Id_etat -Nom_eng -Prénom_eng -Département -Grade_eng -Type_activite -email -tel -Nbrcour -nbrTD -nbrTP -mois -brut -ss -irg -net -valider
- Supprimer - Modifier

IV.7 Le modèle logique :

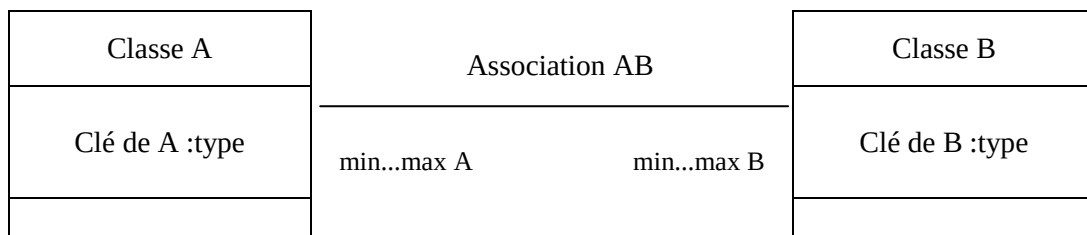
Le schéma relationnel (ou model logique) peut être obtenu à partir d'un schéma conceptuel tel qu'un diagramme de classe, pour ce faire un certain nombre de règles du passage d'UML doivent être respectées. Ces règles sont les suivantes

➤ **Les classes :**

La règle qui concerne les classes est la suivante : chaque classe devient une relation, chaque attribut de la classe devient un attribut de la relation. Concernant la clé, soit il existe une clé pour la classe qui deviendra la clé de la relation, ou bien, on ajoute une clé artificielle à la relation.

➤ **Traduction des associations :**

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



Le tableau suivant donne les cas possibles de traduction, concernant le maximum des cardinalités de A et de B :

Max A \ Max B	1	>1
1	Ajouter la clé de B dans la relation de A comme attribut ou ajouter la clé de A dans la relation de B	Ajouter la clé de B dans la relation de A comme attribut
>1	Ajouter la clé de A dans la relation de B comme attribut	Créer une relation AB ayant comme attribut la clé de A et la clé de B

Tableau IV.3: Les règles de passage du modèle conceptuel au modèle relationnel

Après la définition des règles du passage UML du conceptuel vers le modèle logique, nous obtenons les relations suivantes :

IV.7.1 Modèle logique du site département :

Filière (id_filiere, désignation)

Spécialité (id_spec, désignation, \$id_filiere)

Section (id_section, section, annee_cour,\$id_spec)

Semestre (id_semestre, semestre, \$id_section)

Groupe (id_group, groupe, \$id_section)

Module (id_module, nom_module, coefficient, \$id_semestre,\$id_enseignant)

Enseignant (id_engs, nom, prénom, date_naiss, adresse, département, faculté, mail, Tel, grade, login, pwd)

Etat_vacation (id_etat, mois, type_activité, nbr_heurs,\$id_engs)

Etudiant (id_etud, nom, prénom, date_naiss, adresse, mail, Tel, login, pwd, \$id_group)

Employ_temps (id_emploi, libelle,\$id-section,\$id-semester)

Ensg_enseigne_module(id_module, id_engs).

IV.7.2 Modèle logique du site service budget :

Exercice (id_exe, année, date_debut, date_fin)

Budget (id_budg, désignation, type, \$id_exe)

Section (id_sec, désignation, num,\$id_budg)

Chapitre (id_chap, désignation,\$id_sec)

Article (id_art, désignation,\$id_chap)

Dépense (id_deps, date, désignation, montant, \$id_année, \$id_dépense)

Régularisation(id_regul,libellé ,date_reg)

Année_dep(id_année, année_univ, année_budg)

Mandat(id_mandat, date, désignation, montant, \$id_dépense, id_regul)

Net_Grade(id_grade, grade, net, S.S, IRG,brut)

IV.7.3 Modèle logique du site faculté :

Département(id_dep, nom_dep, adresse_dep,)

Service (id_service, nom_service, adresse_service, section, \$id_admin_ser)

Admin_dep(id_admin_dep, nom_admin, prénom_admin, login, mot_passe, \$id_dep)

Admin_service(id_admin_serv, nom_admin, prénom_admin, login, mot passe, \$id_service)

Message (id_message, destinataire, destinateur, objet, message, date,
\$id_admin_dep, \$id_admin_serv)

IV.8 Les Bases de données :**IV.8.1 Base de données de département :**❖ **Table enseignants**

Nom de champ	Type de champ	Description	Clef
<u>ID_ensID</u>	int(11)	Identifiant de l'enseignant	Primaire
nomEns	varchar(45)	Nom de l'enseignant	
prenomEns	varchar(45)	Prenom de l'enseignant	
Departement	varchar(15)	Département de l'enseignant	
adrEns	Text	Adresse de l'enseignant	
statuEns	varchar(45)	Situation de l'enseignant	
gradeEns	varchar(45)	Grade de l'enseignant	
Email	varchar(20)	Email de l'enseignant	
Tel	int(11)	Téléphone de l'enseignant	
num_compte	int(50)	Numéro de l'enseignant	
Filiere_engs	varchar(11)	Filière de l'enseignant	
Specialite_engs	varchar(11)	Spécialité de l'enseignant	
Login	varchar(11)	Login	
mp	varchar(11)	Mot de passe	

❖ **Table Etudiant :**

Nom de champ	Type de champ	Description	Clef
<u>ID_etudiant</u>	bigint(20)	Identification de l'étudiant	Primaire
nomEtud	varchar(40)	Nom de l'étudiant	
prenomEtud	varchar(40)	Prenom de l'étudiant	
dateNaissanceEtud	Datetime	Date de naissance de l'étudiant	
adresseEtud	varchar(40)	Adresse de l'étudiant	
Groupe	Text	Groupe de l'étudiant	
Departement	smallint(6)	Département de l'étudiant	
Filiere	varchar(11)	Filière de l'étudiant	
Specialite	varchar(11)	Spécialité de l'étudiant	
Tel	varchar(11)	Teéléphone de l'étudiant	
Mail	int(11)	Email de l'étudiant	
Login	varchar(15)	Login	
mp	varchar(11)	Mot de passe	

❖ **Table etat_vacation :**

Nom de champ	Type de champ	Description	Clef
<u>id_etat</u>	int(11)	Identifiant de l'état de vacation	Primaire
type_activite	varchar(11)	Type d'activité	
Nbrcour	int(11)	Nombre de cours	
Nbrtd	int(11)	Nombre de TD	
ID_Ens	smallint(6)	Identifiant de l'enseignant	Etrangère

❖ **Table Filière :**

Nom de champ	Type de champ	Description	Clef
<u>ID_fil</u>	smallint(6)	Identifiant de la filière	Primaire
nomFil	varchar(40)	Nom de la filière	
departement	varchar(20)	Département	

❖ **Table groupe :**

Nom de champ	Type de champ	Description	Clef
<u>ID_groupe</u>	smallint(6)	Identifiant de groupe	Primaire
groupe	varchar(40)	Groupe	
sections	smallint(6)	Sections de groupe	Etrangère

❖ **Table Module :**

Nom de champ	Type de champ	Description	Clef
<u>ID_mod</u>	int(11)	Identifiant de module	primaire
nomMod	varchar(40)	Nom de module	
semestre	smallint(6)	Semestre	

❖ **Table Section :**

Nom de champ	Type de champ	Description	Clef
<u>ID_sec</u>	smallint(6)	Identifiant de la section	Primaire
section	varchar(40)	Section	
annetud	smallint(6)	Année d'étude	

❖ **Table Semestre :**

Nom de champ	Type de champ	Description	Clef
<u>ID semestre</u>	smallint(6)	Identifiant du semestre	Primaire
Id_semestr	varchar(6)	Semestre	Etrangère
Id_sections	smallint(6)	Section	Etrangère

❖ **Table Spécialité :**

Nom de champ	Type de champ	Description	Clef
<u>id_spec</u>	smallint(6)	Identifiant de la spécialité	Primaire
spécialité	varchar(11)	Spécialité	
id_fil	smallint(6)	Identifiant de la filière	Etrangère

❖ **Table Emploitemps :**

Nom de champ	Type de champ	Description	Clef
<u>ID Emploi</u>	smallint(6)	Identifiant de l'emploi du temps	Primaire
Libelle	varchar(40)	Libellé de l'emploi du temps	
Id_emestre	smallint(6)	Semestre	Etrangère
Id_section	smallint(6)	Section	Etrangère

❖ **Table Ens_mod :**

Nom de champ	Type de champ	Description	Clef
Code indexe	Int(11)	Code de l'index	Primaire
<u>ID_ens</u>	int(11)	Identifiant de l'enseignant	
<u>ID_mod</u>	int(11)	Identifiant du module	

IV.8.2 Base de données du service budget et comptabilité :**❖ Table Administrateur :**

Nom du champ	Type de donnée	Description	Clef
Id-admin	smallint(6)	Identification de l'administrateur	Primaire
Nom_admin	varchar(11)	Nom de l'administrateur	
Prenom_admin	varchar(11)	Prenom de l'administrateur	
Login	varchar(11)	Login de l'administrateur	
Motpasse	varchar(11)	Mot de passe de l'administrateur	
Type	varchar(11)	Type de l'administrateur	

❖ Table Budget :

Nom de champ	Type de donnée	Identification	Clef
<u>id_budget</u>	smallint(6)	Identification de budget	primaire
Designation	varchar(11)	Désignation de budget	
type_bg	varchar(11)	Type de budget	
id_exe	smallint(6)	Identification de l'exercice	Etrangère

❖ Table Chapitre :

Nom de champ	Type de donnée	Identification	Clef
<u>id_chap</u>	smallint(6)	Identification du chapitre	Primaire
designation_chap	varchar(11)	Désignation du chapitre	
id_section	smallint(6)	Identification de la section	Etrangère

❖ **Table Section :**

Nom de champ	Type de champ	Identification	Clef
<u>id_sec</u>	smallint(6)	Identification de la section	Primaire
désignation_sec	varchar(11)	Désignation de la section	
num_sec	smallint(6)	Numéro de la section	
id_budg	smallint(6)	Identification du budget	

❖ **Table Exercice :**

nom de champ	type de champ	Identification	Clef
id_exe	smallint(6)	Identification de l'exercice	Primaire
annee	smallint(6)	Année	
date_debut	Date	Date début de l'exercice	
date_fin	date	Date fin de l'exercice	Etrangère

❖ **Table Régularisation :**

Nom de champ	Type de champ	Identification	Clef
<u>id_reg</u>	smallint(6)	Identification de la régularisation	Primaire
Date_regl	date	date de la régularisation	

❖ **Table Dépense :**

Nom de champ	Type de champ	Identification	clef
<u>id_deponse</u>	smallint(6)	Identification de la deponse	Primaire
Date	Date	Date de la deponse	
designation_dep	varchar(11)	Désignation de la dépense	
annee_budg	smallint(6)	Année budgétaire	
annee-univ	smallint(6)	Année universitaire	

❖ **Table Mandat**

Nom de champ	Type de champ	Identification	Clef
<u>id_mandat</u>	smallint(6)	Identification du monda	Primaire
date_mondat	Date	Date du monda	
designation_mondat	varchar(11)	Désignation du monda	
Montaux	float	Montaux	
id_deponse	smallint(6)	Identification de la deponse	
id_regl	smallint(6)	Identification de la régularisation	Etrangère

❖ **Table Net- Grade :**

Nom de champ	Type de champ	Identification	Clef
<u>id_grade</u>	smallint(6)	Identification de grade	Primaire
Grade	varchar(11)	Grade	
Brut	Float	Brut	
SS	Float	Sécurité sociale	
I.R.G	Float	I.R.G	
Net	float	net	

IV.8.3 Base de données faculté :❖ **Administrateur :**

Nom de champ	Type de champ	Identification	Clef
<u>Id_admin</u>	smallint(6)	Identifiant de l'administrateur	Primaire
nom_admin	varchar(11)	Nom de l'administrateur	
prenom_admin	varchar(11)	Prénom de l'administrateur	
type_admin	varchar(11)	Type de l'administrateur	
login_admin	varchar(11)	Login de l'administrateur	

❖ **Table de la vue matérialisée enseignants -fac**

Nom de champ	Type de champ	Identification	Clef
<u>id_ens</u>	Smallint(6)	Identifiant de l'enseignant	Primaire
nom_ens	varchar(11)	Nom de l'enseignant	
prenom_ens	varchar(11)	Prénom de l'enseignant	
grade	varchar(11)	Grade de l'enseignant	
dep	varchar(11)	Département de l'enseignant	
email	varchar(20)	Email d l'enseignant	
tel	int(11)	Téléphone de l'enseignant	

❖ **Table de la vue matérialisée étudiant- faculté :**

Nom de champ	Type de champ	Identification	Clef
<u>id_etud_fac</u>	smallint(6)	Identifiant de l'étudiant faculté	Primaire
nom_etud	varchar(11)	Nom de l'étudiant	
prenom_etud	varchar(11)	Prénom de l'étudiant	
faculte	varchar(11)	Faculté	
departement	varchar(11)	Département de l'étudiant	
filiere	varchar(11)	Filière de l'étudiant	
specialite	varchar(11)	Spécialité de l'étudiant	
annee_cour	varchar(11)	Année en cour	

❖ **Table de la vue matérialisée des états de vacation des enseignants de la faculté :**

Nom de champ	Type de champ	Identification	Clef
<u>id_vue_M</u>	smallint(6)	Identification de la vue matérialisée	Primaire
nom_engg	varchar(11)	Nom de l'enseignant	
prenom_engg	varchar(11)	Prénom de l'enseignant	
Grade	varchar(11)	Grade de l'enseignant	
Département	varchar(11)	Département de l'enseignant	
type_activ	varchar(11)	Type d'activité de l'enseignant	
nbr_h	smallint(6)	Nombre d'heurs total	
Mois	varchar(11)	Mois	
Brut	Float	Brut	
Ss	float	Sécurité sociale	
Irg	Float		
Net	Float	Net à payer	
valider	varchar(11)	Validation	

❖ **Table des administrateurs des départements :**

Nom de champ	Type de champ	Description	Clef
<u>id_admin_dep</u>	int(6)	Identifiant de l'administrateur de département	Primaire
Nom	varchar(20)	Nom de l'administrateur	
Prenom_admin	varchar(20)	Prénom de l'administrateur	
departement_admin	varchar(20)	Département de l'administrateur	

❖ **Table des administrateurs d'E-services communs :**

Nom de champ	Type de champ	Description	Clef
<u>id_admin_ser</u>	int(11)	Identifiant de l'administrateur	Primaire
nom_admin	varchar(20)	Nom de l'administrateur	
prenom_admin	varchar(20)	Prénom de l'administrateur	
nom_service	varchar(20)	Nom de service	
Section	varchar(20)	Section	

❖ **Table département :**

Nom de champ	Type de champ	Description	Clef
<u>id_dep</u>	int(6)	Identifiant de département	Primaire
nom_dep	varchar(20)	Nom de département	
adresse_dep	Text	Adresse de département	

❖ **Table service communs**

Nom de champ	Type de champ	Description	Clef
id_ser	int(11)	Identifiant de service	Primaire
Nom	varchar(20)	Nom de service	
adresse	texte	Adresse de service	

Conclusion

Ce chapitre est consacré à l'analyse et à la conception de l'application avec le langage UML.

Nous avons présenté le processus de conception de notre application en deux niveaux, le niveau applicatif et le niveau de données. En premier lieu, nous avons commencé l'analyse et la conception par le niveau applicatif qui concerne les fonctionnalités et les traitements de l'application, ensuite nous sommes passés au niveau de données qui nous a permis d'avoir le modèle logique de la base de données de chaque site et aussi nous avons présenté la procédure d'intégration de données d'une base à l'autre qui nous a permis de compléter le schéma globale.

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

Chapitre V : *Réalisation*

Introduction:

Après avoir présenté dans le chapitre précédent les différentes étapes de conceptions, nous allons passer dans ce présent chapitre à la traduction de ces dernières en code source.

Nous commençons d'abord par la présentation de l'environnement de développement et les outils qui serviront à la réalisation de notre application, puis nous présentons l'application développée à travers ces interfaces prise en captures d'écrans.

V.1 Architecture de notre application :

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

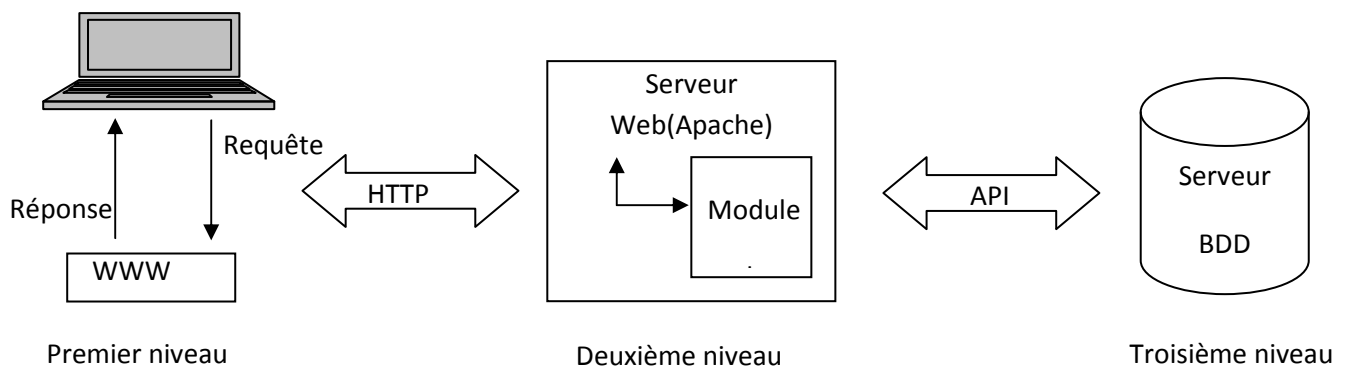


Figure V.1 : L'architecture du déploiement de l'application

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

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

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

V.2 Environnement de développement et d'implémentation:

V.2.1 Les langages de programmation :

Durant notre développement de l'application nous avons utilisé les langages de programmation suivants :

- **HTML**(Hyper TextMarkupLanguage) : est un langage hypertexte à balises, ces dernières permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents. Cela veut dire qu'il va gérer la façon dont un texte va s'afficher au sien du navigateur. L'HTML est souvent utilisé conjointement avec d'autre langage de programmation tel que le JavaScript et aussi les formats de présentation CSS (Feuilles de style en cascade).

Exemple de code HTML :

```
<html>
<head>
L'entête de la page
<title>Ma page web</title>
</head>
<body>
Corps de la page
</body>
</html>
```

- **CSS (en anglais "Cascading Style Sheets", abrégé CSS)** Les feuilles de styles (*feuilles de style en cascade*) sont un langage qui permet de gérer la présentation d'une page Web. Le principe des feuilles de style consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments. Il suffit de définir par un nom un ensemble de définitions et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte. Les feuilles de style ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation.
- **Langage de requête SQL** : Pour l'élaboration des requête d'interrogation de la base de données on utilise le langage SQL. C'est un langage de manipulation de base données mis au point dans les années 70 par IBM, il permet notamment :
- ✓ La manipulation des tables : création, suppression, modification de la structure des tables
 - ✓ La manipulation des données: sélection, modification et suppression d'enregistrement
 - ✓ La gestion des droits d'accès aux tables : contrôle des données et validation des modifications
- **JavaScript** : Est un langage développé par Netscape, c'est une extension du code html c'est-à-dire que les scripts sont ajoutés aux balises html et s'exécutant dans le navigateur, sans faire appel aux ressources du serveur. Le JavaScript permet d'effectuer des contrôles de saisie pour valider les formulaires, ouvrir ou fermer des fenêtres ou encore de gérer des éléments graphiques.

Les balises utilisées pour écrire un code JavaScript sont les suivantes :

<**SCRIPT** language= javascript>

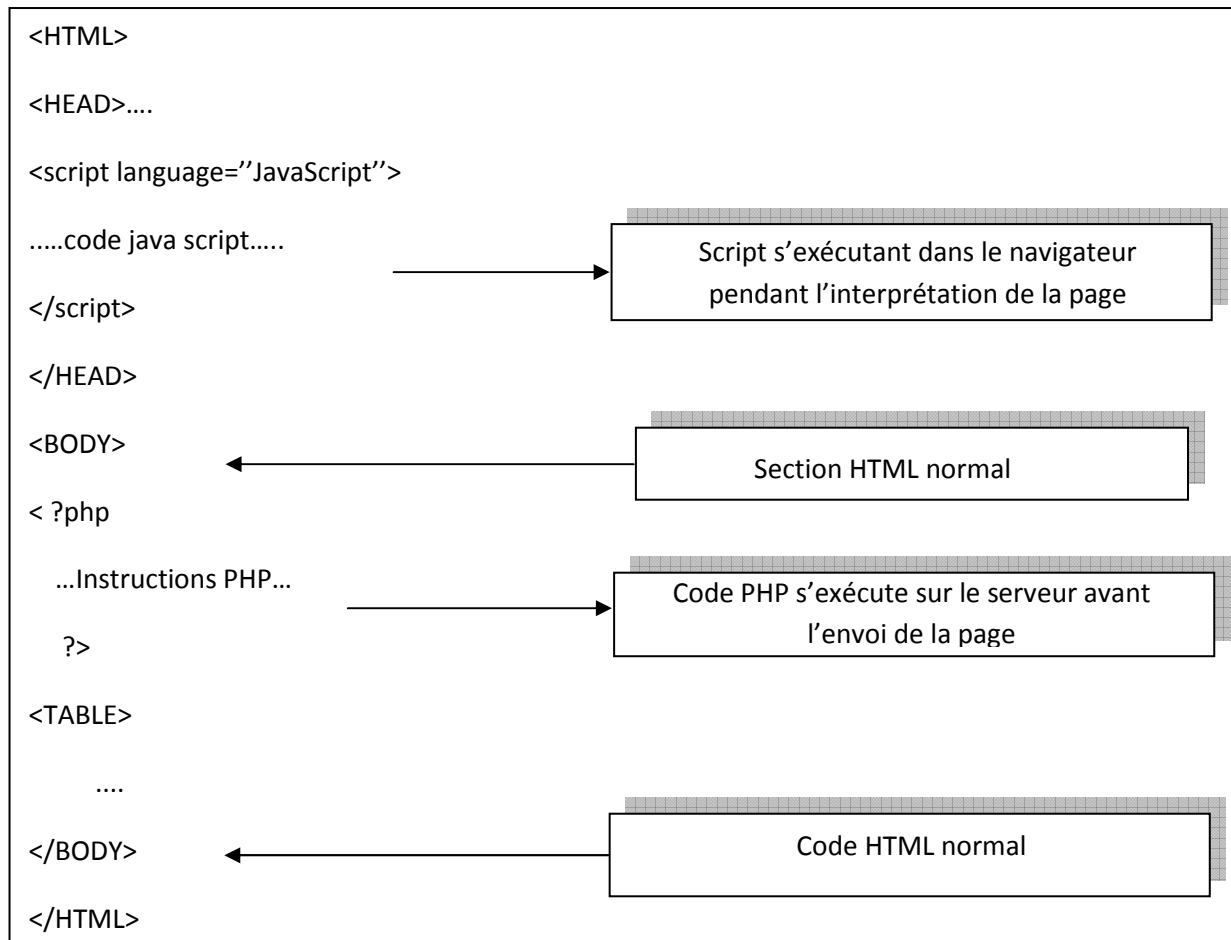
Instructions du code

</**SCRIPT**>

- **PHP** : Pour la partie dynamique du site. Il constitue le langage pilier de l'application, c'est-à-dire, que c'est sur lui que reposent le dynamisme de l'application, c'est la raison pour laquelle nous allons mieux le détailler dans ce qui suit :

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

- PHP est facile à utiliser, il permet de créer avec un minimum d'efforts des pages Web dynamiques destinées aux applications Internet multimédia et de E-commerce.
- PHP est multi plate forme, il existe pour les différentes versions de Windows, Unix et Linux, ainsi que pour de nombreux serveurs Web dont APACHE et IIS.
- PHP est libre, donc « OPEN SOURCE », on peut ajouter de nouvelles fonctionnalités sans attendre une nouvelle version.
- PHP utilise le moteur de scripts Zend, qui améliore la rapidité de traitement
- PHP a été conçu pour fonctionner sur le Web, la connexion et l'interrogation d'une base de données sont extrêmement simple (peuvent être accomplies en deux ou trois lignes de code).
- La plus grande qualité et le plus important avantage du langage PHP est le support d'un grand nombre de bases de données (plus de 20). PHP possède des fonctions ODBC, permettant de s'interfacer de manière conviviale à toute base de données ODBC.
- Avec plus de 1200 fonctions utilisables dans des applications très variées, PHP couvre pratiquement tous les domaines en rapport avec les applications Web.

Structure d'une page PHP :

Remarque : Il faut noter la différence entre :

- **Les autres langages de scripts coté serveur** tels que les servelets Java, qui doivent écrire un programme avec de nombreuses lignes de commande afin d'afficher une page HTML, alors que on écrit une page HTML avec du code PHP inclus à l'intérieur.
- **Les autres langages de script coté client** comme le Java script sont exécutées par le navigateur, alors que le code PHP est exécuté sur le serveur. Si on a un script PHP sur un serveur, le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat.

Fonctionnement de PHP :

Le serveur web reconnaît à l'extension des fichiers, que le document appelé par le client est une page qui comporte du code PHP qui diffère d'une simple page HTML, en conséquence.

- Le serveur Web lance l'interpréteur PHP ;
- L'interpréteur PHP traduit le document demandé et exécute le code source de la page ;
- Les commandes figurant dans la page sont interprétées et le résultat prend la forme d'une page HTML publiée à la place du code source dans le même document ;
- La page modifiée est envoyée au client pour y être affichée par le navigateur.

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

V.2.2 Serveur web Apache :

Apache est le plus populaire des serveurs Internet. Il a été dans la volonté de développer et de manipuler un serveur http sécurisé, efficace et évolutif pour les systèmes d'exploitation modernes.

Apache est multi plate forme et gratuit, son installation est facile, rapide et son utilisation pas très compliqué, Grace à une association avec PHP apache devient un serveur web dynamique

V.2.3 Serveur MySQL :

MYSQL est un véritable serveur de base de données SQL (StructuredQueryLanguage) qui est un langage de requêtes vers les bases de données exploitant le modèle relationnel. Il est configuré selon l'architecture client/serveur qui est souvent utilisée avec le langage de création de pages Web dynamiques PHP. De plus MYSQL est un gestionnaire de base de données SQL multitraitement, il est caractérisé par sa rapidité et sa robustesse.

V.2.4 PHPMyAdmin :

L'outil PhpMyAdmin est développé en PHP (ensemble de scripts PHP), il offre une interface graphique pour l'administration des bases de données MYSQL via un navigateur Web. Les fonctions principales de phpMyAdmin sont :

- ✓ Création de nouvelles bases de données ;
- ✓ Création/suppression/modification des tables ;
- ✓ L'édition, l'ajout et la suppression de champs ;
- ✓ L'exécution de commandes SQL et de requêtes ;
- ✓ Gérer les privilèges des utilisateurs.

V.2.4.1 Implémentation des bases de données :

Pour implémentation des bases de données de notre application, on a utilisé l'outil PHPMyAdmin fourni avec le logiciel Wampserver. Son interface graphique, nous a permis de créer et de gérer l'ensemble des bases de données et les tables associés.

La figure suivante illustre l'interface de PHPMyAdmin avec l'ensemble de bases utilisées :

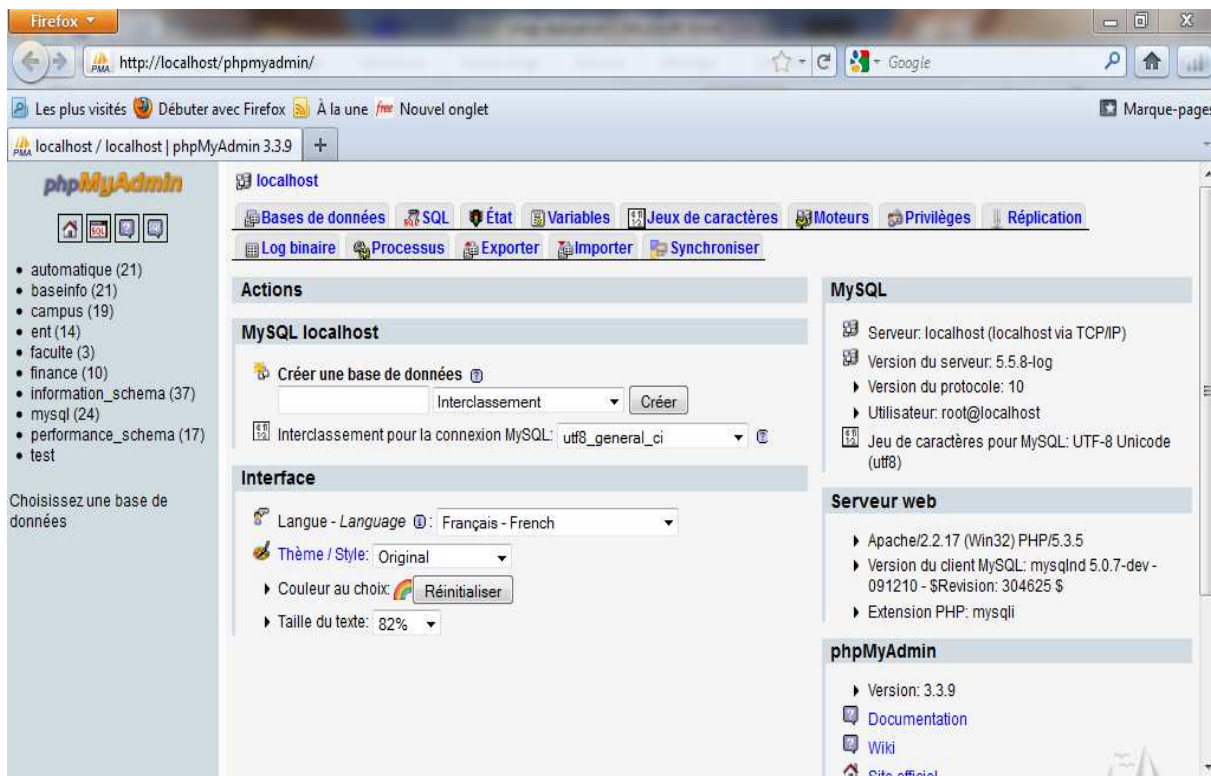


Figure V.2 : Interface de PHPMyAdmin

V.2.5 Les logiciels :

V.2.5.1 WAMP (Windows, Apache, MySQL, PHP):

Afin de faciliter l'installation et la configuration des outils précédemment définis on a opté pour l'utilisation de WampServer, qui permet d'installer à la fois : le serveur web apache, le serveur de base de données MySQL et le PHP offrant ainsi la possibilité de tester les pages en PHP localement. WampServer joint également PhpMyAdmin à MySQL permettant ainsi la gestion de la base de données que se soit par interface graphique ou par exécution des instructions SQL.

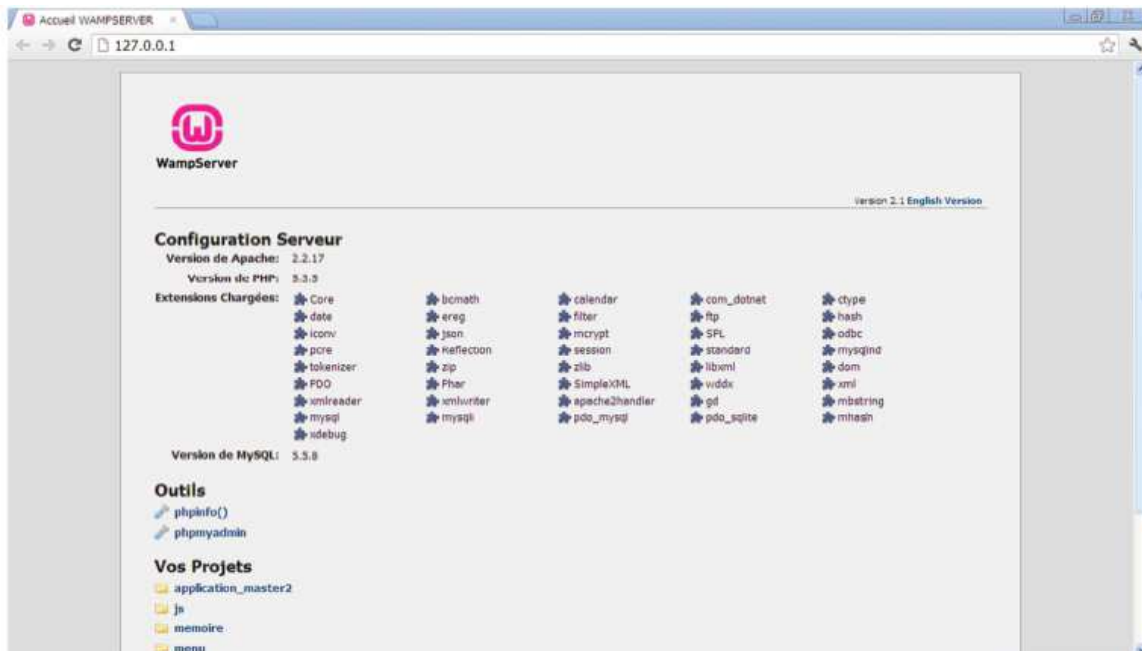


Figure V.3 : Interface de WampServer

V.2.5.2 Macromedia Dreamweaver 8 :

Est un système professionnel par excellence pour la création de sites web et d'applications. Sa puissante combinaison d'outils de mise en forme visuelle, de fonctions de développement d'applications et d'édition de code permet aux développeurs et aux concepteurs de créer des sites et des applications visuellement attrayantes et normalisés.

Qu'il s'agisse de la prise en charge des concepts basés sur CSS ou de fonctionnalités de codage manuel, Dreamwear fournit aux professionnels les outils dont ils ont besoin un environnement intégré et optimisé.

L'interface du logiciel macro média Dreamwear est illustrée dans la figuresuivante :



Figure V.4 : Interface Dreamweaver

V.3 présentation de quelques interfaces de notre application :

1. La page d'accueil

C'est la première page visité par le visiteur.



Figure V.5 : page d'accueil du campus

2. La page d'authentification « cas étudiant » :

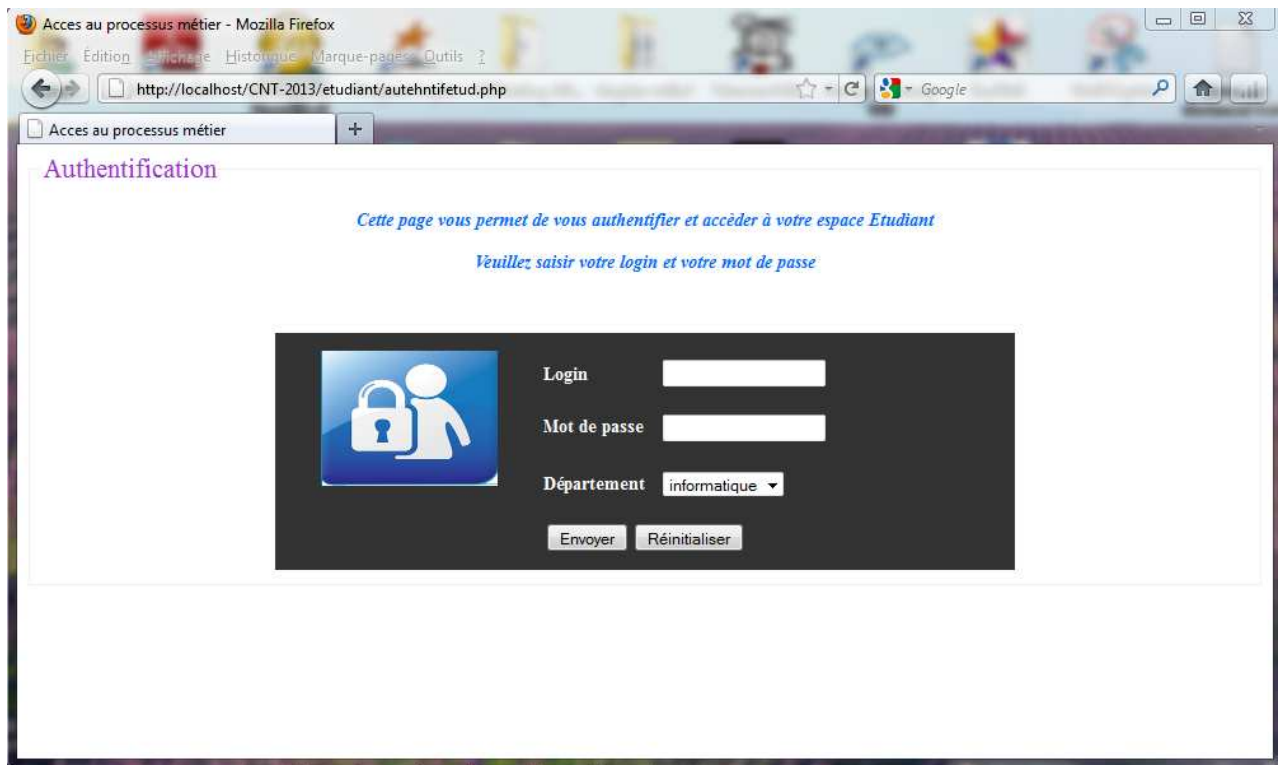


Figure V.6 : Page d'authentification

3. Inscription enseignant :

L'enseignant s'inscrit au campus, suit à cette inscription l'enseignant est enregistré dans la base de données de son département et au niveau de la faculté aussi.

Firefox

inscription enseignants

Adresse

Telephone

Numéro de compte

Informations sur enseignement

Faculté (*)

Département (*)

Grade (*)

Filière enseignée

Spécialité enseignée

Professeur

Mot de passe de validation

Email (*)

Identifiant (*)

Mot de passe (*)

Confirmez votre mot de passe (*)

Rinitialiser

VALIDER VOTRE INSCRIPTION

Figure V.7 :page d'inscription de l'enseignant

4. Inscription étudiant :

Même procédure que l'enregistrement de l'enseignant, l'étudiant dès son inscription, il sera enregistré dans la base de données de son département ainsi dans la base faculté.

The screenshot shows a web browser window with the address bar displaying 'inscription enseignants'. The page contains three main sections for data entry:

- Vos coordonnées**: Fields for Nom (*), Prénom (*), Date de naissance (*), Adresse, and Téléphone.
- Informations sur votre cursus universitaire**: Fields for Faculté (*), Département (*), Filière, Spécialité, Année en cours, N° de section, N° de groupe, and N° de la carte d'étudiant.
- Authentification & validation**: Fields for Email (*), mot de passe (*), Répéter le mot de passe (*), and Confirmez votre email de poste (*).

At the bottom of the form, there are two orange buttons: 'Retour' and 'VALIDER VOTRE INSCRIPTION'.

Figure V.8 : page d'inscription étudiant

5. Page de site faculté de GEI :

La page faculté permet d'accéder aux différents sites département et aussi aux services communs de la faculté.

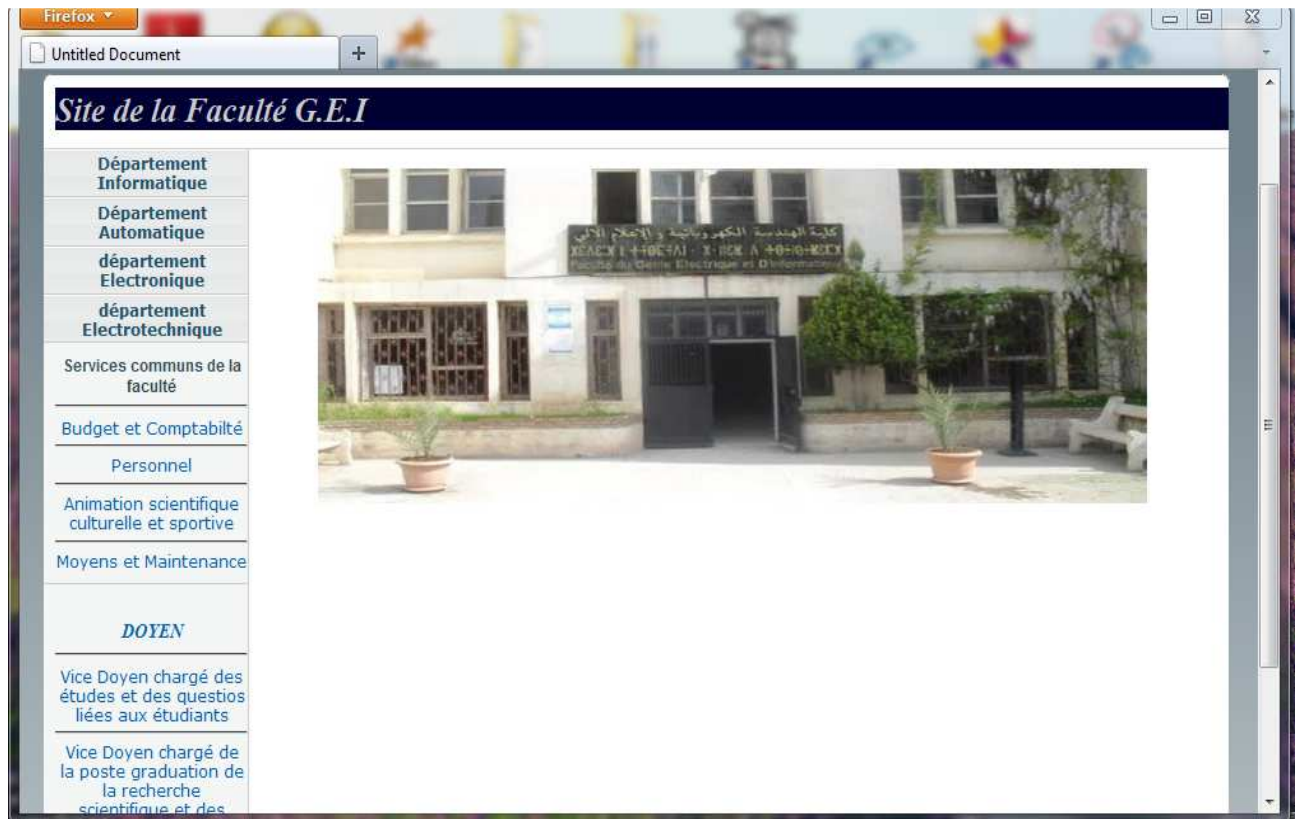


Figure V.9 : Page de site faculté GEI

6. Authentification du chef de section budget :

Cette page permet au chef de section budget d'accéder à son espace.



Figure V.10 : Page d'authentification de chef de section budget

7. L'espace de chef de section budget :

Le chef de section budget trouvera dans son espace toutes les applications administratives qu'utilise.

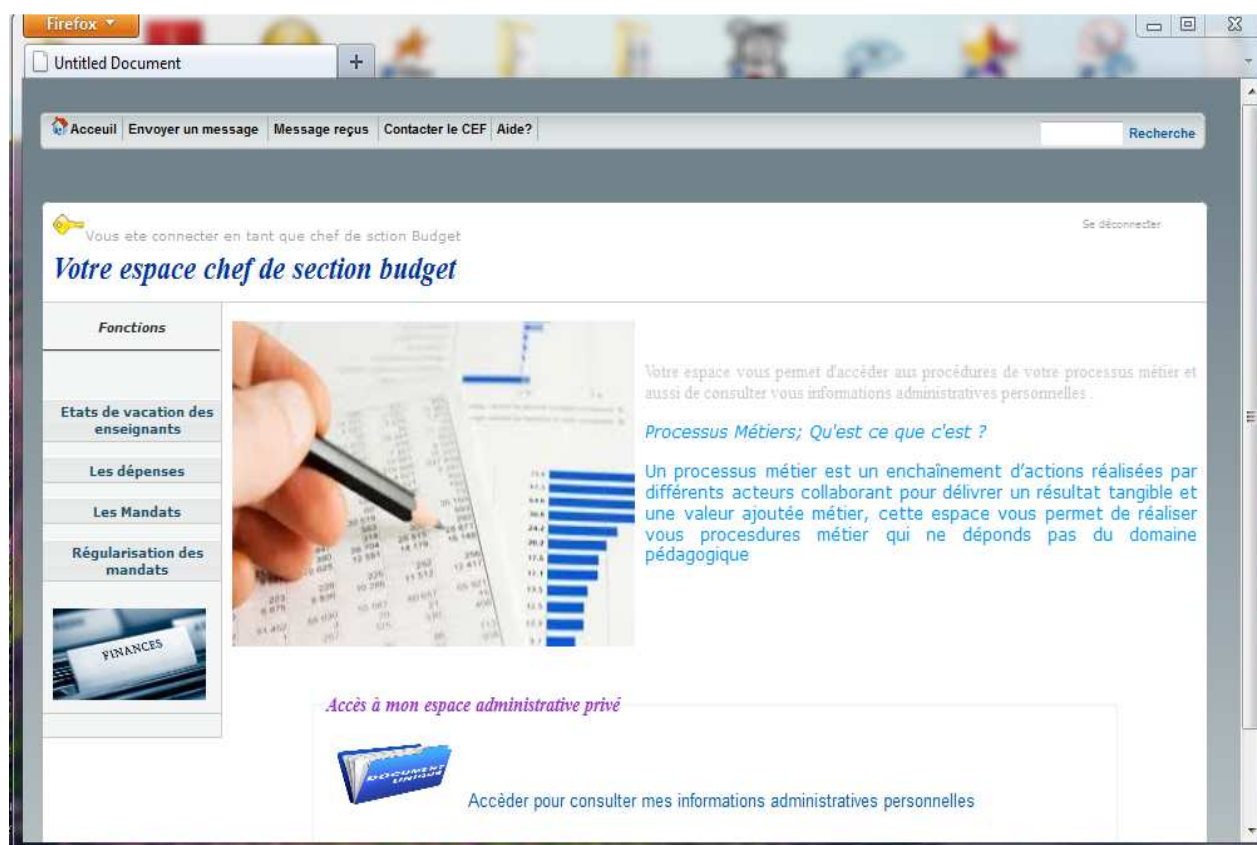


Figure V.11 : Espace de chef de section budget

8. Page de gestion des états de vacation :

Dans cette page permet au chef de section budget de gérer les états de vacation des enseignants de la faculté GEI.

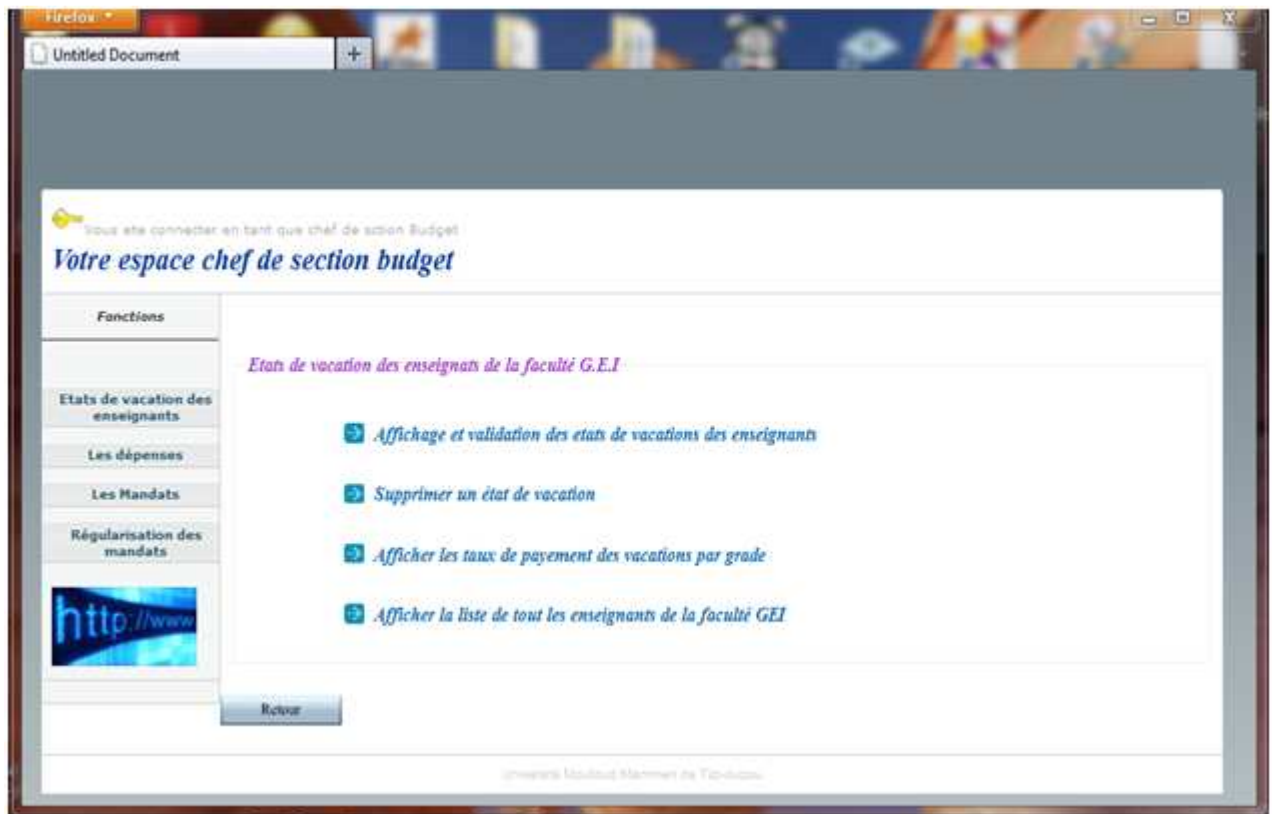


Figure V.12 : Gestion des états de vacation des enseignants de la faculté GEI

9. Affichage des états de vacation des enseignants de la faculté FEI

Dans cette page le chef de section budget va consulter tous les états de vacation des enseignants de la faculté.

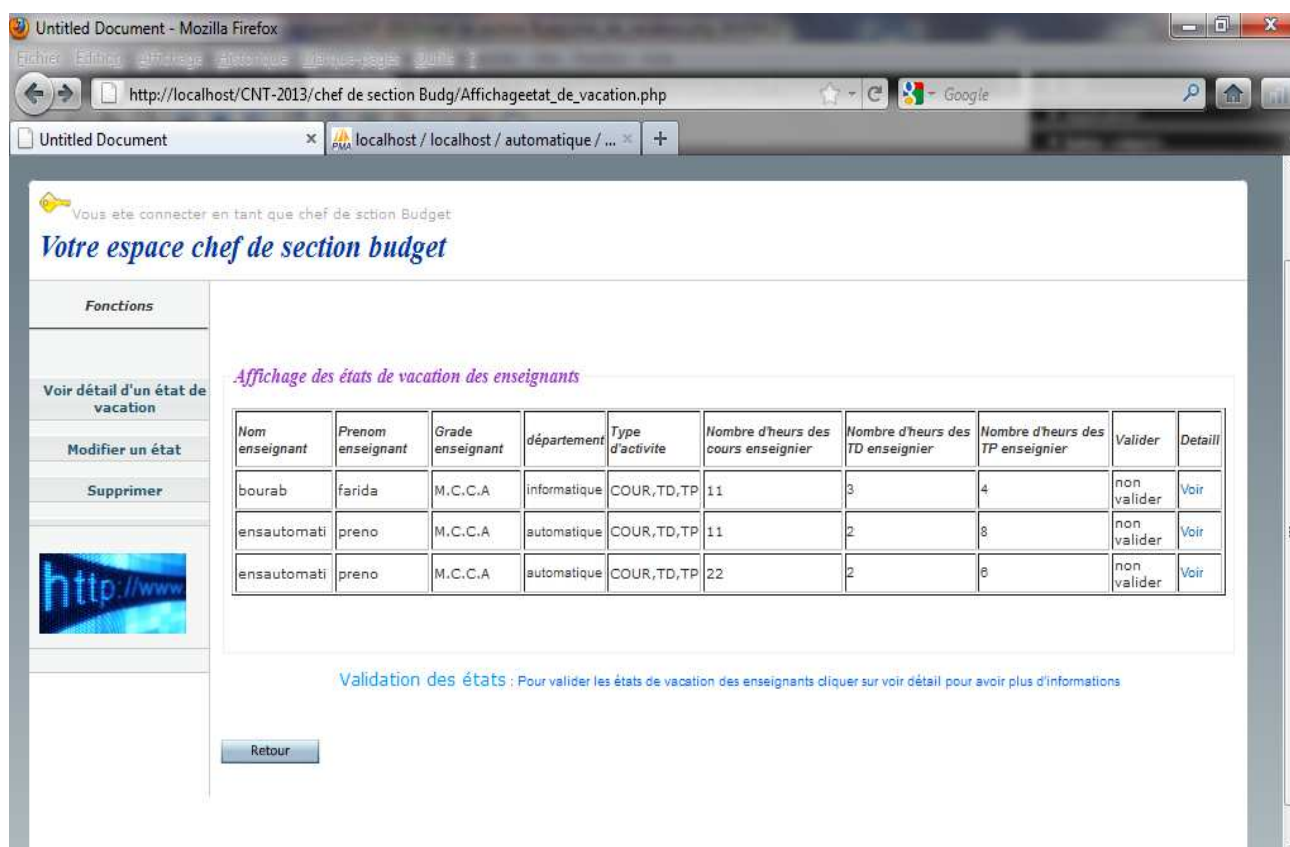


Figure V.13 : Affichage des états de vacation des enseignants

10. Page d'E-faculté Administration

L'espace d'E-faculté admin permet à l'administrateur de gérer le site faculté tel que la gestion des enseignants et des étudiants de la faculté et aussi la gestion des services communs.

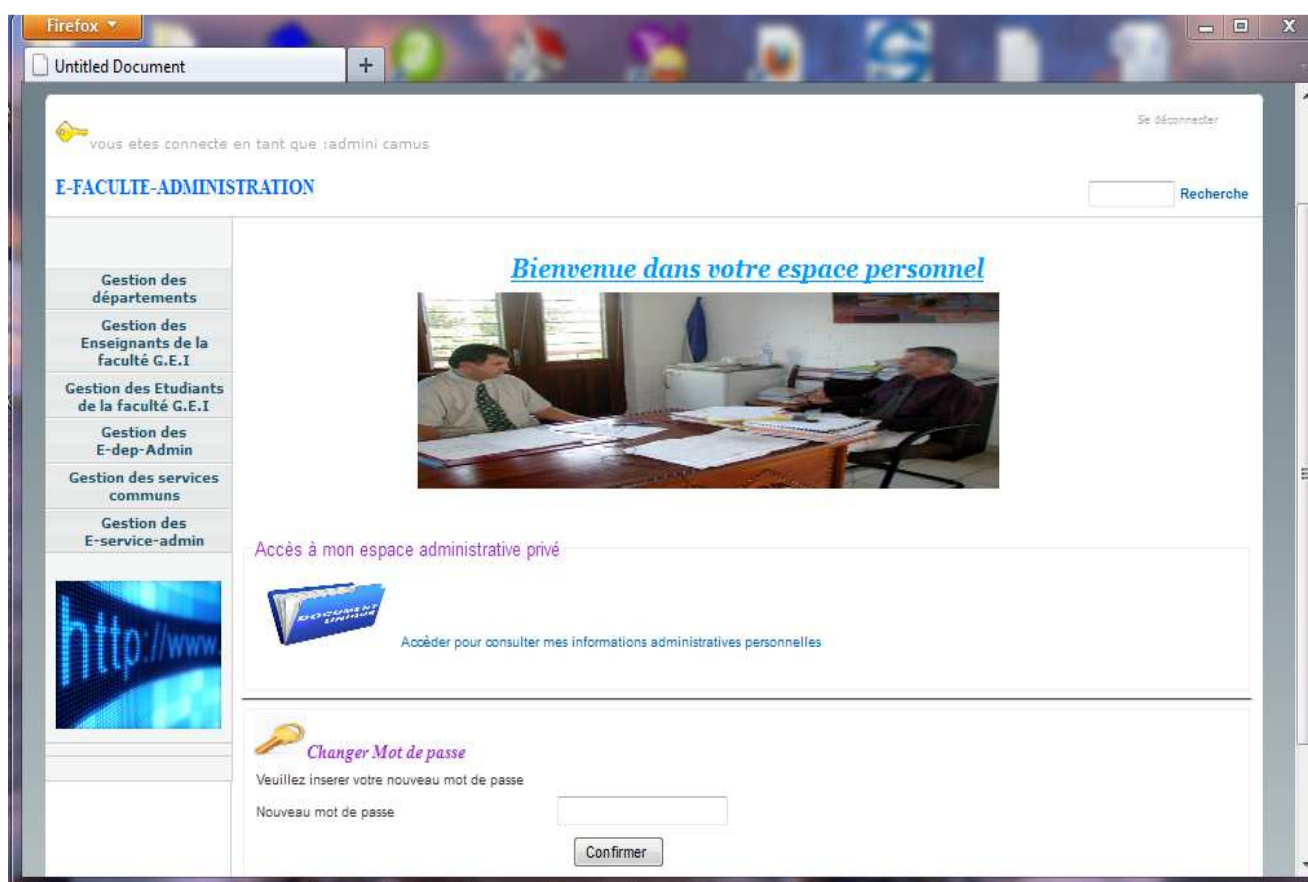


Figure V.14 : Espace d'E-faculté administration

11. Ajouter un département à la faculté :

Ajouter un département à la faculté GEI

Nom de département : Automatique

Adresse de département : Université Mouloud Mammeri Tizi-ouzou

Réinitialiser

Ajouter le département

Figure V.15 : Ajouter un département

12. Page de gestion des départements :

vous êtes connecté en tant que : admin@camus

E-FACULTE-ADMINISTRATION

Gestion des départements

La liste des départements de la faculté G.E.I

Code département	Département	Adresse	Modifier	Supprimer
1	informatique	université mouloud mammeri	Modifier	Supprimer
2	Automatique	Université Mouloud Mammeri Tizi-ouzou	Modifier	Supprimer

[Retour](#)

Figure V.16 : page de gestion des départements

13. Gestion des étudiants de la faculté :

Universit  Mouloud Mammeri de Tizi-ouzou

vous  tes connect  en tant que :admini camus [Se d connecter](#)

E-FACULTE-ADMINISTRATION

Fonctions

- Fonctions
- Modifier
- Supprimer

La liste des Etudiant de la facult  GEI

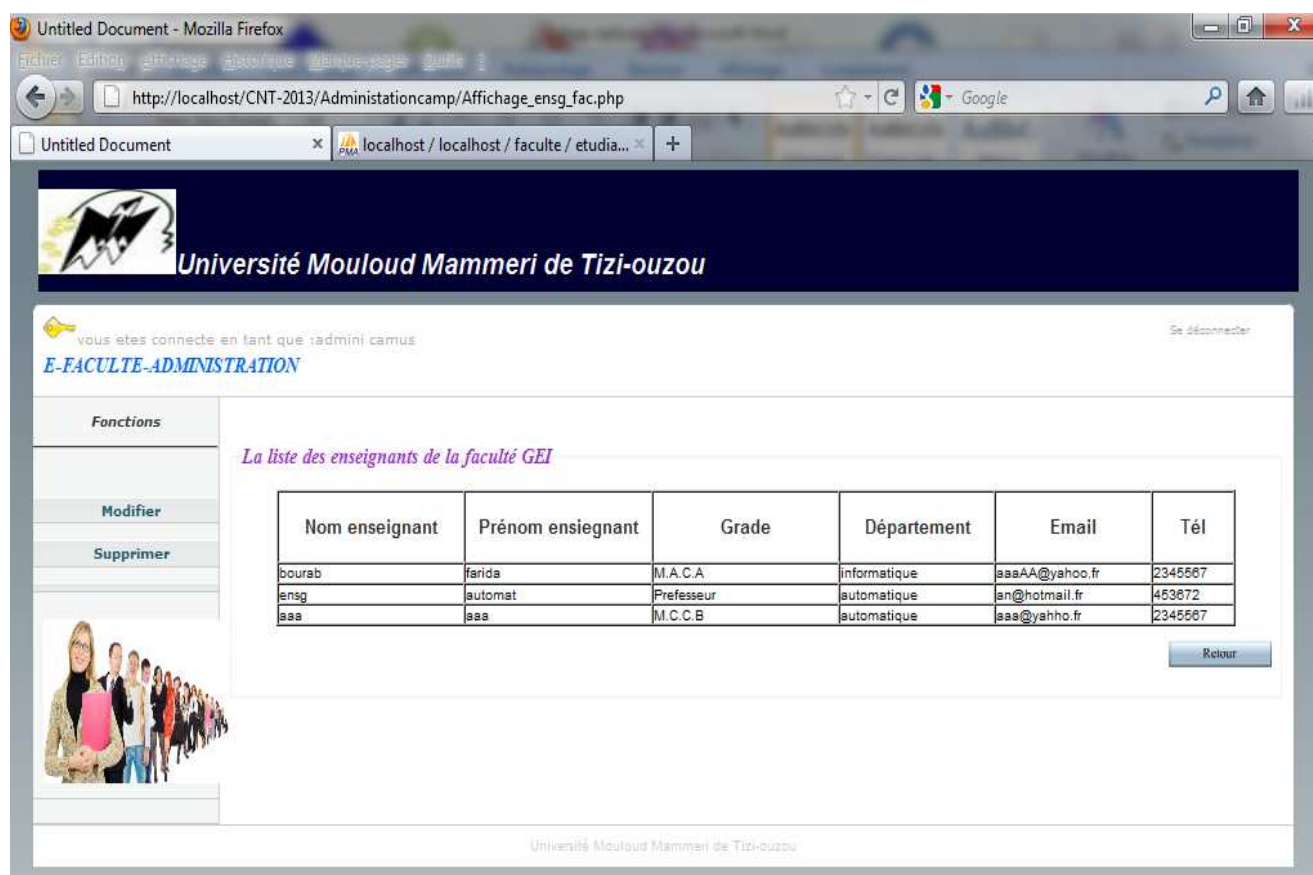
NOM	PRENOM	D�partement	fil�re	E-mail	T�l
etud1	automat1	automatique	automat	automat@hotmail	23455
ww	bb	informatique	info	www@yahoo.fr	5254
etud1	etud1	automatique	automat	auto@yahoo.fr	2345567

[Retour](#)

Universit  Mouloud Mammeri de Tizi-ouzou

Figure V.17: Gestion des  tudiants de la facult 

13. Page de gestion des enseignants de la faculté :



vous êtes connecté en tant que : admini camus

E-FACULTE-ADMINISTRATION

Fonctions

Modifier

Supprimer

La liste des enseignants de la faculté GEI

Nom enseignant	Prénom enseignant	Grade	Département	Email	Tél
bourab	farida	M.A.C.A	informatique	aaaAA@yahoo.fr	2345567
ensg	automat	Professeur	automatique	an@hotmail.fr	453672
aaa	aaa	M.C.C.B	automatique	aaa@yahoo.fr	2345567

Retour

Université Mouloud Mammeri de Tizi-ouzou

Figure V.18 : Gestion des enseignants de la faculté

14. L'espace enseignant :

L'espace réservé à l'enseignant lui permet d'accéder au module pédagogique et aux informations spécifiques aux départements, comme il pourra accéder à des fonctions administratives.

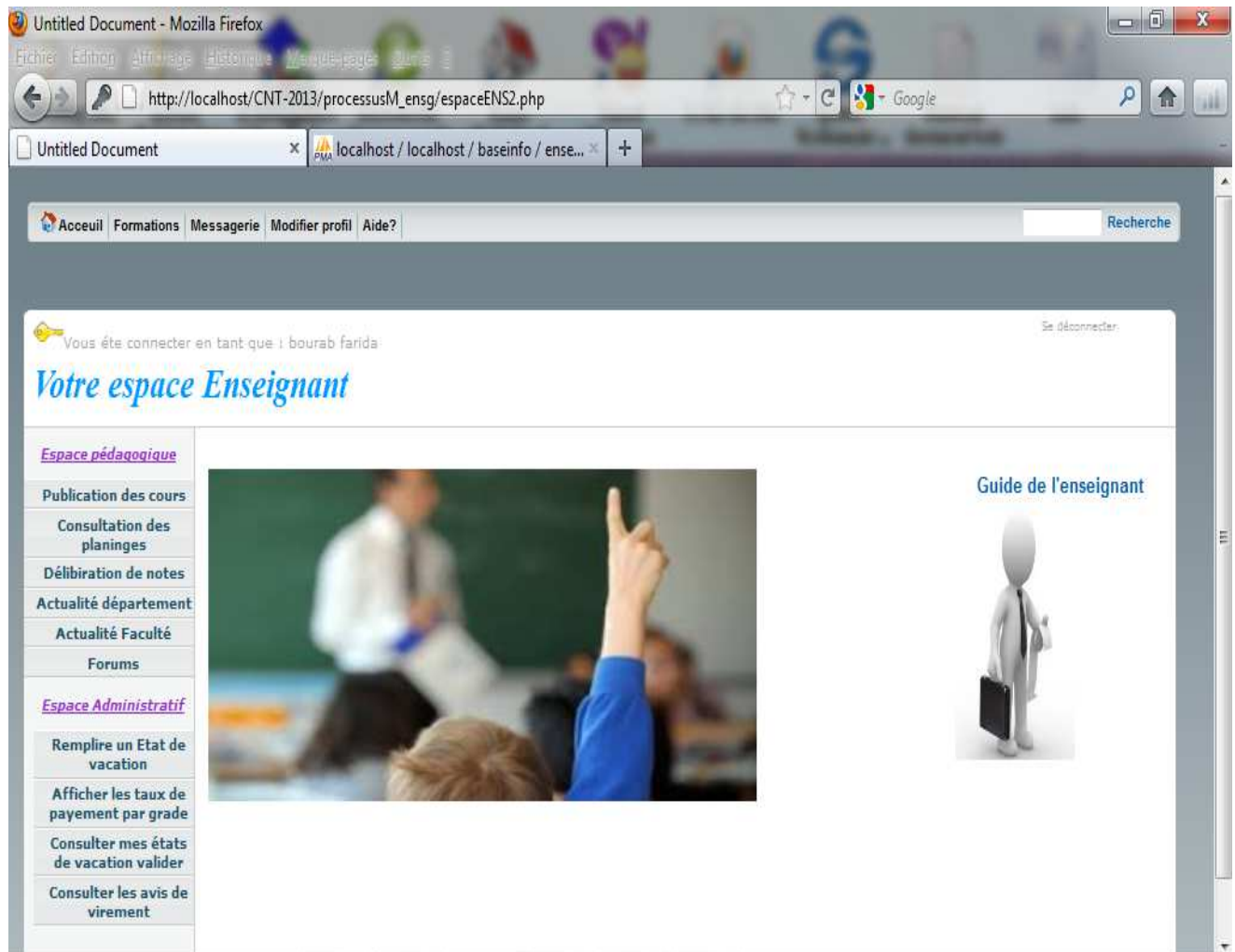


Figure V.19 : Espace enseignant

15. Remplir un état de vacation :

Parmi les fonctionnalités administratives que l'enseignant pourra exercer sur son espace enseignant est de remplir son état de vacation.

The screenshot shows a Mozilla Firefox browser window with the title "Remplir un Etat de vacation". The address bar displays the URL "http://localhost/CNT-2013/processusM_eng/ajoute_etaVac.PHP". The browser tabs include "Remplir un Etat de vacation" and "localhost / localhost / finance / v_de...". The main content area features a form with a header "Veuillez Remplir tous les champs de votre état de vacation". The form contains several fields: "Mois" (Month) with a dropdown menu showing "Janvier", "Type d'activité" (Activity type) with a dropdown menu showing "COUR", and three input fields for "Nombre d'heurs de cour enseigner", "Nombre d'heurs de TD enseigner", and "Nombre d'heurs de TP enseigner". At the bottom of the form, there are two orange buttons: "Réaliser" and "VALIDER VOTRE ETAT DE VACATION".

Figure V.20 : Remplir un état de vacation

16. Espace étudiant :

L'espace réservé à l'étudiant lui permet de consulter les données pédagogiques telles que les forums, les cours publiés, les affichages virtuels..., et aussi les informations administratives telles que les dossiers administratifs (bourse, résidences universitaires), la scolarité....



Figure V.21 : Espace étudiant

Conclusion :

Dans ce chapitre nous avons parlé des outils de développement utilisés pour implémenter notre application. Ensuite nous avons décrit le fonctionnement de notre application en présentant quelques interfaces de notre application.

Conclusión general

Conclusion Générale

Dans ce mémoire nous avons présenté une démarche pour mettre en œuvre la conception et l'implémentation d'une base de données distribuées pour l'espace numérique de travail de l'université Mouloud Mammeri de Tizi-Ouzou (UMMTO). Et nous avons aussi proposé une organisation générale et fonctionnelle pour ce dernier.

Pour ce faire, nous avons commencé par la présentation générale des espaces numériques de travail, ensuite les concepts de base de données réparties, ainsi les différentes techniques d'intégration des sources de données, par la suite nous avons passé à la phase de conception de notre base de données réparties, et au final nous avons présenté quelques interfaces de notre application.

La réalisation de ce travail a été très fructueuse, vu son apport considérable en matière de connaissances acquises, notamment sur les bases de données distribuées. Elle nous a permis, entre autre, d'effectuer des recherches, d'apprendre le formalisme UML, d'utiliser le SGBD MySQL, d'apprendre le langage de programmation PHP.

Ce travail peut être encore amélioré en envisageant les perspectives suivantes:

- ✓ Compléter et améliorer les fonctionnalités du campus par l'intégration de nouvelles plateformes tel que le E-Learning, l'affichage virtuel, gestion des emplois de temps ...etc.
- ✓ Compléter le processus de fédération des bases de données par l'extension de cette dernière vers le rectorat et l'inclusion de toutes les facultés d'UMMTO.
- ✓ Mettre en place des mécanismes de sécurité, par exemple un protocole d'authentification avancée ou un crypto-système pour les transferts de données à distance.

*Références
bibliographiques*

Bibliographie :

[1] :Les Espaces Numériques de Travail

Alain MAYEUR Septembre 2003

[2]:Schéma directeur des espaces numériques de travail Version 2.0

Ministère de l'éducation nationale, de l'enseignement supérieur et de la recherche
07/11/2006

[3] : Schéma directeur des espaces numériques de travail

12/01/2004

[6] :Mémoire master II « Développement d'un espace numérique de travail pour un département universitaire » BRAHAMI Saliha 2011/2012

[7]:Bases de données réparties, Présentation Générale

Matthieu ExbrayatLicence Pro Informatique2008-2009

[8] :Manipulations multibases et distribuées Partie 1WitoldLitwin2012

[9] : Bases de données Réparties, Fédérées et Réplication,Bernard ESPINASSE « Professeur à Aix-Marseille Université (AMU) Ecole Polytechnique Universitaire de Marseille » Avril 2013

[12] : Systèmes de Gestion de Bases de Données Réparties & Mécanismes de Répartition avec Oracle (partie I : les BDDs réparties) Rim Moussa 2005-2006

[13] :Mémoire Master II, Université A/Mira de Bejaia « Conception et réalisation d'une base de données repartie sous oracle : cas de l'hébergement des résidences universitaires »
Réalisé par Hakim MADI, année 2009

[19] : Apprendre SQL avec MySQL avec 40 exercices corrigés CHRISTIAN SOUTOU 2006

[20] :Entrepôt Distribué de Données Une méthodologie pour l'emplacement optimal des données dans les entrepôts distribués, Mr CHOUDER LAMRI,01/2007

Webographie :

[4]: <http://eduscol.education.fr/numerique/dossier/archives/eformation/notion-virtuel-numerique>

[5] :<http://fr.wikipedia.org>

[10]: <http://www.univ-mlv.fr/~ocure>

[11]:http://tice.univ-nc.nc/~taladoire/Pedagogie/RessourcesBD/EPFL/poly3_fichiers/15/15.html

[14] : www-lsr.imag.fr/Les.Personnes/Cyril.Labbe/M2P

[15] :<http://www.ibm.com>

[16] : <http://www.developpez.com/>

[17] : www.siteduzero.com

[18] :<http://www2.lirmm.fr/~libourel/MIFPRU/M2/Mediation.pdf>