



Mémoire

Présenté par

Omar DERRICHE et Chabane DEGDAG

Pour obtenir le grade de

Master en Informatique

Spécialité Conduite de Projet Informatique

Thème

**Conception et Réalisation d'une
Application Web pour le Suivi
Pédagogique des Etudiants :
Cas Faculté de Génie Electrique et d'Informatique**

Soutenu le, 04/10/2012 Devant le jury :

Président : BOUARABE Farida
Rapporteur : BERKANE Tassidit
Examineurs :

- OUGHILIS Farida
- SINI Ghenima

Résumé

Le travail qui nous a été suggéré dans le cadre de notre projet de fin d'étude, consiste en une conception et une réalisation d'une application web pour la gestion de la scolarité de la faculté de génie électrique et d'informatique (FGEI) de l'université MOULOUD MAMMERI de TZI-OUZOU (UMMTO).

Pour réaliser ce travail, nous avons utilisé le langage de modélisation UML pour la conception du système d'information, et les langages de programmation PHP et SQL pour la réalisation.

L'utilisation de cette application par les principaux acteurs (Chef de service de la scolarité, Agents de la scolarité, Chefs de départements, Enseignants, Etudiants) de la FGEI, permet d'offrir un petit pas vers la professionnalisation des études universitaires au sein de la FGEI.

Mots clés

- **Application Web**
- **Scolarité**
- **FGEI**
- **UMMTO**
- **Architecture Client/serveur**
- **Cas d'utilisation**
- **Classe et Objet**
- **CSS**
- **JavaScript**
- **LMD (Licence Master Doctorat)**
- **MYSQL**
- **PHP (*Hypertext Preprocessor*)**
- **SQL (Structured Query Language)**
- **SGBD Relationnels**
- **Serveur Web**
- **UML (Unified Modeling Language)**
- **XAMPP (X Apache MySQL Perl PHP)**

Remerciements

Nous tenons à remercier à travers ces quelques lignes toutes les personnes impliquées à la réalisation de ce modeste travail.

Nous remercions T. **BERKANE**, notre promotrice, pour avoir bien voulu nous y accueillir, et pour son aide dans la réalisation de ce projet.

Nous remercions vivement Monsieur **KADRI**, le chef de service de FGEI de l'UMMTO, et toutes les personnes qui travaillent au sein de service de la scolarité de FGEI pour l'aide qu'ils nous ont accordé.

Les membres du jury pour avoir accepté d'honorer par leur jugement notre travail.

Nous sommes également reconnaissants à nos familles (la famille **DERRICHE** & la famille **DEGDAG**), et à tous nos amis pour l'aide morale qu'ils nous ont accordé.

Enfin, merci à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Table des Matières

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

I. Introduction aux réseaux informatiques

Introduction	3
1. Réseaux informatiques	3
1.1 Définition d'un RI	3
1.2 Apports des RI	4
1.3 Terminologie	5
2. Types de réseaux informatique	5
3. Différentes topologies logiques des RI	6
4. Interconnexion des réseaux informatique	8
4.1 Modèle de référence à sept couches (OSI)	8
4.2 Architecture TCP/IP	10
4.3 Modèle UIT-T	11
5. Architecture Client/serveur	12
6. L'application client-serveur de bases de données	15
Conclusion	16

II. Présentation de l'organisme d'accueil

Introduction	17
1. Présentation de l'université de TIZI-OUZOU	17
1.1 Historique	17
1.2 Organigramme de l'université de Tizi-Ouzou	19
2. Le système LMD (Licence Master Doctorat)	20
3. Faculté de génie électrique et d'informatique (FGEI) de l'UMMTO	23
3.1 Définition d'une faculté	23
3.2 Présentation de la faculté FGEI	23

3.3 Organigramme de la FGEI	24
3.4 Missions de FGEI	25
4. Etude de l'existant	25
4.1 Activités du service de la scolarité de la FGEI	25
4.2 Situation informatique au service de la scolarité FGEI	25
4.3 Problématiques	26
4.3 Objectif de l'étude	26
4.3 <i>Description</i> du Graphe des flux d'informations	27
4.3 Règles de gestion de la scolarité de FGEI	28
Conclusion	29

III. Analyse et conception

Introduction	30
1. Présentation d'UML	30
1.1 Définition	30
1.2 Extension UML pour les applications web	31
2. La démarche d'élaboration du projet	31
2.1 Diagrammes structurels	32
2.2 Diagrammes comportementaux	32
3. Définition des besoins	34
3.1 Les besoins non fonctionnelles	34
3.2 Identification des acteurs et leurs besoins fonctionnels	34
3.3 Identification des cas d'utilisations	35
3.4 Spécification des scénarios	37
3.5 Diagrammes représentatifs	43
3.5.1 Diagramme de contexte de l'application	43
3.5.2 Diagrammes des cas d'utilisation	44
3.5.3 Diagrammes de séquence	49
4. Analyse	52
4.1 Les diagrammes de séquence d'analyses détaillés du cas d'utilisation	52
5. Conception	58
5.1 Diagramme de séquence de réalisation des cas d'utilisation	58
5.2 Les diagrammes de classes de l'application	63
5.3 Diagramme de classes général	66
6. Conception de la base de données	67
Conclusion	70

IV. Réalisation et Mise en œuvre

Introduction	71
1. Architecture logicielle de l'application	71
2. Environnement de développement	71
3. Outils de développement	72
4. Langages de programmation	72
5. Environnement logiciel	74
6. Présentation des interfaces de l'application	76
Conclusion	86
 Conclusion Générale et perspectives	 87
 Index bibliographique	 88
Annexe	89
Annexe I : Extension UML pour les applications web	89

INTRODUCTION GENERALE

Introduction Générale

Durant ces dernières années l'informatique s'est imposée d'une manière très impressionnante dans les entreprises, cela est dû à son apport extraordinaire dans le domaine de gestion des bases de données.

Elle est de plus en plus utilisée dans tous les domaines d'activités y compris celui de la gestion des scolarités aux universités auquel nous rattacherons d'ailleurs notre étude, et cela pour une meilleure gestion des différents traitements exigés par cette activité.

De ce fait, on a été sollicité par les responsables de la faculté de GEI afin de leur concevoir un système d'information automatisé.

Pour la réalisation de cette tâche, notre choix s'est porté sur le Processus Unifié (UP). En effet, le processus unifié est une solution de développement logiciel adaptée à tout type de projet. Ses traits distinctifs tiennent en trois notions : piloté par les cas d'utilisation, centré sur l'architecture, itératif et incrémental.

Le langage de modélisation qu'on a utilisé est UML (Unifier Modeling Language), qui est une partie intégrante de la démarche UP. Ses diagrammes sont largement utilisés dans chaque étape et phases de ce processus de développement.

Pour l'implémentation, le choix du langage de programmation s'est porté sur le langage de programmation PHP. La base de données est implémentée avec MySQL qui est largement compatible avec PHP.

Pour réaliser ce travail, nous avons subdivisé le plan du mémoire en quatre principaux chapitres.

- Dans le premier chapitre intitulé « Généralités sur les réseaux informatiques » nous présentons quelques concepts de base de la méthode et outils qui nous ont servi pour l'accomplissement de notre projet.
- Dans le deuxième chapitre de ce travail, nous présentons l'organisme d'accueil, ou nous approfondissons la compréhension du contexte du système par un processus continu de collecte d'information auprès des utilisateurs du domaine, déterminer les inconvénients majeurs de la gestion actuelle du service d'accueil, énumérer des suggestions informatiques qui peuvent remédier aux difficultés rencontrées, et enfin, tenant compte des moyens de la faculté, nous proposons la solution qui paraît la plus adéquate.
- Dans le troisième chapitre : « Analyse et conception », nous étendons la représentation des diagrammes effectués au niveau de l'analyse en y intégrant les aspects techniques les plus proches des préoccupations physiques.
- Finalement dans le dernier chapitre qu'on a nommé « Réalisation » nous présentons les outils de développement qui nous ont servi pour le développement de l'application, et enfin l'activité test qui consiste à tester dans le but de s'assurer de son bon fonctionnement.

Chapitre 1

Introduction aux réseaux informatiques

Introduction

Les réseaux informatiques ont pour fonction de transporter des données d'une machine terminale à une autre.

Une série d'équipements matériels et de processus logiciels sont mis en œuvre pour assurer ce transport, depuis les câbles terrestres ou les ondes radio dans lesquels circulent les données jusqu'aux protocoles et règles permettant de les traiter. Cette première partie rappelle les principes de fonctionnement des réseaux informatiques et présente en détail les matériels et architectures protocolaires sur lesquels ils se fondent.

1. Réseaux Informatiques

1.1 Définition d'un Réseau Informatique

Un réseau informatique est un ensemble **d'éléments matériels** reliés entre eux dans le but de permettre aux **utilisateurs** de **partager des ressources** et d'échanger des informations sous forme numérique.

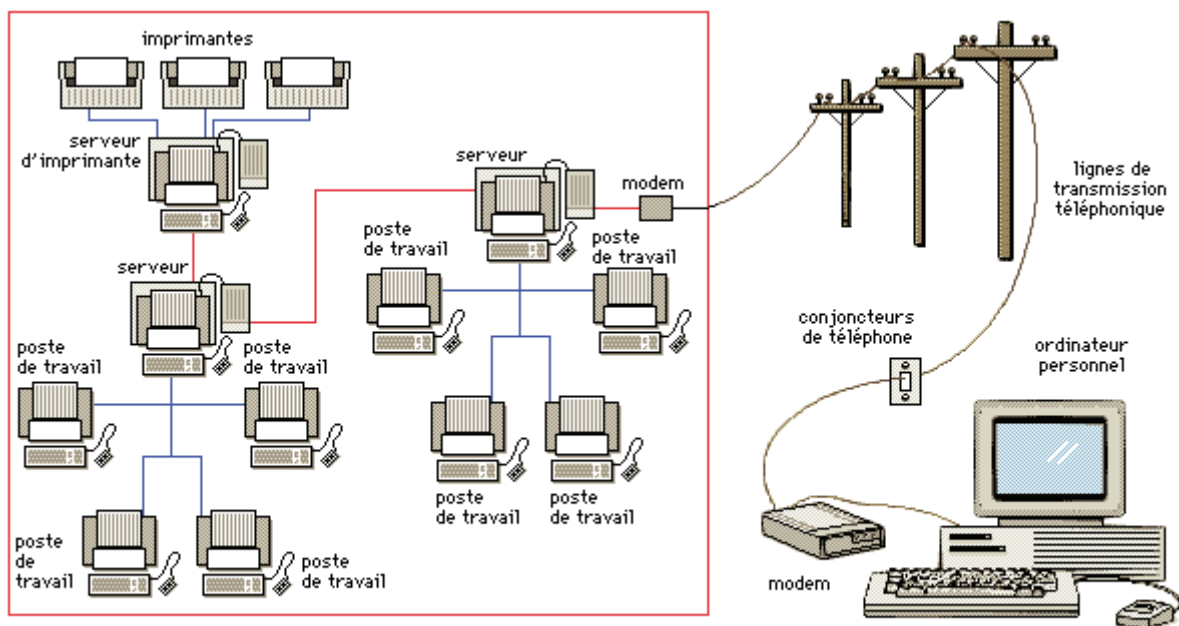


Fig. I.1 : Les réseaux informatiques [I.1]

1.2 Apports des réseaux informatiques

Les Réseaux permettent :

- De partager des fichiers.
- Le transfert de fichiers.
- Le partage d'applications : compilateur, système de gestion de base de données (SGBD).
- Partage d'imprimantes.
- L'interaction avec les utilisateurs connectés : messagerie électronique, conférence électronique, Talk,
- Le transfert de données en générale.
- Le transfert de la parole, de la vidéo et des données (réseaux à intégration de services ou multimédia)

1.3 Terminologie

- **nœud** : Equipement (ordinateur, concentrateur, imprimante etc.) du réseau informatique possédant une adresse physique.
- **nœud TCP/IP** : Equipement fonctionnant avec le protocole TCP/IP.
- **adresse physique** : Adresse unique attribuée par le fabricant, gérée par un organisme international et inscrite de façon définitive dans l'équipement. Appelée aussi adresse Ethernet ou adresse MAC (Media Access Control).
- **adresse logique** : Adresse unique attribuée par les répondants informatique locaux et gérée par le NIC pour les adresses IP (organisme international).
- **paquet (trame)** : Information électrique transitant sur le réseau et représentant une suite de bits. Il s'agit de l'unité de base du réseau.
- **protocole** : Description des mécanismes permettant la gestion de paquets d'information et leur transition du réseau à l'application. Par extension, logiciel (software) fonctionnant sur une machine et permettant cette gestion interne.
- **suite de protocoles** : Ensemble de protocoles communiquant entre eux.
- **couches réseau** : Composante protocole fonctionnant sur un équipement et lui permettant de communiquer à travers le réseau.
- **media** : Support permettant le passage de l'information : différents types de câbles, ondes hertziennes, laser etc.
- **bande-passante** : Comme en téléphonie ou en hi-fi, la bande passante détermine la quantité d'informations capables de transiter par seconde sur un media donné.
- **backbone** : Réseau physique (câble et matériel actif) servant de colonne vertébrale au reste du réseau. En général composée de fibres optiques et assurant parfois une plus grande bande-passante (155Mbits/s).
- **interconnexion** : Ensemble matériel et logiciel permettant la connexion d'équipements utilisant des média et des protocoles différents
- **segment physique** : Câble ou ensemble de câbles reliés sans équipement électronique d'interconnexion (il s'agit d'un seul domaine de collisions).
- **segment logique** : Sous réseau regroupant des nœuds correspondant à un même groupe d'adressage. Un segment logique peut englober plusieurs segments physiques.

2. Types de réseaux informatiques

Lorsque l'on parle de réseau informatique, il faut distinguer 3 types de réseaux (LAN, MAN, WAN) dont les limites ne sont pas fixées de manière absolue et qui peuvent former ensemble, un réseau d'entreprise.

Ces différents types de réseaux peuvent très bien cohabiter au sein d'un même réseau d'entreprise : le Backbone est un anneau à grande bande-passante en fibre optique, les ordinateurs individuels sont reliés à un bus, tandis que les ordinateurs du centre de calcul sont connectés en étoile.

a) Les réseaux locaux (LAN : Local Area Network)

Ces réseaux sont en général circonscrits à un bâtiment ou à un groupe de bâtiment pas trop éloignés les uns des autres (site universitaire, usine ou 'campus'). L'infrastructure est privée et est gérée localement par le personnel informatique. De tels réseaux offrent en général une bande-passante comprise entre 4Mbit/s et 100 Mbits/s.

b) Les réseaux métropolitains (MAN : Metropolitan Area Network)

Ce type de réseau est apparu relativement récemment et peut regrouper un petit nombre de réseaux locaux au niveau d'une ville ou d'une région. L'infrastructure peut être privée ou publique. Par exemple, une ville peut décider de créer un 'MAN' pour relier ses différents services disséminés sur un rayon de quelques kilomètres et en profiter pour louer cette infrastructure à d'autres utilisateurs. La bande-passante peut être de quelques centaines de Kbits/s à quelques Mbits/s.

c) Les réseaux distants (WAN : Wide Area Network)

Ce type de réseau permet l'interconnexion de réseaux locaux et métropolitains à l'échelle de la planète, d'un pays, d'une région ou d'une ville.

L'infrastructure est en général publique (PTT, Télécom ...etc.) et l'utilisation est facturée en fonction du trafic et/ou en fonction de la bande-passante réservée, pour les lignes louées (une ligne louée est réservée exclusivement au locataire, 24h sur 24, pour la durée du contrat).

Les modems sont un des éléments de base des WANs.

La bande-passante va de quelques Kbits/s à quelques Mbit/s. Une valeur typique pour une ligne louée est de 64kbits/s (en fonction des services offerts).

3. Différentes Topologies de Réseaux Informatiques

Il convient de distinguer la topologie logique de la topologie physique :

La topologie logique décrit le mode de fonctionnement du réseau, la répartition des nœuds et le type de relation qu'ont les équipements entre eux.

La topologie physique décrit la mise en pratique du réseau logique (câblage etc.).

La topologie des réseaux en général et des réseaux informatiques en particulier peut se partager en trois groupes, qui peuvent de combiner :

a) Les réseaux en étoile (Star)

Chaque nœud est relié directement à un nœud central : l'information passe d'un nœud périphérique au nœud central, celui-ci devant gérer chaque liaison. (voir figure I.2)



Fig. I.2 Réseaux en étoile [I.1]

b) Les réseaux en anneau (RING)

Chaque nœud est relié au nœud suivant et au nœud précédent et forme ainsi une boucle : l'information transite par chacun d'eux et retourne à l'expéditeur. (voir figure I.3)



Fig. I.3 Réseaux en anneau [I.1]

c) Les réseaux en bus

Chaque nœud est connecté sur un bus : l'information passe devant chaque nœud et s'en va mourir à l'extrémité du bus.

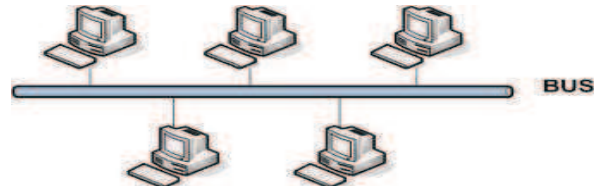


Fig. I.4 Réseaux en bus

D) Structure Hybride

La structure hybride de réseau emploie un mélange de différentes structures de réseau, comme l'anneau, le bus et également l'étoile.

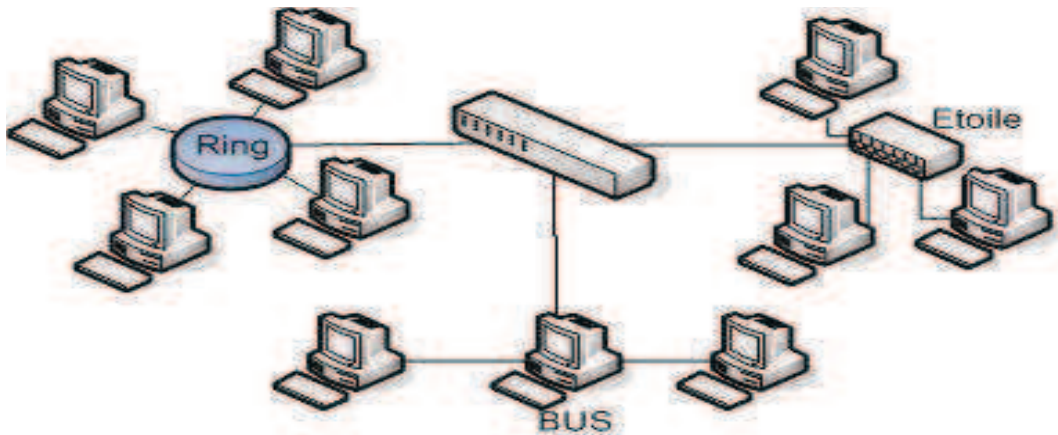


Fig. I.5 Topologie hybride

Une topologie logique en bus peut très bien correspondre à une topologie physique en étoile, suivant comment les câbles ont été posés, mais ce qui importe au niveau de la compréhension des mécanismes du réseau informatique est bel et bien la topologie logique.

4. Interconnexion des réseaux Informatiques

La nécessité d'interconnecter plusieurs réseaux locaux apparaît dans les entreprises qui disposent de plusieurs établissements ou dans celles qui désirent communiquer entre elles et s'échanger des informations. Assurer cette interconnexion consiste à établir un dialogue entre des machines d'origines diverses ; celles-ci peuvent être en outre reliées par des modèles différents de réseaux. L'objectif est d'assurer aux utilisateurs une parfaite transparence de l'interconnexion, quels que soient les divers protocoles mis en œuvre par les équipements de communication.

4.1. Modèle de référence à sept couches (OSI)

Le modèle dit d'interconnexion des systèmes ouverts, appelé encore **OSI** (**O**pen **S**ystems **I**nterconnection) a été défini par l'ISO (International Standards Organization). Ce modèle répartit les protocoles utilisés selon sept couches, définissant ainsi un langage commun pour le monde des télécommunications et de l'informatique. Il constitue aujourd'hui le socle de référence pour tous les systèmes de traitement de l'information.

Chaque couche du modèle regroupe des dispositifs matériels (dans les couches basses) ou logiciels (dans les couches hautes). Entre couches consécutives sont définies des interfaces sous forme de primitives de service et d'unités de données rassemblant les informations à transmettre et les informations de contrôle rajoutées.

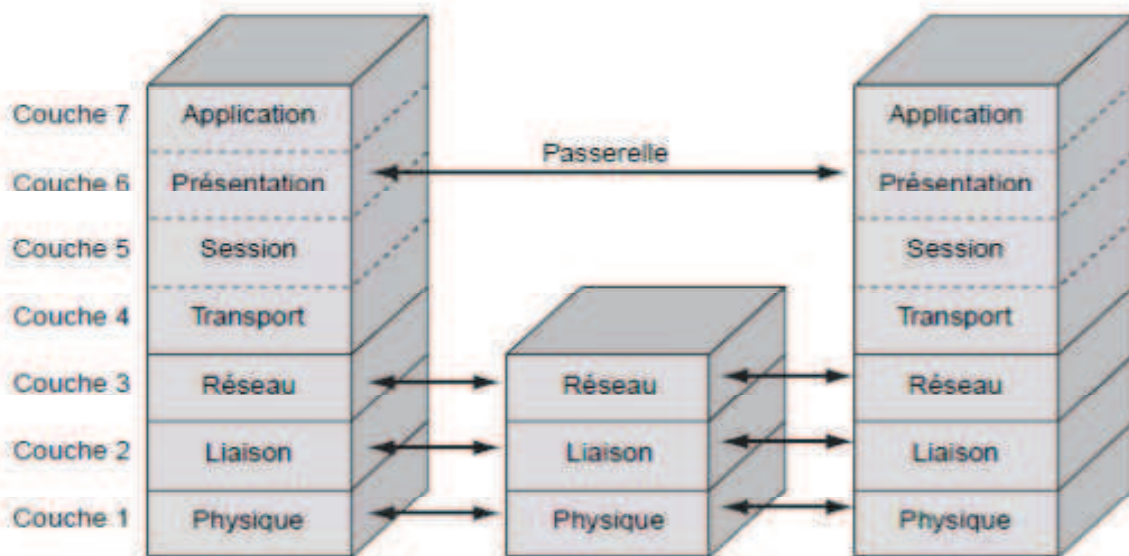


Fig. I.6 Architecture OSI

Couche 1 : physique

La couche physique rassemble les moyens électriques, mécaniques, optiques ou hertziens par lesquels les informations sont transmises. Les unités de données sont donc des bits 0 ou 1.

Couche 2 : liaison

La couche liaison gère le transfert de bits d'un nœud à l'autre du réseau, comprenant entre autres les dispositifs de détection et de correction d'erreurs, ainsi que les systèmes de partage des supports. L'unité de données à ce niveau est appelée *trame*.

Couche 3 : réseau

La couche réseau aiguille les données à travers un réseau à commutation. L'unité de données s'appelle [en général] un *paquet*.

Couche 4 : transport

La couche transport regroupe les règles de fonctionnement de bout en bout, assurant ainsi la transparence du réseau vis-à-vis des couches supérieures. Elle traite notamment l'adressage, l'établissement des connexions et la fiabilité du transport.

Couche 5 : session

La couche session réunit les procédures de dialogue entre les applications : établissement et interruption de la communication, cohérence et synchronisation des opérations.

Couche 6 : présentation

La couche présentation traite les formes de représentation des données, permettant la traduction entre machines différentes.

Couche 7 : application

Source et destination de toutes les informations à transporter, la couche application rassemble toutes les applications qui ont besoin de communiquer par le réseau : messagerie électronique, transfert de fichiers, gestionnaire de bases de données, etc.

4.2. Architecture TCP/IP

En raison des difficultés technique et des problèmes de normalisation, s'est développée à la fin années 1970, une famille de protocoles appelée **TCP/IP** (Transmission Control Protocol / Internet Protocol).

Ces protocoles constituent un modèle d'interconnexion quelque peu différent mais surtout plus simple que le modèle OSI à sept couches. Comme ces protocoles sont faciles à mettre en place et permettent d'interconnecter tous les types de réseaux, ils se sont rapidement développés, étant en outre d'un usage gratuit, puisque n'appartenant pas à une compagnie particulière. C'est ainsi qu'est apparu le réseau Internet, qui désigne un ensemble de machines reliées entre elles à travers le monde selon ces protocoles, chacune pouvant communiquer avec toutes les autres.

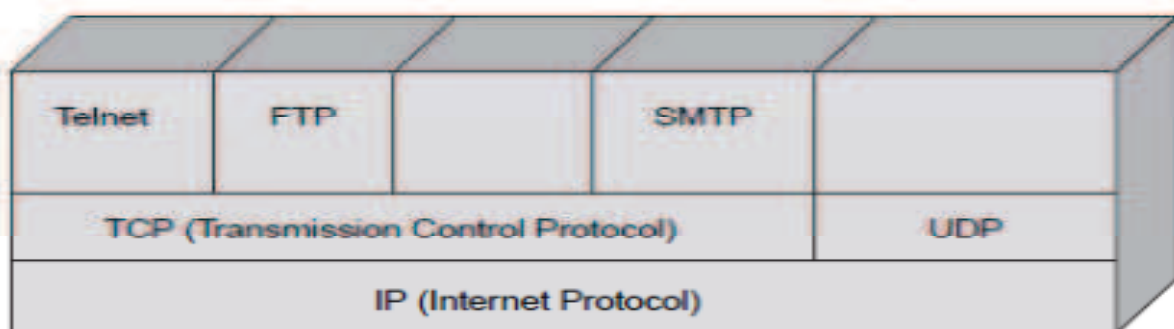


Fig. I.7 Architecture TCP/IP

- **IP** (Internet Protocol), de niveau réseau, qui assure un service sans connexion.
- **TCP** (Transmission Control Protocol), de niveau transport, qui fournit un service fiable avec connexion.
- **Telnet**, Protocole de gestion de terminal virtuel, permet d'obtenir les logiciels d'un autre ordinateur grâce au réseau.
- **FTP** (File Transfère Protocol), pour le transport de fichiers.
- **SMTP** (Simple Mail Transfère Protocol), pour le transfert du courrier électronique.
- **UDP** (User Datagram Protocol), est une couche haute située en dessus de IP et en parallèle avec TCP. Le mécanisme d'UDP est beaucoup plus rustique que TCP puisqu'il ne met pas en place de solution de renvoi de paquet(s) en cas d'erreur ou de perte. Cette couche est peu utilisée, si ce n'est pour des applications ne nécessitant pas une garantie quant à la réussite de chaque transaction (TFTP par exemple (Trivial File Transfert Protocol)). UDP/IP et TCP/IP peuvent cohabiter sur le même équipement.

4.3. Modèle UIT-T

L'architecture **UIT-T** a été définie pour le **RNIS** large bande en 1988, mais elle reste valide pour les réseaux de la future génération **NGN** (Next Generation Network).

Les réseaux de télécommunication de nouvelle génération utilisent une technique de commutation particulière, appelée commutation de cellules. La cellule est une petite trame ATM de longueur fixe facile à manipuler. Pour répondre à cette commutation et faire en sorte que les nouveaux réseaux prennent facilement en compte les applications multimédias, l'UIT-T a développé un nouveau modèle de référence qui s'appuie sur un mode avec connexion.

L'architecture du modèle UIT-T peut être considérée comme compatible avec le modèle de référence TCP/IP, bien que les fonctionnalités ne soient pas regroupées aux mêmes niveaux.

- La couche physique, par exemple, est dotée de fonctionnalités élargies par rapport à la couche équivalente du modèle de référence, qui améliorent sa rapidité de fonctionnement. Cette couche physique est de surcroît capable de reconnaître le début et la fin de la cellule ATM, ce qui permet de dire que la cellule est une trame et que le réseau est un réseau de niveau 2.
- La couche AAL recoupe en partie le niveau message du modèle de référence.

Le découpage en couches de l'architecture UIT-T est illustré à la figure I.5.

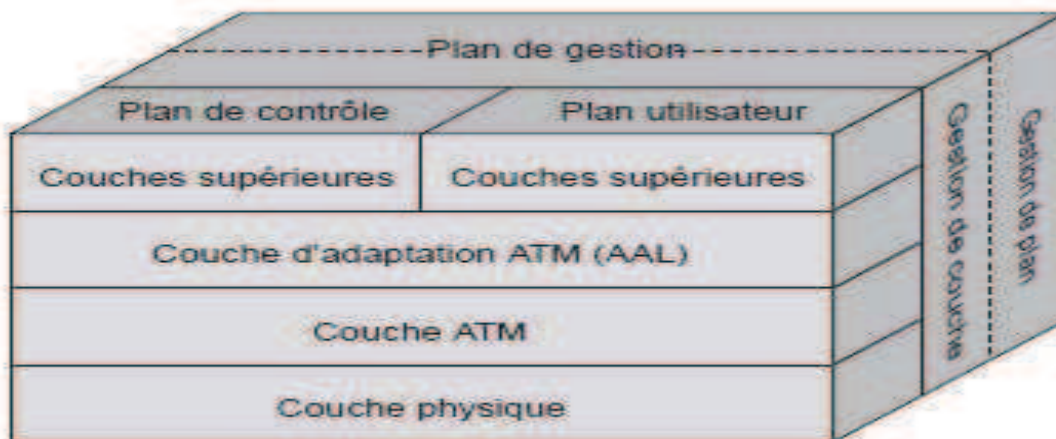


Fig. 1.5 Couches de l'architecture UIT-T [1.2]

5. Architecture Client/serveur

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

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client FTP, client de messagerie, ..., lorsque l'on désigne un programme, tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès du serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client messagerie il s'agit de courrier électronique). [I.3]

a. Principe de l'architecture Client/serveur

Un système client/serveur fonctionne selon le schéma suivant :

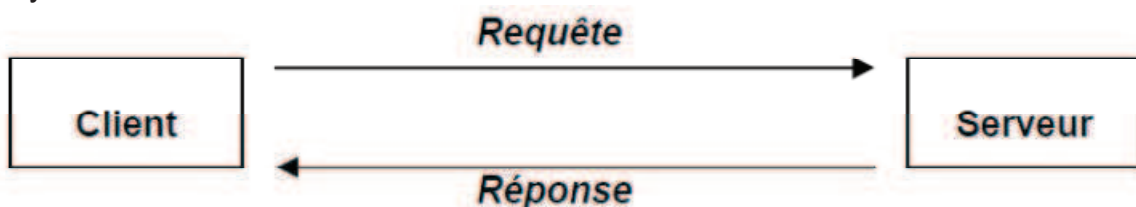
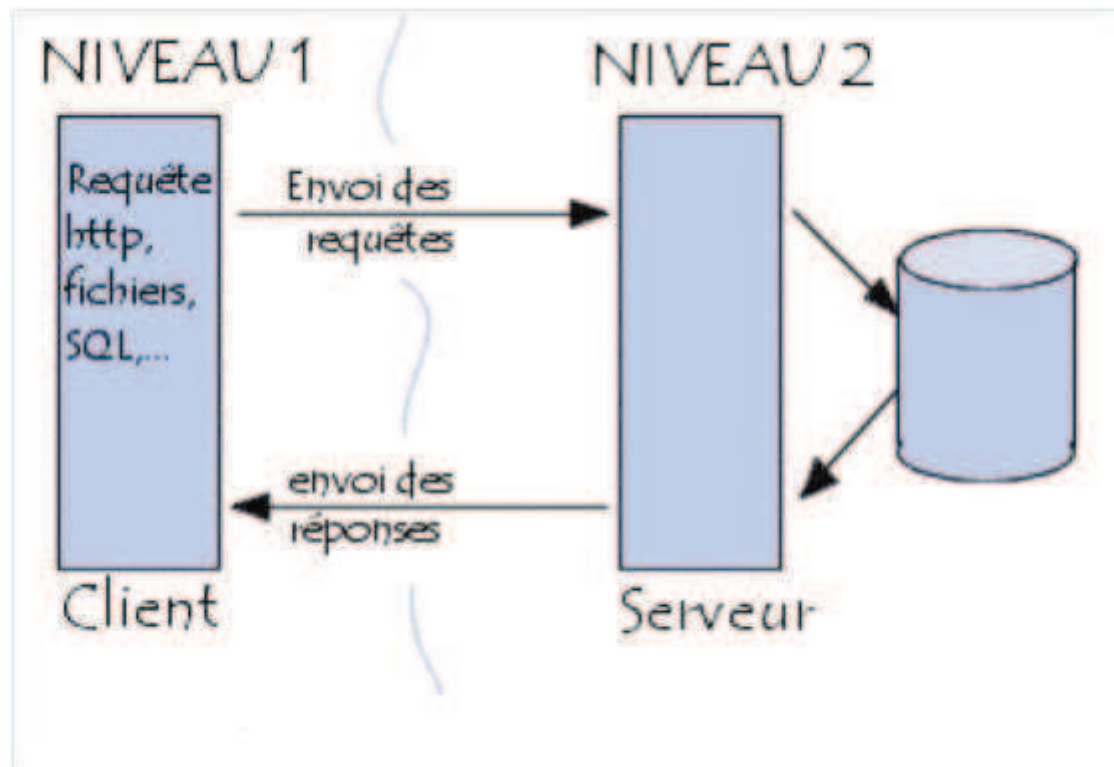


Fig I.6 : fonctionnement d'un système client/serveur

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

b. Architecture à deux niveaux [I.4]



Le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service.

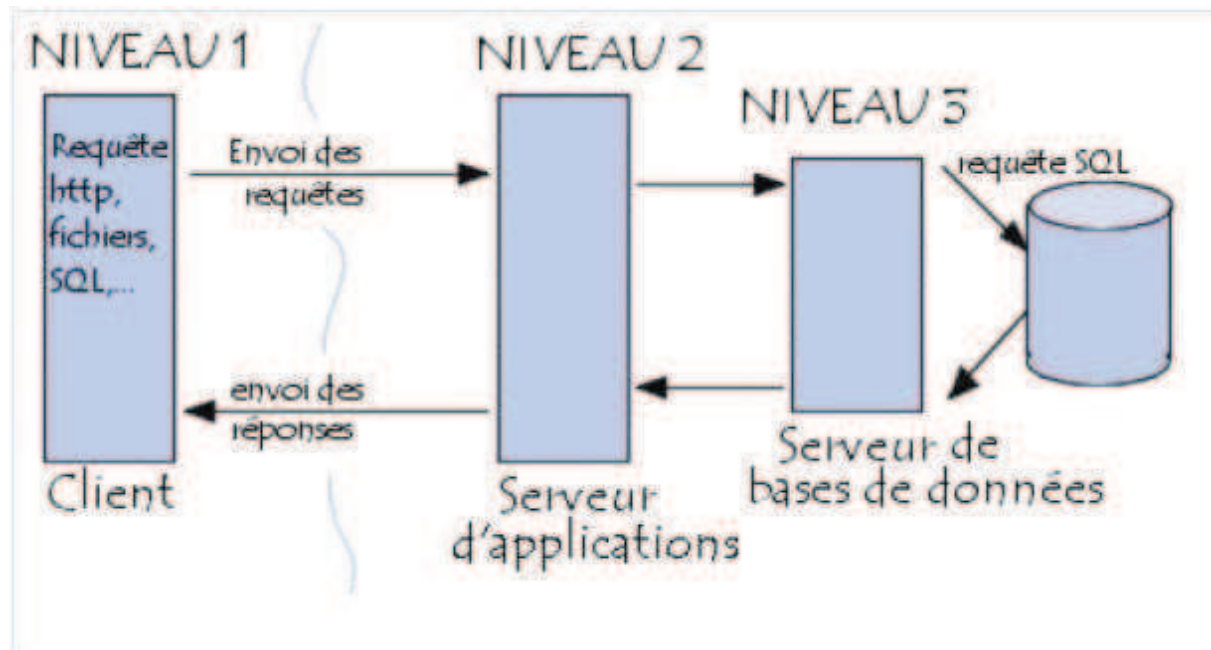
Exemples simples :

- La consultation de pages sur un site web fonctionne sur une architecture client/serveur. Le client est l'internaute connecté au réseau via son ordinateur et un navigateur web, le serveur est l'ordinateur serveur WEB contenant les pages Web demandées. La communication utilise le protocole http.
- L'envoi d'une requête SQL depuis un poste client sur un serveur de bases de données SQL Server
- La réception et l'émission de courrier : le client est l'internaute qui réceptionne et expédie ses courriers avec son logiciel de messagerie (ex Outlook) le serveur est le ou les serveurs de messagerie. Cette communication utilise les protocoles POP (serveur courrier entrant) et SMTP (serveur courrier sortant).

Parmi les risques et inconvénients liés à cette architecture on peut citer les plus évidents :

- si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge
- si le serveur n'est plus disponible, plus aucun des clients ne marche

c. Architecture à trois niveaux [I.4]



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

Exemple : sur un site marchand le client émet une requête demandant la liste de certains produits :

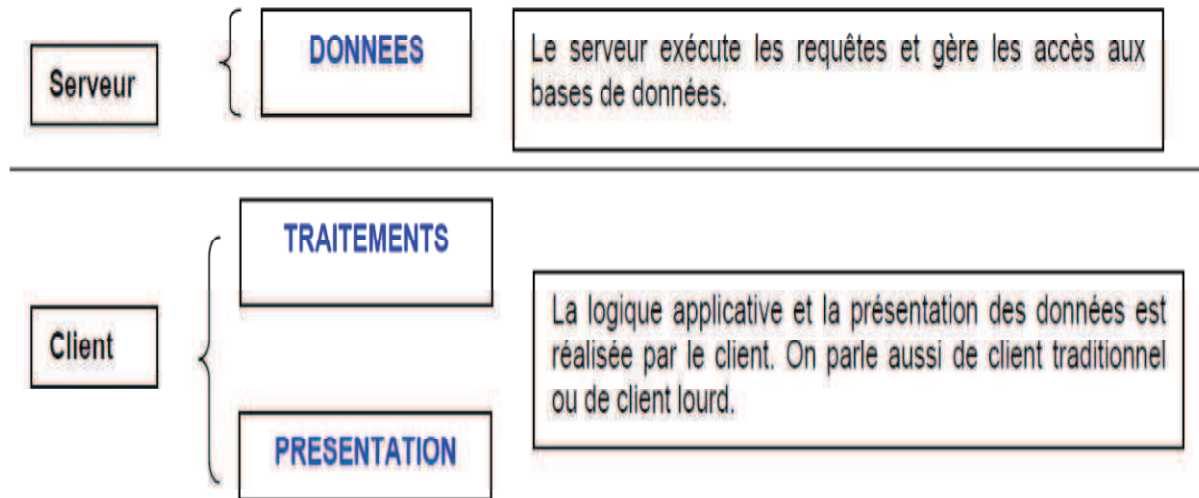
Le serveur Web transmettra la demande à un serveur de bases de données puis mettra en forme dans une page web la table résultat renvoyée par le serveur de bases de données.

6. L'application client-serveur de bases de données [I.3]

Toute application peut être découpée en trois parties :

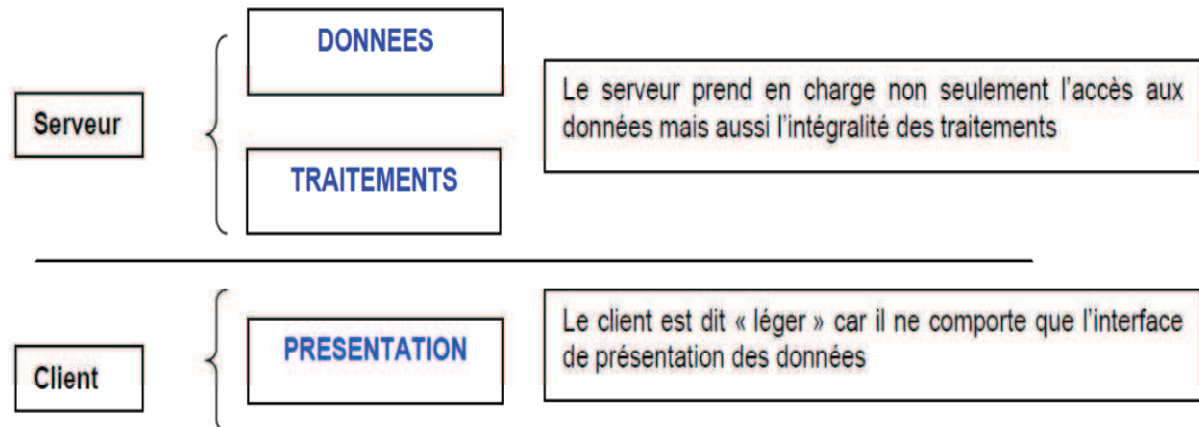
- Les **données**
 - Les **traitements** : actions sur les données en vue de produire un résultat utilisable en l'état par le destinataire (programmation traditionnelle ou par un langage de requêtes SQL)
 - La **présentation** ou **interface utilisateur** : concerne la présentation des données
- La répartition de ces composants entre le client et le serveur se fera en fonction de la nature des traitements, de la puissance des machines et de l'étendue du réseau.

a. Application client-serveur à deux niveaux (architecture 2 tiers)



Avantages	Inconvénients
<ul style="list-style-type: none"> • solution fréquemment adoptée, de nombreux logiciels professionnels fonctionnent selon ce mode • le serveur de bases de données assure les contrôles d'intégrité et de validité des Données 	<ul style="list-style-type: none"> • Inadaptés dans le cas d'un grand nombre d'utilisateurs connectés • La mise à jour des applications est lourde (les applications sont en effet à mettre à jour au niveau de chaque client)

b. Application Client-serveur de bases de données à 3 niveaux (architecture 3 tiers)



Avantages	Inconvénients
<ul style="list-style-type: none"> • Interface conviviale • Mise à jour facile des applications (celles ci sont centralisées sur un seul et même serveur) 	<ul style="list-style-type: none"> • Risque de forte sollicitation du serveur (échanges nombreux) • -Le serveur doit être surdimensionné (en mémoire vive pour l'exécution des programmes et en disque pour les données)

c. Localisation des données : stratégies de segmentation et/ou de réplication

En cas d'augmentation de volume important de la base de données, il convient d'opérer une répartition des données sur plusieurs serveurs et/ou entre le serveur et les clients.

- Tout ou partie d'une base de données peut être stockée à l'identique sur plusieurs sites : on parle de **réplication**.
- Une base de données peut être découpée en plusieurs parties : on parle alors de **segmentation** (ou partition)

Conclusion

Toutes les applications finies fondées sur les bases de données sont entièrement logées dans au moins une architecture réseau, c'est pour cela il est indispensable de familiariser avec les réseaux informatiques et l'architecture client/serveur.

Chapitre II

Présentation de l'organisme d'accueil

Introduction

La gestion des scolarités au sein d'une université est une opération rigoureuse, qui mérite d'être perfectionnée et analysée soigneusement.

Mais avant d'essayer de porter une solution informatique pour ce processus, la présentation de l'organisme d'accueil en général et le service qui gère les mouvements des étudiants au niveau de la Faculté de génie électrique et d'informatique en particulier est nécessaire, et c'est ce qui est conseillé d'ailleurs dans toute démarche informatique de Génie Logiciel. Toutefois, il est important de signaler à l'attention des utilisateurs de mieux réaliser les prochaines étapes de notre plan de travail, la spécification et la précision de notre sujet et champ d'étude doivent être bien comprises, cernées et clarifiées.

1. Présentation de l'université de TIZI-OUZOU [II.1]**1.1. Historique**

L'université de Tizi-Ouzou a ouvert ses portes en 1977, elle avait alors le statut de centre universitaire et était à **OUED-AISSI** à 6 km de sa résidence actuelle (L'axe de la nouvelle ville). Les modes de fonctionnement de l'université de 1977 à ce jour sont les suivant :

- **De 1977 à 1984 :**

Elle avait le statut de centre universitaire dirigé par un directeur au plan administratif assisté par trois directeurs adjoints chargés respectivement de la post-graduation, de la planification ainsi que d'un secrétaire général.

A leur tour, les directeurs adjoints étaient secondés par des chefs de départements administratifs qui étaient à leur tour secondés par des chefs de division.

- **En 1985 :**

Le centre universitaire fut dissout et les instituts pédagogiques qui le constituaient ont été érigés en instituts nationaux d'enseignement supérieur (INES).

Les INES étaient dirigés par des directeurs assistés chacun de trois directeurs adjoints chargés respectivement de la pédagogie, de la recherche et de l'administration générale.

- **En 1990 :**

Les INES furent à leur tour dissout, et érigés en instituts dépendants de l'université de Tizi-Ouzou.

L'université de Tizi-Ouzou, de 1990 à 1997 fonctionnait avec instituts et départements pédagogiques.

- **En 1999 :**

Les instituts ont été dissouts également et érigés en facultés. Selon le décret exécutif N° 98/32 du 02/12/1998, l'article stipule la création au sein de l'université de Tizi-Ouzou des facultés suivantes :

Facultés	Départements correspondants
Médecine	Sciences médicales
Sciences	Sciences exactes
Droit	Sciences juridiques et administratives
Lettres et sciences humaines	Langues étrangères
	Langues et littérature arabe
	Langue et culture amazighe
	Psychologie et science de l'éducation
Génie de la construction	Architecture
	Génie civil
	Génie mécanique
Génie électronique et informatique	Electronique
	Electrotechnique
	Automatique
	Informatique
Sciences biologiques et sciences agronomiques	Sciences de la science nature agronomique
Sciences économies et sciences de gestion	Sciences économiques et sciences de gestion

Tab. II.1 : Les facultés de l'UMMTO.

En 2005 :

Cette année a été marquée par l'ouverture de nouvelles filières et troncs communs dans différentes facultés de l'UMMTO. Il s'agit de :

- **La faculté des lettres** : la filière Traduction et interprétariat.
- **La faculté des sciences économiques et des sciences de gestion** : le tronc commun des sciences commerciales.
- **La faculté de droit** : le tronc commun des sciences politiques et relations internationales.
- **La faculté des sciences** : Avec la réforme qu'introduit le système **LMD (Licence Master Doctorat)**, la faculté des sciences englobe trois domaines : Sciences et techniques (code **D01**), Science de la matière (code **D02**), et Mathématique et Informatique (code **D03**).

1.2. Organigramme de l'université de Tizi-Ouzou

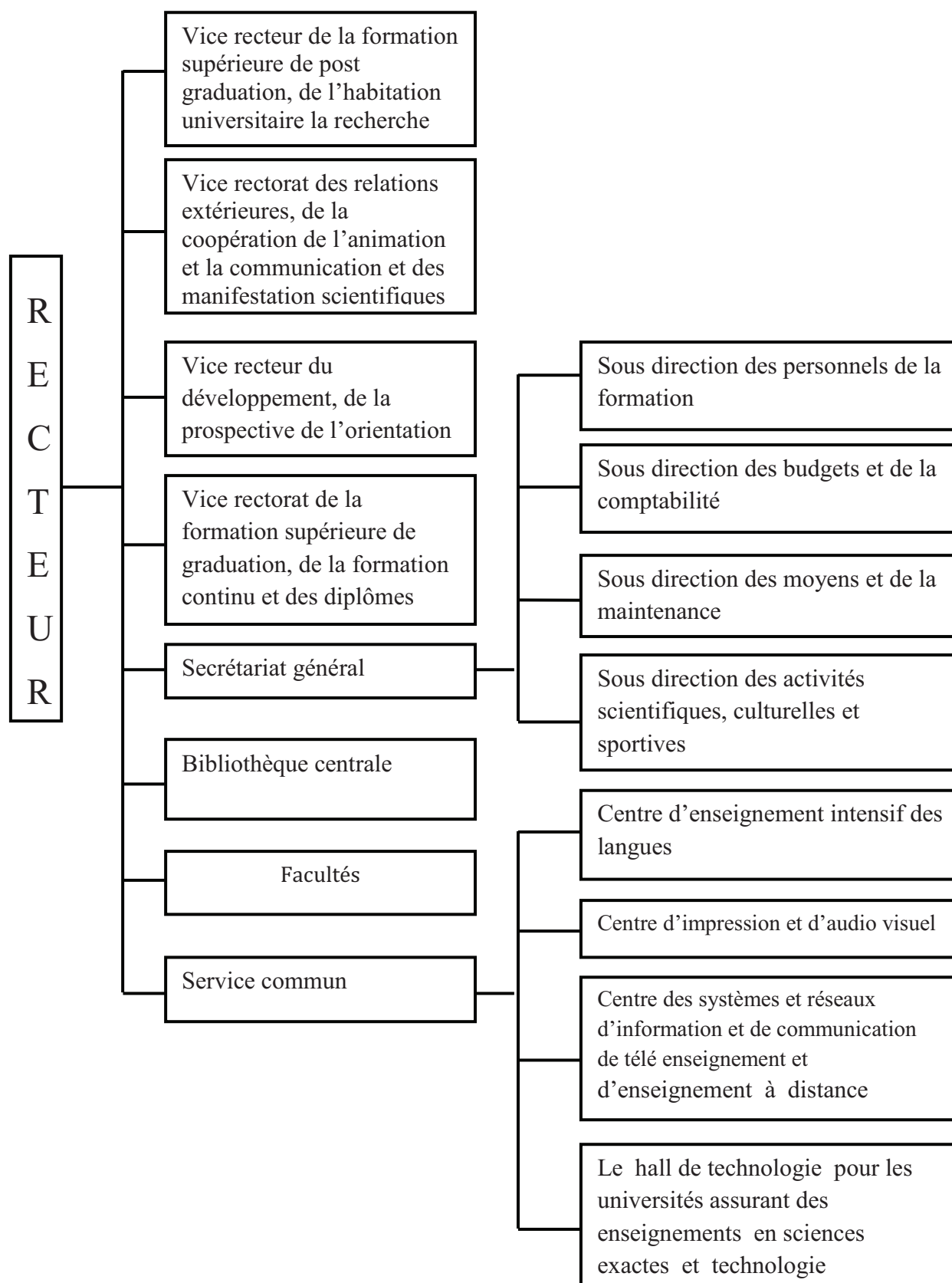


Fig. II.1 : Organigramme de l'université de Tizi-Ouzou

2. Le système LMD (Licence Master Doctorat) (I.2)

Ce système est d'origine anglo-saxonne, engendré sur des normes mondiales. Le système LMD (appelé identiquement BMD ou 3, 5, 8) a cours depuis longtemps dans les universités nord-américaines (USA, Canada) et britanniques. Celui-ci a été adopté progressivement dans la majorité des pays d'Europe, du Maghreb tels que la Tunisie, le Maroc et autres régions du monde arabe. L'objectif de ce système est de préparer à l'ouverture sur l'évolution mondiale des différentes sciences et à une meilleure insertion professionnelle. Ce système a été mis en place en Algérie en application du décret exécutif n°04- 371 du 21 novembre 2004.

Les travaux ont commencé depuis l'année 2003, ce n'est qu'en 2004 que le département de tutelle a mis en place les structures susceptibles de redonner une efficacité à ce projet. Il est convenable de mentionner que ce projet a été exposé à des responsables pédagogiques lors de nombreuses rencontres telle que celle des chefs d'établissement qui a eu lieu les 7 et 8 janvier 2004 à l'université de Badji-Mokhtar de Annaba.

Ce système se base sur un nouveau schéma de formation qui s'articule autour de trois paliers (fig. II.2) :

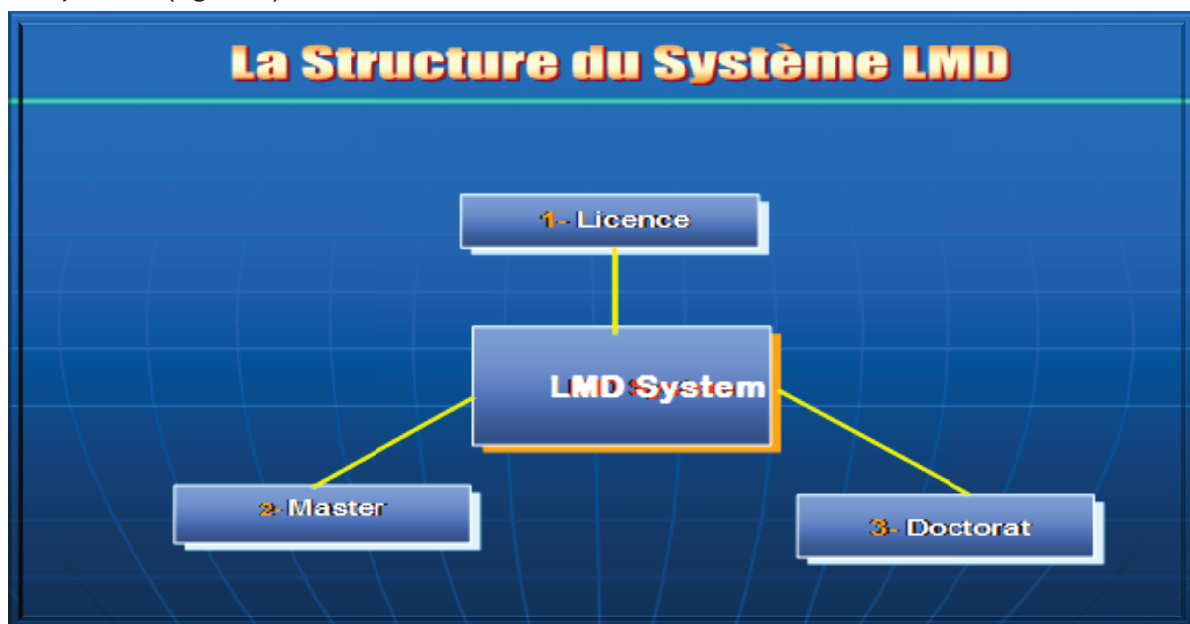


Fig. II.2 : La structure du système LMD

2.1. Définition du système LMD

Le LMD se compose de trois phases :

- **Licence** avec 06 semestres (trois ans d'étude, l'équivalent de Bachelier)
- **Master** de deux ans d'étude (04 semestres)
- **Doctorat** étude de trois ans de recherche scientifique (06 semestres)

Dans chaque semestre, les étudiants sont sensés à assister à 400 heures dans une période de 16 semaines (i.e. 25 heures par semaine).

Pour mieux définir les composantes de ce système, une explication peut être utile. Les trois phases du système LMD sont composées d'**Unités d'Enseignement (UE)** rassemblées en **semestres** d'études. Chaque unité peut être utilisable et transférable, d'une manière que son acquisition est définitive et pourra être utilisée dans un autre cursus de formation. Toutefois, cette **capitalisation** et la possibilité de transfert ouvrent des **passerelles** entre plusieurs branches et infèrent une mobilité des étudiants en offrant un grand choix qui leur permettra d'accompagner un parcours universitaire individualisé autrement dit : **autonome**

La nouvelle terminologie qu'apporte le système LMD se résume ainsi (fig. II.3) :

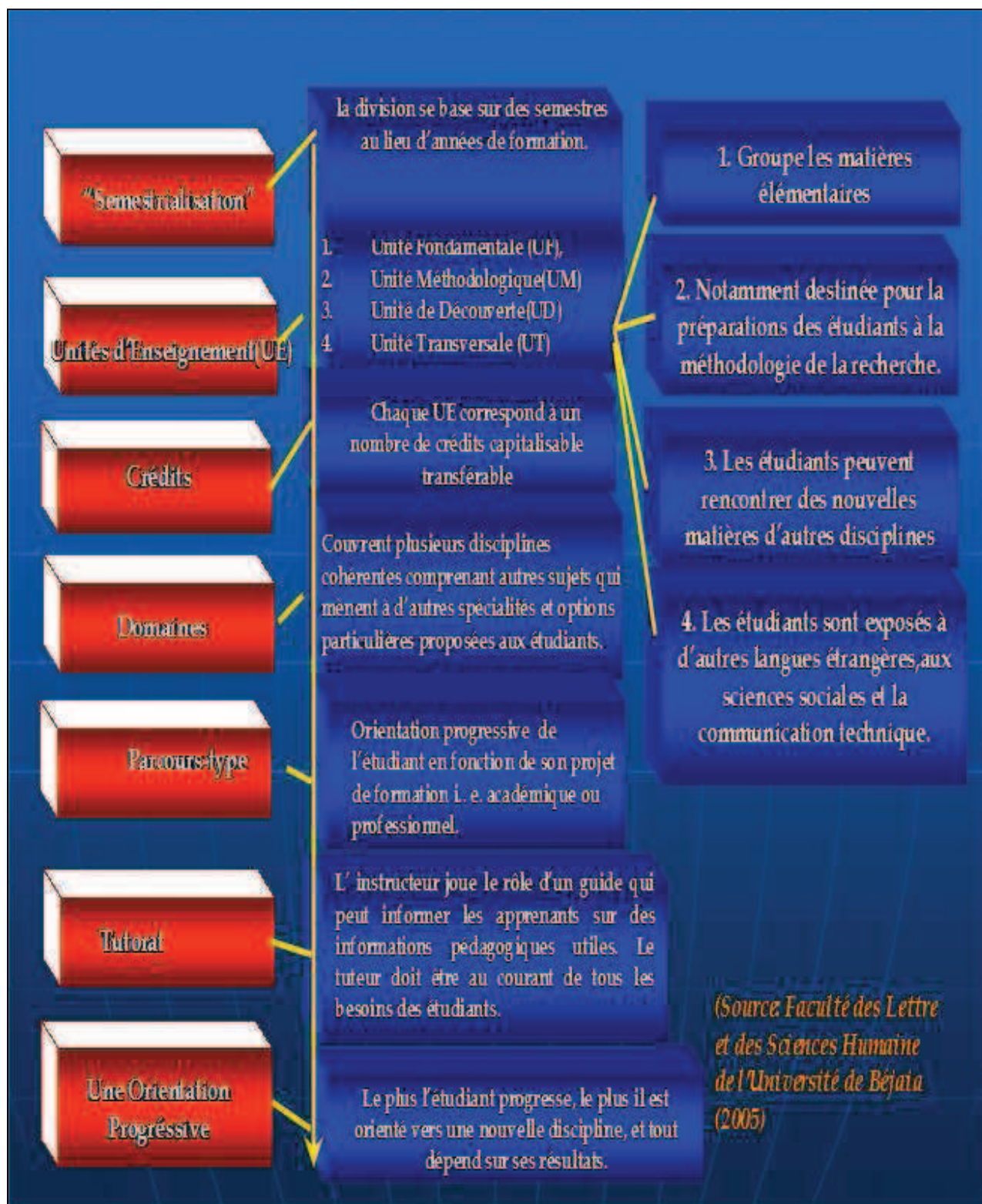


Fig. II.3 : La terminologie du système LMD

3. Présentation de la Faculté de génie électrique et d'informatique (FGEI)

3.1. Organigramme de la FGEI [II.3]

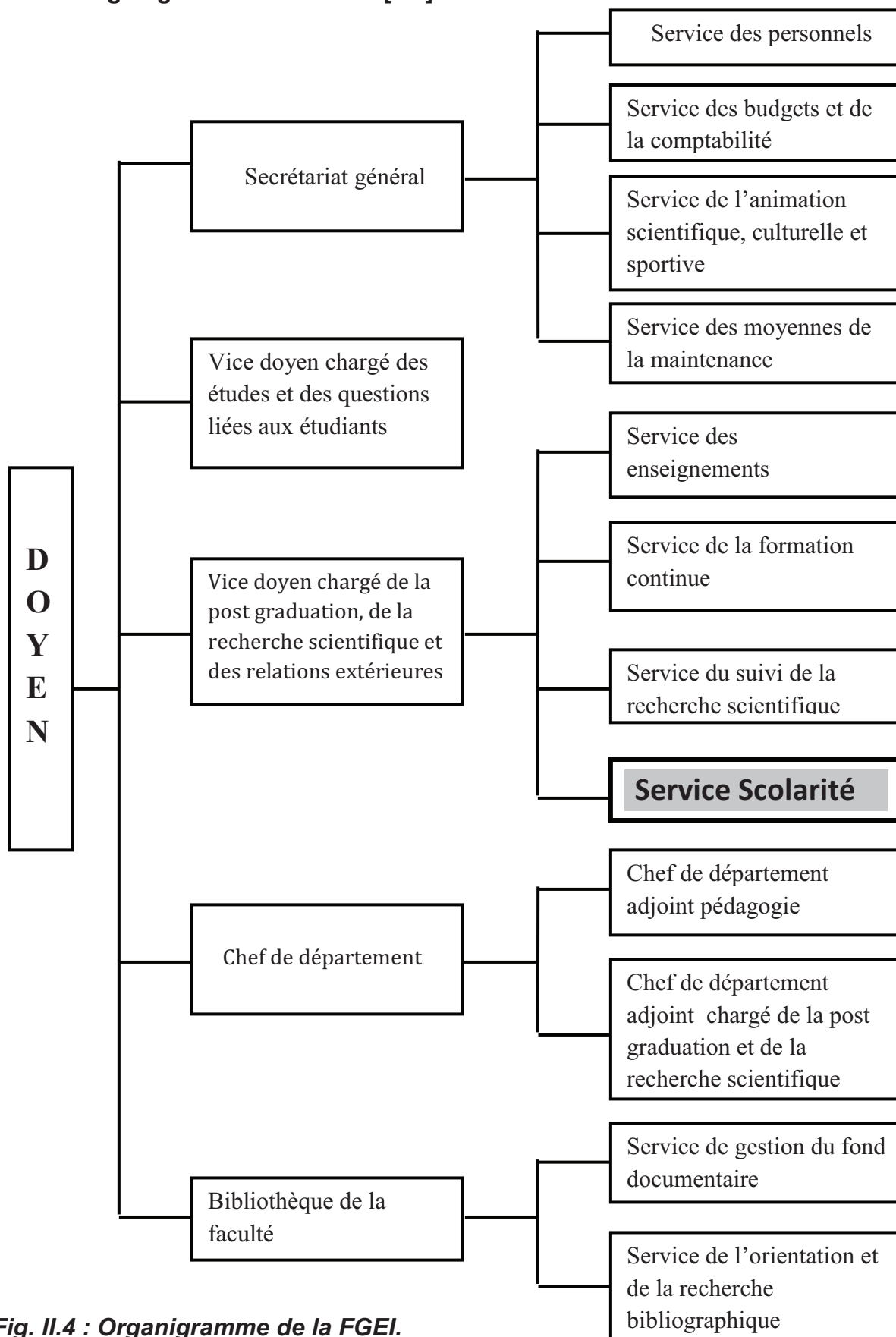


Fig. II.4 : Organigramme de la FGEI.

Une faculté est une unité autonome d'enseignement et de la recherche scientifique dans les universités.

Elle est pluridisciplinaire mais peut être le cas échéant créé autour d'une discipline dominante.

La faculté de génie électrique et d'informatique (**FGEI**) est composée de quatre départements :

- ✓ **Département d'automatique,**
- ✓ **Département d'électronique,**
- ✓ **Département d'informatique,**
- ✓ **Département d'électrotechnique (ETH).**

La FGEI assure la coordination des activités et comporte une bibliothèque organisée en services et sections.

Le département a été créé par le ministère chargé de l'enseignement supérieur, de la programmation, la réalisation, l'évaluation et le contrôle des activités d'enseignement et de la recherche dans le domaine scientifique.

3.2. Missions de la FGEI

Les missions de la faculté consistent à :

- Gérer les enseignements de graduation et post-graduation.
- Gérer les activités de recherche scientifique.
- Coordonner les travaux des différents départements.
- Aider les différents départements dans l'accomplissement de leurs objectifs.
- Administrer les différentes structures correspondantes.

4. Etude de l'existant

4.1. Activités du service de la scolarité de la FGEI

Les activités de la scolarité se résument en :

- ✓ Effectuer les inscriptions des nouveaux étudiants reçus et la réinscription des anciens étudiants qui ont suivi au moins une année d'étude.
- ✓ Le suivi de l'évolution pédagogique de l'étudiant.
- ✓ Etablissement des documents (cartes étudiant, des certificats de scolarités, relevé de note, attestation de cursus, attestation de langue, PV semestriels et globaux, attestations de succès, diplômes définitifs...)
- ✓ Contrôle des dossiers pédagogiques.
- ✓ Contrôle des dossiers de transfert.
- ✓ Contrôle des dossiers d'abandon.

4.2. Situation informatique

Le service de la scolarité de Génie Electrique et Informatique de l'UMMTO dispose d'un matériel informatique assez important qui est constitué de :

➤ **10 Ordinateurs du bureau :**

1 ordinateur est présent dans le bureau de chef de service de la scolarité de la FGEI dont 9 sont réservés pour les agents de la scolarité. Ces ordinateurs ont les caractéristiques suivantes :

- Processeur Intel (R) pentium Dual, vitesse d'horloge : 2.0 GHZ.
- Capacité mémoire 512 Mo à 1Go de RAM.
- Capacité de disque dur est 160 Go.
- Un système d'exploitation Windows XP.

➤ **5 Imprimantes** de marque HP et canon.

➤ **Logiciels**

Le service de la scolarité de la FGEI utilise le logiciel de gestion de la scolarité **Oscar 3**. Les bases de données sont dispersées dans les ordinateurs sous ce logiciel ; ce qui entraîne des pertes continues des données car éparpillées dans les ordinateurs. Ce logiciel (OSCAR3) présente un grand inconvénient, celui de ne pas permettre de rassembler toutes les données dans une base de données, et permet pas de faire toutes les activités, il est facile à exploiter et à interroger à tout moment. La croissance de l'Institution exige d'adopter une nouvelle méthode de gestion en choisissant un logiciel de gestion plus efficace car OSCAR3 n'est pas approprié pour assurer une gestion automatique d'un grand volume d'informations.

➤ **Réseau**

La scolarité de la FGEI ne dispose pas actuellement d'un réseau intranet.

4.3. Problématiques

Au cours de notre passage par les différents services de la faculté, nous avons constaté que :

- Le volume d'informations est assez important vu le nombre d'étudiant gérés.
- La majorité des traitements sont manuels à cause des limites de logiciel OSCAR3.
- Dégradation de l'état des documents due à manipulation excessive ou fréquente des traitements manuels.
- Difficulté dans la recherche de l'information.
- Mauvaise répartition des tâches entre les postes de travail.
- Importance des charges des travaux et l'exécution quotidienne des mêmes traitements.
- Problème de transport manuel des documents.

4.4. Objectifs de l'étude

Suit à tous ces problèmes et aux souhaits des gestionnaires d'automatiser toutes les activités de la faculté, nous avons pu tracer les objectifs suivants :

- ✓ Améliorer le fonctionnement et l'organisation du système actuel.

- ✓ Accélérer l'exécution des différents traitements réalisés.
- ✓ Améliorer l'image de marque de la faculté.
- ✓ Offrir un système automatique de délibération administrative et d'édition des relevés de notes.

4.5 Description de Graphe des flux d'informations [II.4]

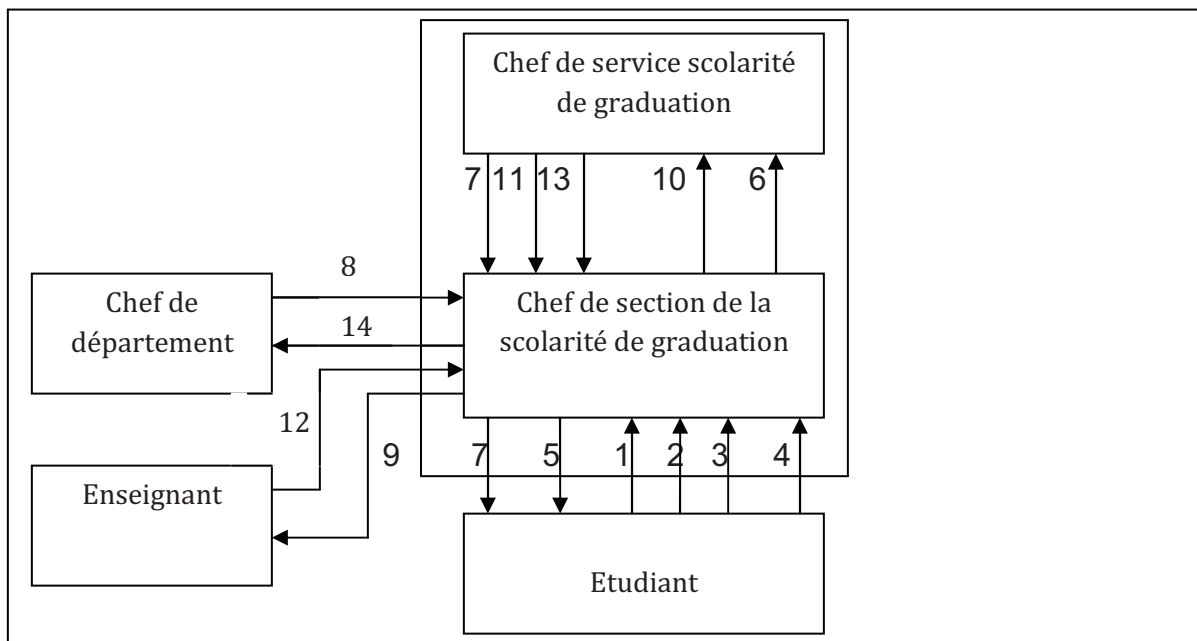


Fig. II.5 : Graphe des flux d'informations

N° du flux	Désignation du flux
1	Demande des pièces scolaires (relevé des notes, carte d'étudiant ...).
2	demande de diplôme.
3	Dossier de la réinscription.
4	Demande de transfert.
5	Remise de diplôme.
6	Pièces scolaires.
7	Pièces scolaires signées.
8	Liste des étudiants.
9	PV modulaires et annuels.
10	Demandes du transfert.
11	Demande de transfert signé.
12	PV modulaires remplis.
13	Diplômes signés.
14	Liste des étudiants par option (spécialité), section, groupe.

Tab. II.2 : Description des flux d'informations

4.6. Règles de gestion de la scolarité de FGEI

- A chaque inscription, l'étudiant doit fournir un dossier administratif d'inscription.
- Après l'inscription, un certificat de scolarité et une carte d'étudiant sont délivrés à l'étudiant. Le dossier d'étudiant sera ensuite archivé à la scolarité.
- La réinscription de l'étudiant se fait selon les PV de délibération après la session de rattrapage.
- La réinscription de l'étudiant est justifiée par la délivrance du certificat de scolarité et d'un relevé de notes.
- En cas de transfert, le dossier de transfert est étudié minutieusement par la commission d'étude de demande de transfert.
- Dans le cas où le transfert serait accepté, le dossier d'inscription doit être transmis à l'établissement d'accueil.
- Les délibérations après session normal auront lieu après les examens du rattrapage.
- Le jury de délibération est constitué de tous les responsables du module.
- Après chaque délibération, un PV de délibération est établi.
- Un PV de délibération doit être affiché et rendu public.
- L'attestation provisoire du diplôme n'est délivrée que si une demande manuscrite est formulée.
- Un étudiant est en état d'abandon s'il ne se présente pas à la réinscription.
- L'exclusion de l'étudiant est prononcée selon la décision du conseil de discipline.
- Un PV d'exclusion sera établi et diffusé aux autres établissements.
- Un étudiant inscrit, qui ne se présente pas aux examens est considéré comme défaillant.
- Edition des tableaux statistiques selon les résultats des étudiants.
- Les étudiants en fin de cycle préparent leurs projets de fin d'étude, un projet porte sur un thème et est encadré par un promoteur.
- Un étudiant demande une autorisation de soutenance.
- Une soutenance est assistée par un président (obligatoire), un ou plusieurs examinateurs (obligatoires) et un promoteur.
- Pour les étudiants fin de cycle, une note de 10/20 au mémoire est obligatoire.

Conclusion

Cette première phase du Processus Unifié nous a permis non seulement d'avoir une vue détaillée de l'état actuel de l'organisme d'accueil, mais aussi de nous familiariser avec les différentes activités et traitements qui se font au sein de la scolarité de génie électrique et informatique.

Il faut noter que les informations acquises au niveau de cette étape nous a donné déjà un premier aperçu sur l'application à concevoir, ouvrant ainsi la porte à la deuxième étape du Processus Unifié intitulé «Analyse et conception», que nous allons détailler dans le prochain chapitre de notre projet.

Chapitre III

Analyse et conception

Introduction

Les techniques de programmation n'ont cessé de progresser depuis l'époque de la programmation en langage binaire à nos jours. Cette évolution a toujours été dictée par le besoin de concevoir et de maintenir des applications toujours plus complexes. Pour ce faire, toute conception d'une application informatique, se doit de suivre et d'adopter une démarche logique et cohérente.

Pour cela nous avons adopté le formalisme UML pour la conception de notre application. Nous avons choisi le langage de modélisation UML car il permet de modéliser de manière claire et précise la structure et le comportement d'un système indépendamment de tout langage de programmation.

1. Présentation d'UML

1.1. Définition

UML (Unified Modelling Language) n'est pas une nouvelle méthode, mais une synthèse des différentes notations que l'on retrouve dans **OMT** (en anglais *Object Modeling Technique*, « technique de modélisation objet »), **OOSE** (Object Oriented Software Engineering), **Booch** (Son nom vient de celui de son concepteur, Grady Booch), qui recouvrent à elles trois plus de la moitié du marché des méthodes objets. UML que ses concepteurs ont voulu simple, est une méthode plus intuitive, plus expressive, plus cohérente, plus homogène que les autres méthodes. Elle est en particulier conçue pour être lisible sur des supports courants et variés, comme le papier, les écrans d'ordinateur, etc. [III.1]

La figure ci-après illustre les étapes d'élaboration d'UML :

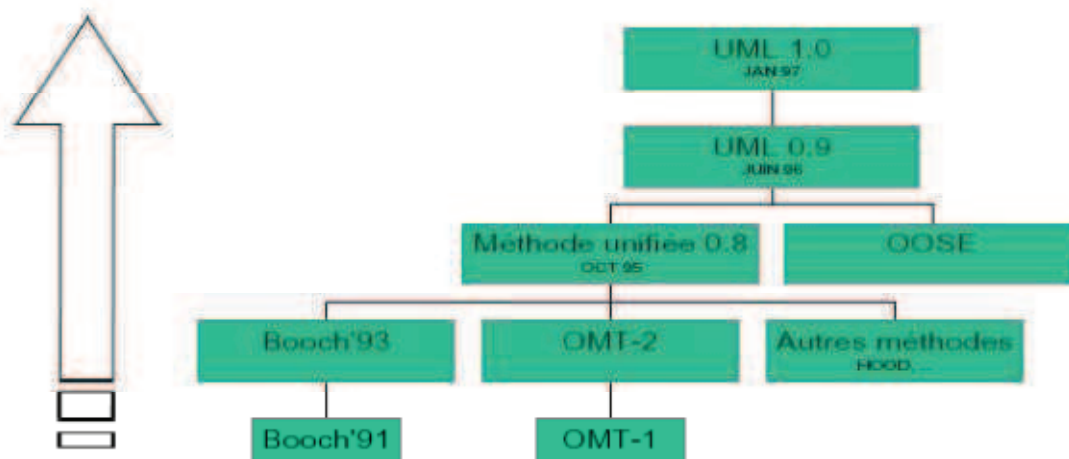


Fig.III.1 : Processus d'unification qui a mené à UML.

Elle a également la particularité d'être une notation ouverte car en tant qu'outil graphique de représentation, elle peut s'appuyer sur n'importe quelle méthode de conception orientée objet comme **OMT** mais aussi, **OOSE** et **Booch**, ... D'autre part, ses concepteurs ont essayé d'éviter d'imposer un formalisme trop contraignant. Une grande liberté est donnée à l'utilisateur pour y ajouter des commentaires avec des notes ou des représentations graphiques supplémentaires au moyen du stéréotypage entre autres.

UML suit une démarche en trois étapes :

- **Analyse**
- **Conception**
- **Implémentation**

Des définitions sur les concepts objets utilisés dans UML sont présentées dans l'*Annexe I*.

1.2. Extension UML pour les applications web

Une extension d'UML est définie par de nouveaux stéréotypes, étiquettes et contraintes. En les combinant, on peut créer de nouvelles briques de base que l'on pourra utiliser dans le modèle.

- **Un stéréotype** est une extension de vocabulaire d'UML qui permet d'associer une nouvelle signification à un élément du modèle. Les stéréotypes peuvent être appliqués à presque tous les éléments du modèle et sont habituellement représentés par une chaîne de caractères entre guillemets (« »). On peut aussi les représenter par une icône.
- **Une étiquette** est une extension des propriétés d'un élément du modèle. La plupart des éléments du modèle possèdent des propriétés ; les classes, par exemple, possèdent, entre autre, un nom, une visibilité et une persistance. L'étiquette est la définition d'une nouvelle propriété d'un élément du modèle. Elle est représentée, dans un diagramme, par une chaîne de caractères entre chevrons (<>).
- **Une contrainte** est une extension de la sémantique d'UML. La contrainte édicte une règle que le modèle doit vérifier pour être qualifié de « bien formé ». Les contraintes sont représentées par des chaînes de caractères entre accolades ({}).

Une extension d'UML comporte une brève introduction, puis la liste des **stéréotypes**, **étiquettes** et **contraintes** avec leur description. Une extension contient aussi un ensemble de règles qui garantissent la cohérence sémantique du modèle. Les spécifications complètes de l'extension d'UML pour application Web sont fournies dans l'*Annexe I*.

2. La démarche d'élaboration de notre projet

Comme la démarche d'élaboration de notre projet s'appuie sur la modélisation UML. Pour se faire nous avons adopté la démarche suivante :

- ✓ A partir de la définition des besoins, nous avons identifié les acteurs et leurs activités, à partir desquels nous avons déduit les cas d'utilisation.
- ✓ A partir des cas d'utilisation nous avons établi un ensemble de scénarios d'utilisation standard à l'aide de diagrammes de séquences.
- ✓ Les diagrammes de séquences sont utilisés pour identifier les classes nécessaires pour construire un diagramme de classes.
- ✓ En dernier lieu nous avons décrit l'architecture physique des composants matériels qui supportent l'exécution du système par un diagramme de déploiement. [III.1]
(*figure III.2*)

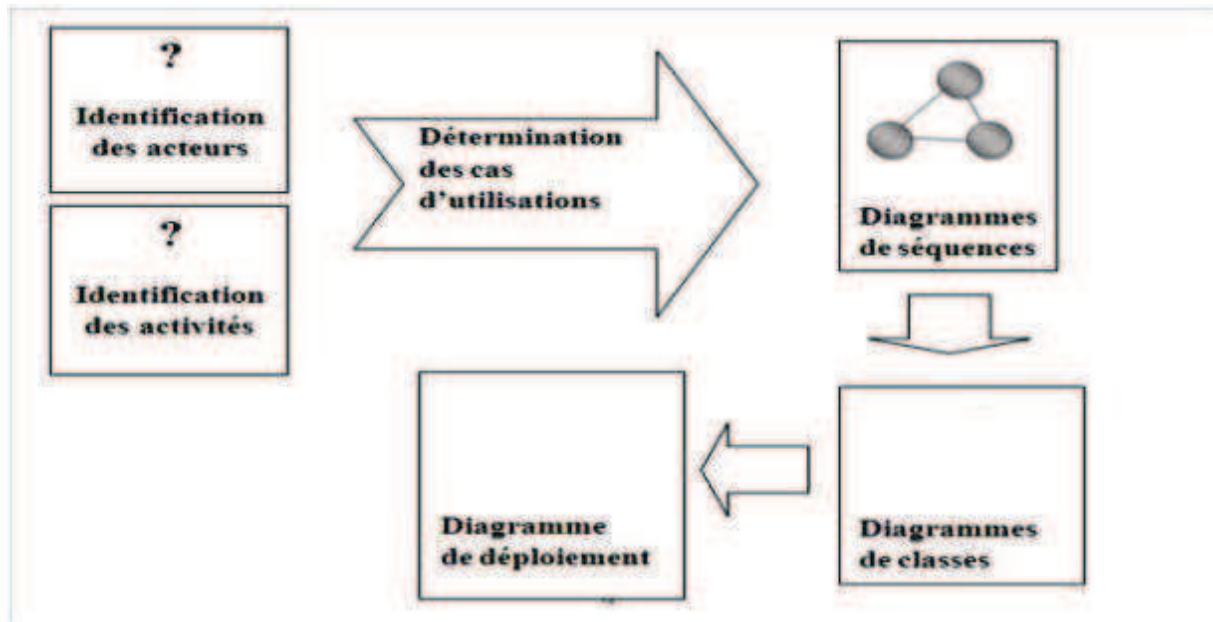


Fig.III.2 : Méthodologie de modélisation du site.

2.1. Diagrammes structurels

Dans les diagrammes structurels on trouve : les diagrammes de classes, les diagrammes d'objets, le diagramme de composants et le diagramme de déploiement.

- **Diagrammes de classes :**

Le diagramme de classe, fournit quasiment l'état final du domaine modélisé avant le passage à la génération des squelettes de composants et du modèle relationnel.

- **Diagrammes d'objets :**

Le diagramme d'objets est une variante du diagramme de classes qui ne représente que les instances de classes instanciables. Pour notre modélisation le diagramme de classes s'avère suffisant.

- **Diagramme de composants :**

Le diagramme de composants permet de décrire les composants du système tels les modules, les programmes et pour notre application nous nous contentons du diagramme de déploiement.

- **Diagramme de déploiement :**

Permet de représenter l'environnement de développement de notre application.

2.2. Diagrammes comportementaux

Les diagrammes comportementaux comportent : les diagrammes de cas d'utilisation, les diagrammes de séquences, les diagrammes d'états transition, les Diagrammes d'activités et le diagramme de collaboration.

- **Diagrammes de cas d'utilisation :**

Nous utilisons les cas d'utilisation pour modéliser les différentes interactions entre les acteurs et l'application. Le but est de modéliser les actions que les acteurs du domaine souhaitent effectuer sur l'application. Ce diagramme nous servira à déterminer le modèle objets sur lequel le système reposera.

- **Diagramme de séquences :**

Le diagramme de séquences est utilisé pour réaliser le passage des cas d'utilisation à la modélisation objets en illustrant la dynamique des cas d'utilisation identifiés.

- **Diagrammes d'états transition :**

Ce diagramme représente le comportement d'un objet ordonnancé par les événements. Dans notre cas nous n'avons pas ce genre d'objets.

- **Diagrammes d'activités :**

Le diagramme d'activités est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activités correspond à sa traduction algorithmique.

Une activité est l'exécution d'une partie d'un cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

- **Diagramme de collaboration :**

Ce diagramme est une variante non temporelle du diagramme de séquences, s'est révélé mieux adapté à notre application, Voir la **figure III.2**.

3. Définition des besoins

3.1. Les besoins non fonctionnelles

a) Les besoins de performance

Ils décrivent les performances d'exécution du système, généralement en termes de temps de réponse. Le temps de chargement des pages web ne devra pas dépasser quelques secondes.

b) Les besoins de disponibilité/fiabilité

Ils concernent le niveau de disponibilité qui doit être explicitement défini pour toute application critique. Notre système doit être opérationnel 24h/24h et 7j/7j.

c) Les besoins de sécurité

Notre système garantira que l'accès aux espaces appropriés pour chaque acteur sera sécurisé par l'authentification par un login et un mot de passe.

d) Les besoins matériels

Ils définissent souvent les configurations matérielles minimales nécessaires au Fonctionnement d'un système.

Notre application ne sera pas très couteuse en termes de besoins matériels, elle Nécessite un réseau intranet, une machine très puissante (serveur), et des machine (clients) dotés d'un processeur Pentium 4, 256 Mo de mémoire, une carte graphique SVGA, des imprimantes avec des palettes de couleurs 16 bits ... sera largement suffisant pour pouvoir utilisé notre application.

3.2. Identification des acteurs

- ❖ **Administrateur** : ce rôle peut être joué par la personne chargée de gérer et mettre à jour la base de données, selon les ordres de chef de service de la scolarité.
- ❖ **chef de service de la scolarité FGEI** : ce rôle peut être joué par la personne qui gère la scolarité de génie électrique et d'informatique.
- ❖ **Chef de département** : Ce rôle peut être joué par la personne chargée des services de département (Automatique, Electronique, Electrotechnique, Informatique).
- ❖ **Agent de la scolarité d'un département** : ce rôle peut être joué par la personne chargée des services de la scolarité de département (Automatique, Electronique, Electrotechnique, Informatique).
- ❖ **Enseignant** : joue un rôle important dans cette application autrement dit la gouvernance des séances (en vacation et hors vacation) effectuées au sein de la FGEI.
- ❖ **Etudiant** : ce rôle peut être joué seulement par les étudiants inscrits au sein de la FGEI.

3.3. Identification des cas d'utilisations

Les cas d'utilisations (en anglais uses cases) permettent de représenter le fonctionnement du système vis-à-vis de l'utilisateur, c'est donc une vue du système dans son environnement extérieur.

Le tableau suivant résume les cas d'utilisations pour chaque acteur qui est défini :

Acteurs	Cas d'utilisation
Administrateur	T : Se connecter au site. T : S'identifier T : Saisir des paramètres généraux de l'application T : Gérer les utilisateurs de l'application. T : Rechercher un utilisateur. T : Editer des tableaux statistiques. T : Modifier son login et son mot de passe. T : Créer les sondages. T : Contacter un utilisateur.
Agent de la scolarité d'un département	T : Se connecter au site. T : S'identifier T : Editer les listes des étudiants inscrits dans son département. T : Saisir/Modifier de nouveaux inscrits. T : Saisir/Modifier des réinscriptions. T : Saisir/Modifier des transférés. T : Saisir/Modifier des abandons. T : Rechercher un étudiant. T : Editer des documents. T : Modifier son login et son mot de passe T : Contacter un utilisateur.

Chef de service de la scolarité FGEI	<p>T : Se connecter au site.</p> <p>T : S'identifier</p> <p>T : Faire les tâches de l'agent de la scolarité d'un département.</p> <p>T : Supprimer des étudiants inscrits.</p> <p>T : Supprimer des étudiants réinscrits.</p> <p>T : Supprimer des abandons.</p> <p>T : Supprimer des transférés.</p>
Chef de département	<p>T : Se connecter au site.</p> <p>T : S'identifier.</p> <p>T : Gérer les listes d'étudiants par sections et groupes</p> <p>T : Gérer les formations offertes</p> <p>T : Gérer les enseignants</p> <p>T : Gérer les soutenances.</p> <p>T : Saisir les paramètres généraux de l'application.</p> <p>T : Rechercher un étudiant.</p> <p>T : Editer les tableaux statistiques.</p> <p>T : Modifier son login et son mot de passe.</p> <p>T : Contacter un utilisateur.</p>
Enseignant	<p>T : Se connecter au site.</p> <p>T : S'identifier.</p> <p>T : Gérer les PV modulaires de ses étudiants.</p> <p>T : Editer les formations offertes.</p> <p>T : Rechercher un étudiant.</p> <p>T : Editer la liste de ses étudiants</p> <p>T : Editer les tableaux statistiques.</p> <p>T : Modifier son Profil.</p> <p>T : Contacter un utilisateur.</p>

Etudiant	<p>T : Se connecter au site.</p> <p>T : Création d'un nouveau compte (espace) étudiant s'il n'a pas encore créé.</p> <p>T : S'identifier.</p> <p>T : Editer les formations offertes.</p> <p>T : Consulter ses notes (résultats) durant son cursus universitaire.</p> <p>T : Editer des tableaux statistiques.</p> <p>T : Modifier son Profil.</p> <p>T : Contacter un utilisateur.</p> <p>T : Participer aux sondages.</p>

Tab III.1. Identification des cas d'utilisations

3.4. Spécification des scénarios

Un scénario est une instance d'un cas d'utilisation dans laquelle toutes les conditions relatives aux différents événements ont été fixées, chaque cas d'utilisation donné correspondent plusieurs scénarios.

Acteurs	Cas d'utilisation	Scénarios
Administrateur	T : Se connecter au site.	<ul style="list-style-type: none"> ✓ Saisir l'URL du site dans un navigateur. ✓ Valider la saisie.
	T : S'identifier	<ul style="list-style-type: none"> ✓ Connecter au site. ✓ Remplir le formulaire d'authentification. ✓ Valider.
	T : Créer les comptes utilisateurs.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Ajouter un compte » ✓ Remplir le formulaire associé. ✓ Valider la saisie
		<ul style="list-style-type: none"> ✓ Sélectionner le lien « Modifier un compte dans la liste des utilisateurs » ✓ Remplir le formulaire associé.

	T : Modifier des comptes	✓ Valider la saisie
	T : Supprimer des comptes utilisateurs	✓ Sélectionner le lien « Supprimer un compte » dans la liste des utilisateurs ✓ Valider la suppression.
	T : Rechercher des utilisateurs	✓ Remplir le formulaire recherche. ✓ Valider la saisie
	T : Editer les tableaux statistiques	✓ Sélectionner le lien « statistiques ». ✓ Afficher les statistiques.
	T : Modifier son Profil	✓ Sélectionner le lien « Mon profil ». ✓ Remplir le formulaire. ✓ Valider la saisie.

Agent de la scolarité d'un département	T : Se connecter au site.	✓ Saisie l'URL de site dans le navigateur choisi. ✓ Valider la saisie.
	T : S'identifier	✓ Connecter au site. ✓ Remplir le formulaire d'authentification. ✓ Valider.
	T : Saisir les nouveaux inscrits.	✓ Sélectionner le lien « inscriptions ». ✓ Remplir le formulaire d'inscription. ✓ Valider la saisie.
	T : Saisie des réinscriptions.	✓ Sélectionner le lien « réinscription ». ✓ Sélectionner l'année pédagogique ✓ Remplir le formulaire de réinscription. Valider la saisie.
	T : Saisir les transférés.	✓ Sélectionner le lien « Ajouter un transféré ». ✓ Remplir le formulaire associé ✓ Valider la saisie
	T : Rechercher des étudiants	✓ Remplir le formulaire recherche ✓ Valider la saisie
	T : Edition des listes des étudiants	✓ Sélectionner le lien « liste des étudiants ». ✓ Sélectionner l'année pédagogique et l'année universitaire. ✓ Afficher la liste. ✓ Valider la saisie.

	T : Edition Documents	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Edition Documents ». ✓ Sélectionner le type de document ✓ Remplir le formulaire.
	T : Modifier son profil.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Changer le mot de passe ». ✓ Remplir le formulaire de changement de mot de passe. ✓ Valider la saisie.

Chef de Service de la scolarité FGEI	T : Se connecter au site.	<ul style="list-style-type: none"> ✓ Saisir l'URL du site dans un navigateur. ✓ Valider la saisie.
	T : S'identifier	<ul style="list-style-type: none"> ✓ Connecter au site. ✓ Remplir le formulaire d'authentification. ✓ Valider la saisie.
	T : Faire les taches de l'agent d'un département	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Départements ». ✓ Sélectionner un département dans la liste. ✓ Accéder aux taches des agents de la scolarité.
	T : Supprimer un étudiant inscrit	<ul style="list-style-type: none"> ✓ Sélectionner le lien « supprimer » dans la liste des étudiants. ✓ Valider la suppression.
	T : Supprimer un transféré.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « supprimer » dans la liste des transférés. ✓ Valider la suppression.
	T : Supprimer un abandon.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « supprimer » dans la liste des abandons. ✓ Valider la suppression.
	T : Modifier son Profil	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Mon profil ». ✓ Remplir le formulaire. ✓ Valider la saisie.

	T : Se connecter au site.	<ul style="list-style-type: none"> ✓ Saisie l'URL de site dans le navigateur choisie. ✓ Valider la saisie.
	T : S'identifier.	<ul style="list-style-type: none"> ✓ Remplir le formulaire d'authentification. ✓ Valider la saisie.
	T : Saisir une formation.	<ul style="list-style-type: none"> ✓ Sélectionner le lien «Ajouter formation ». ✓ Remplir le formulaire de formation.

Chef de département		✓ Valider la saisie.
	T : Modifier une formation.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « modifier formation » dans la liste des formations. ✓ Remplir le formulaire de modification. ✓ Valider la saisie.
	T : Supprimer une formation.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « supprimer formation » dans la liste des formations. ✓ Valider la suppression.
	T : Ajouter un enseignant	<ul style="list-style-type: none"> ✓ Sélectionner le lien «Ajouter enseignant ». ✓ Remplir le formulaire. ✓ Valider la saisie.
	T : Modifier un enseignant	<ul style="list-style-type: none"> ✓ Sélectionner le lien «Modifier enseignant » dans la liste des enseignants. ✓ Remplir le formulaire. ✓ Valider la saisie.
	T : Supprimer un enseignant	<ul style="list-style-type: none"> ✓ Sélectionner le lien «Supprimer enseignant » dans la liste des enseignants. ✓ Valider la suppression.
	T : Saisir des unités d'enseignements et les matières.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Unités et Matières ». ✓ Sélectionner une formation. ✓ Remplir le formulaire d'ajout d'une UE puis le formulaire d'ajout d'une matière. ✓ Valider la saisie.
	T : Modifier les unités d'enseignements et les matières.	<ul style="list-style-type: none"> ✓ Sélectionner le lien «Modifier Unités et Matières » dans la liste des UE et Matières. ✓ Remplir le formulaire associé. ✓ Valider la saisie.
	T : Supprimer les unités d'enseignements et les matières.	<ul style="list-style-type: none"> ✓ Sélectionner le lien «Supprimer Unités et Matières » dans la liste des UE et Matières. ✓ Valider la suppression.
	T : Etablir les PV de soutenance.	<ul style="list-style-type: none"> ✓ Sélectionner le lien «soutenance ». ✓ Remplir le formulaire d'établissement des PV de soutenance. ✓ Valider la saisie.
	T : Mise à jour de la liste des soutenus.	<ul style="list-style-type: none"> ✓ Sélectionner le lien «la liste des soutenus » ✓ Remplir le formulaire de modification. ✓ Valider la saisie.

	T : Supprimer les soutenances	<ul style="list-style-type: none"> ✓ Sélectionner le lien «Supprimer soutenance » ✓ Valider la suppression.
	T : Editer les listes des étudiants	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Liste étudiants ». ✓ Sélectionner l'année pédagogique et l'année universitaire. ✓ Afficher la liste des étudiants.
	T : Editer les statistiques.	<ul style="list-style-type: none"> ✓ Sélectionner le lien «statistiques ». ✓ Afficher les statistiques.
	T : Saisir des paramètres généraux.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Paramètres généraux ». ✓ Remplir les formulaires des paramètres. ✓ Valider la saisie.
	T : Modifier son profil.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Mon profil ». ✓ Remplir le formulaire. ✓ Valider la saisie.

Enseignant	T : Se connecter au site.	<ul style="list-style-type: none"> ✓ Saisie l'URL de site dans le navigateur choisi. ✓ Valider la saisie.
	T : S'identifier.	<ul style="list-style-type: none"> ✓ Remplir le formulaire d'authentification. ✓ Valider la saisie.
	T : Saisir ses PV modulaires	<ul style="list-style-type: none"> ✓ Sélectionner le lien « PV modulaire ». ✓ Sélectionner un PV. ✓ Saisie des notes. ✓ Enregistrer Les notes. ✓ Valider le PV modulaire
	T : Modifier ses PV modulaires	<ul style="list-style-type: none"> ✓ Sélectionner le lien « modifier Notes» dans le PV modulaire. ✓ Modifier la note de l'étudiant. ✓ Valider la saisie.
	T : Editer les listes de ses étudiants	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Liste étudiants ». ✓ Sélectionner une année pédagogique. ✓ Afficher la liste des étudiants.
	T : Editer les tableaux statistiques	<ul style="list-style-type: none"> ✓ Sélectionner le lien «statistiques ». ✓ Afficher les statistiques.
	T : Modifier son Profil.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Mon profil ». ✓ Remplir le formulaire. ✓ Valider la saisie.

Etudiant	T : Se connecter au site.	<ul style="list-style-type: none"> ✓ Saisie l'URL de site dans le navigateur choisi. ✓ Valider la saisie.
	T : Création d'un compte étudiant	<ul style="list-style-type: none"> ✓ Connecter au site. ✓ Sélectionner le lien « créer un compte étudiant » ✓ Remplir les formulaires. ✓ Valider.
	T : S'identifier.	<ul style="list-style-type: none"> ✓ Remplir le formulaire d'authentification. ✓ Valider la saisie.
	T : Editer Les formations offertes.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « formations ». ✓ Afficher la présentation de la formation
	T : Editer les unités d'enseignements et les matières par formation.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « UE et matières ». ✓ Sélectionner une formation. ✓ Afficher les UE et les matières
	T : Editer ses résultats.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Mes résultats ». ✓ Sélectionner une année pédagogique. ✓ Afficher les résultats
	T : Editer les tableaux statistiques.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « statistiques ». ✓ Afficher les statistiques.
	T : Modifier son profil.	<ul style="list-style-type: none"> ✓ Sélectionner le lien « Mon profil ». ✓ Remplir le formulaire. ✓ Valider la saisie.

Tab.III.2. Spécification des scénarios

3.5. Diagrammes représentatifs

3.5.1. Diagramme de contexte de l'application

Le diagramme de contexte permet de spécifier le nombre d'instances d'acteurs connectés à ce système à un moment donné. La définition des besoins nous a permis de dégager le diagramme de contexte suivant :

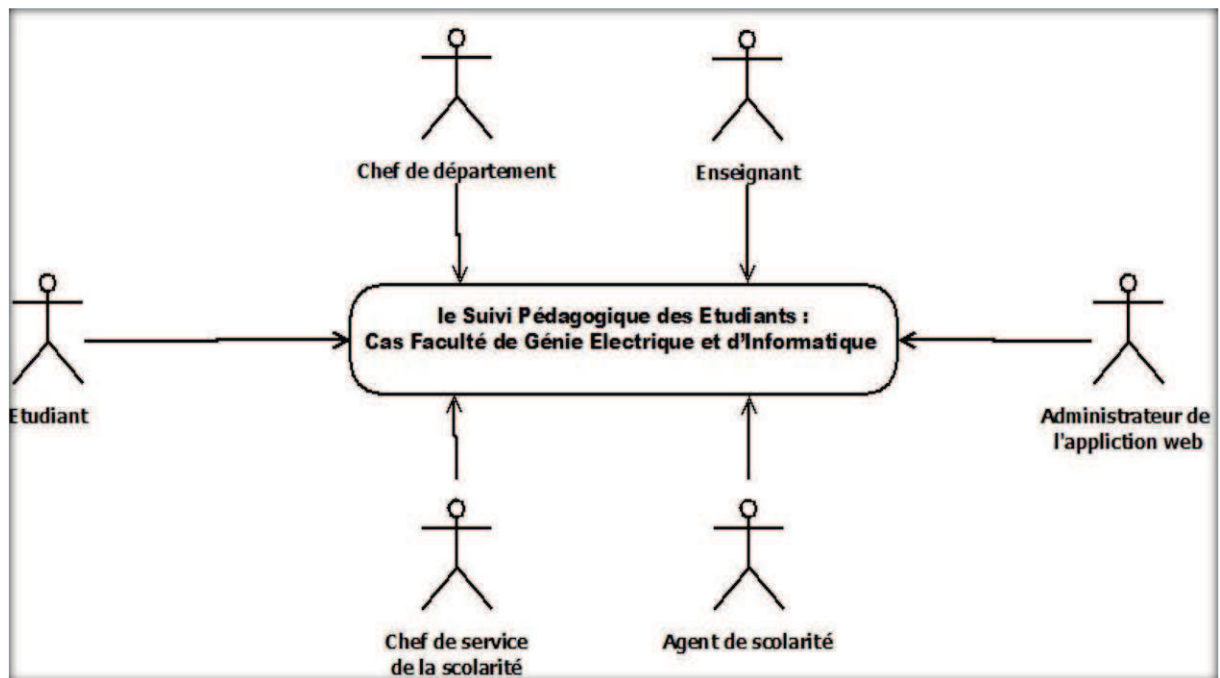


Fig.III.3 : Diagramme de contexte de l'application

3.5.2. Diagrammes des cas d'utilisation

3.5.2.1. Diagramme des cas d'utilisation Administrateur de l'application

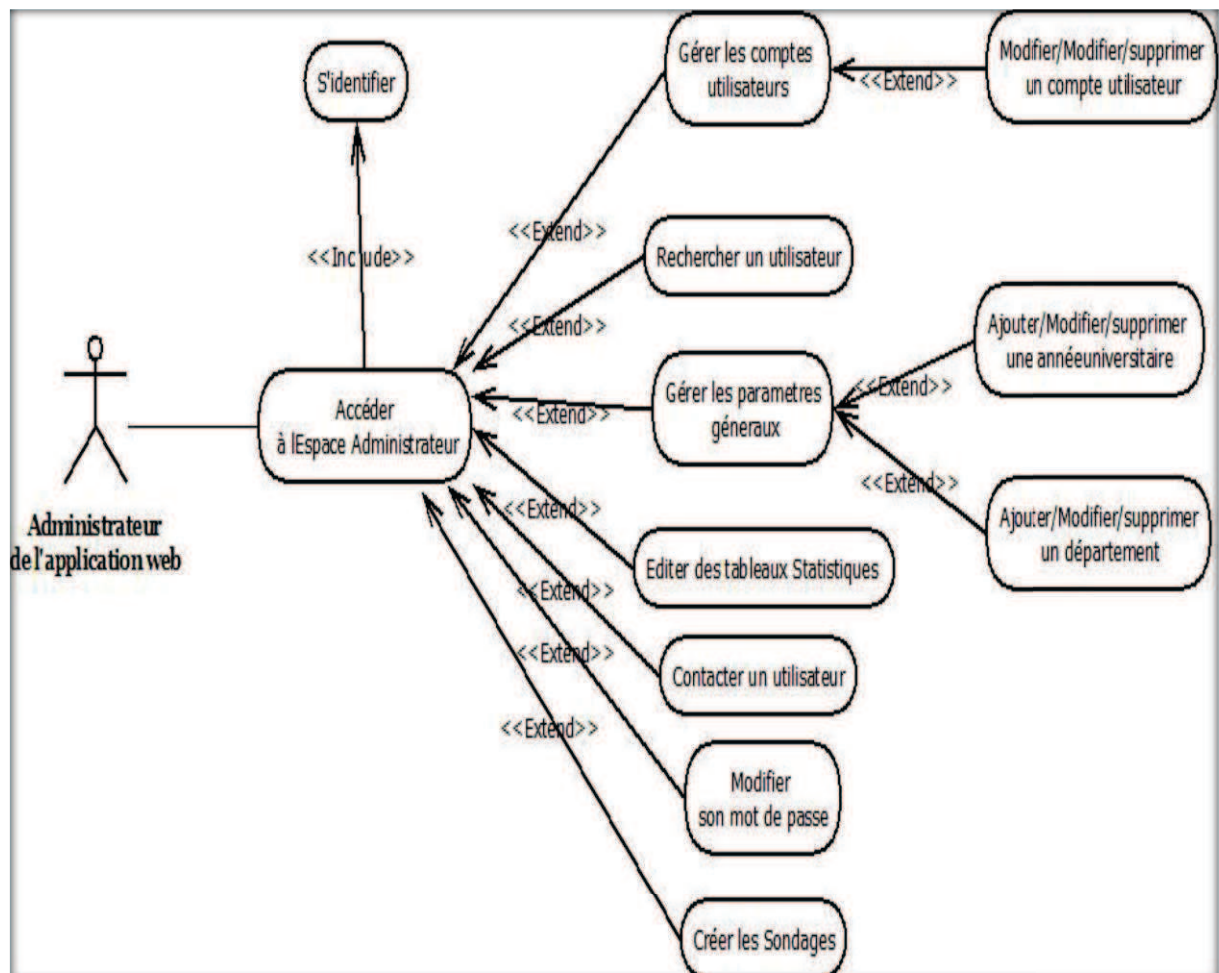


Fig.III.4 : Diagramme des cas d'utilisation Administrateur de l'application

3.5.2.2. Diagramme des cas d'utilisation de l'agent de la scolarité

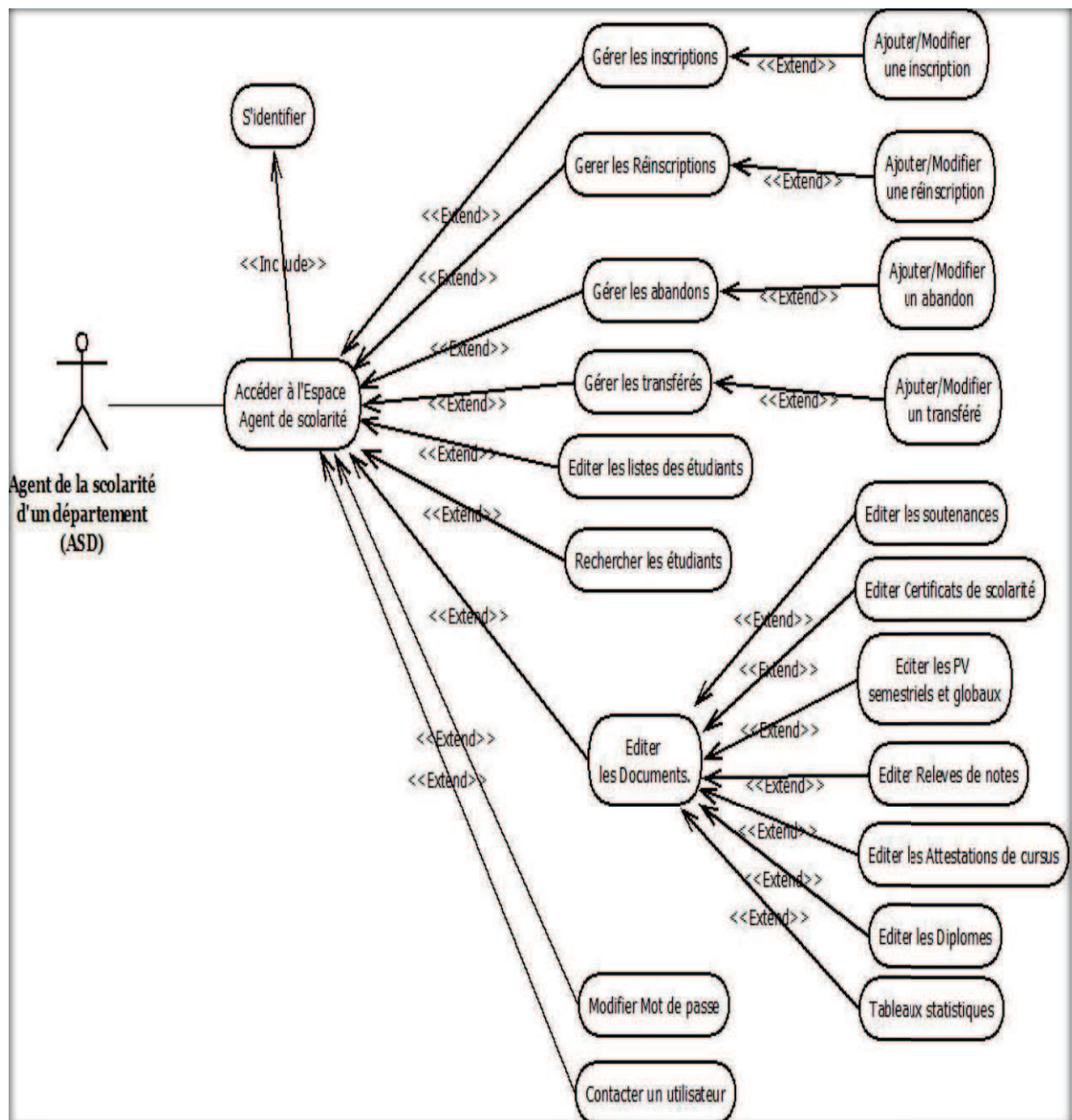


Fig.III.5 : Diagramme des cas d'utilisation de l'agent de la scolarité

3.5.2.3. Diagramme des cas d'utilisation Chef de Service de la Scolarité

*Fig.III.6 : Diagramme des cas d'utilisation Chef de Service de la scolarité*

3.5.2.4. Diagramme des cas d'utilisation de chef de département

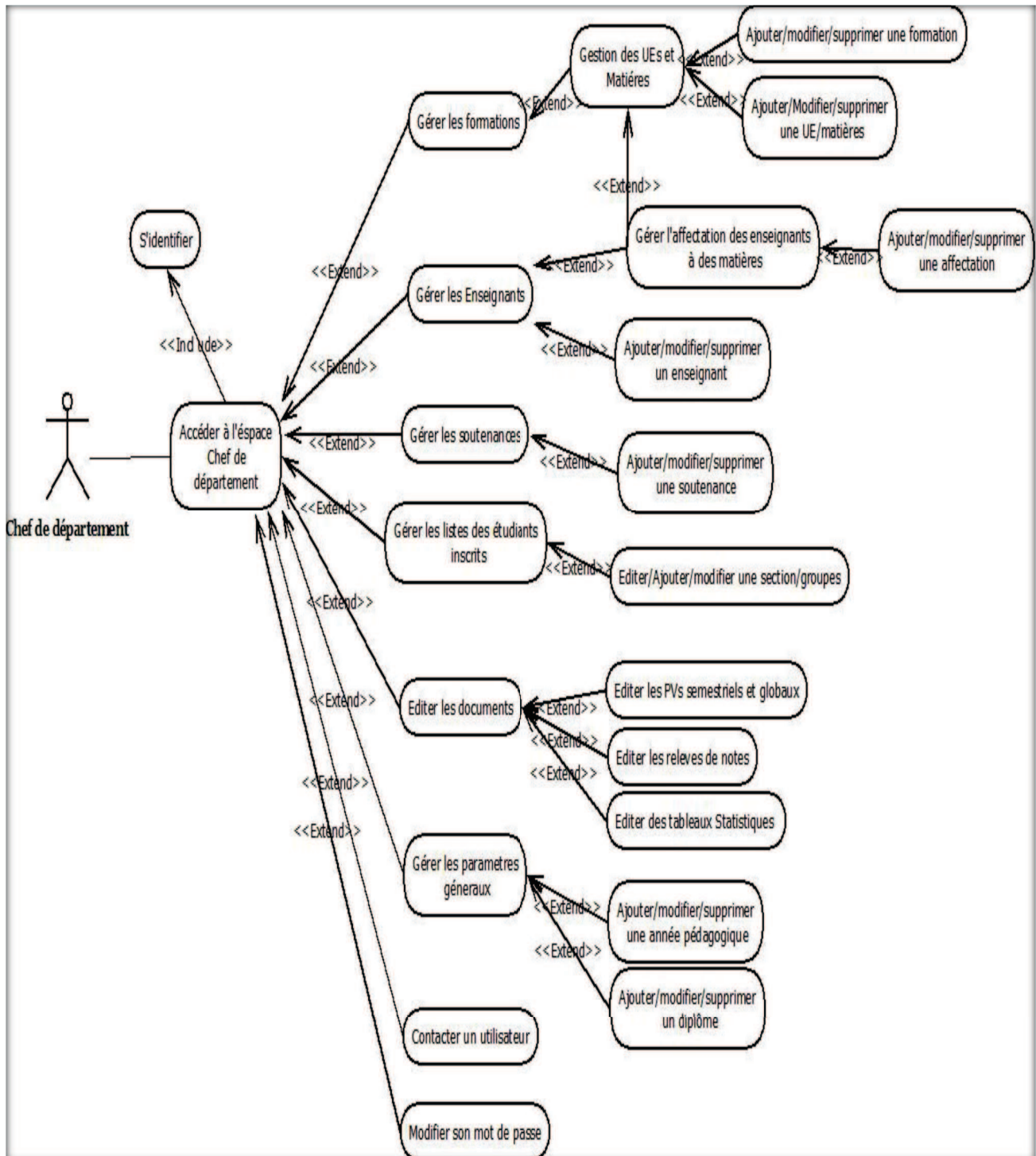


Fig.III.7 : Diagramme des cas d'utilisation chef de département

3.5.2.5. Diagramme des cas d'utilisation Enseignant

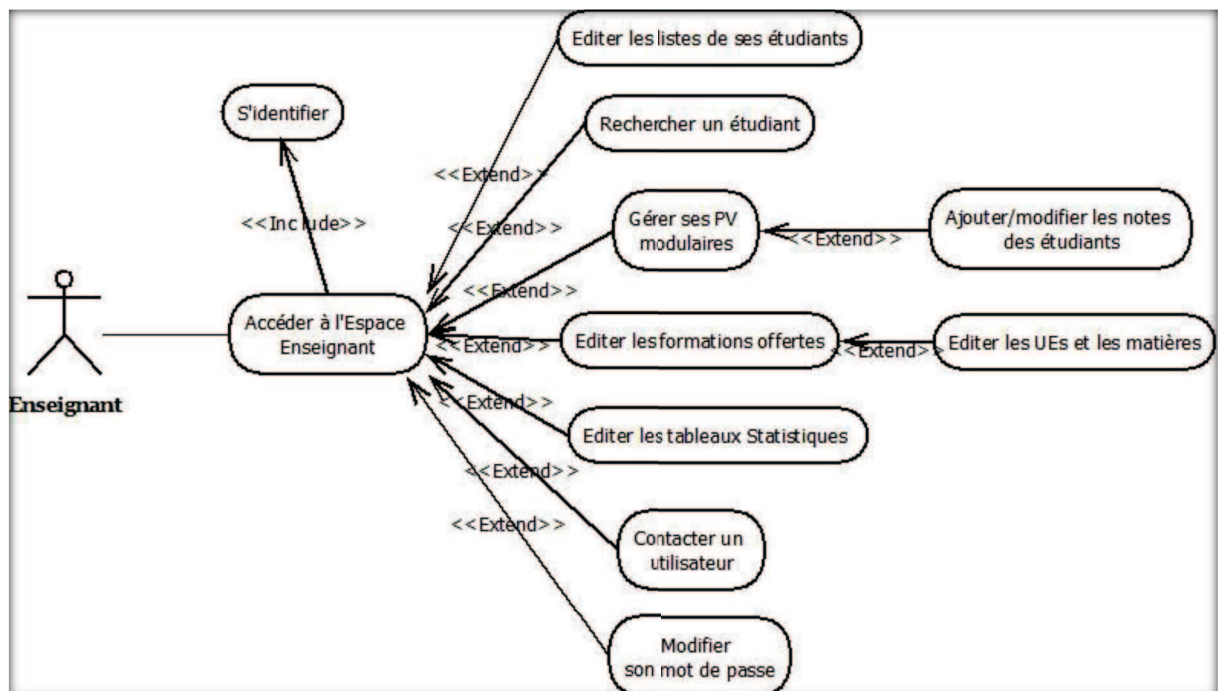


Fig.III.8 : Diagramme des cas d'utilisation Enseignant

3.5.2.6. Diagramme des cas d'utilisation Etudiant

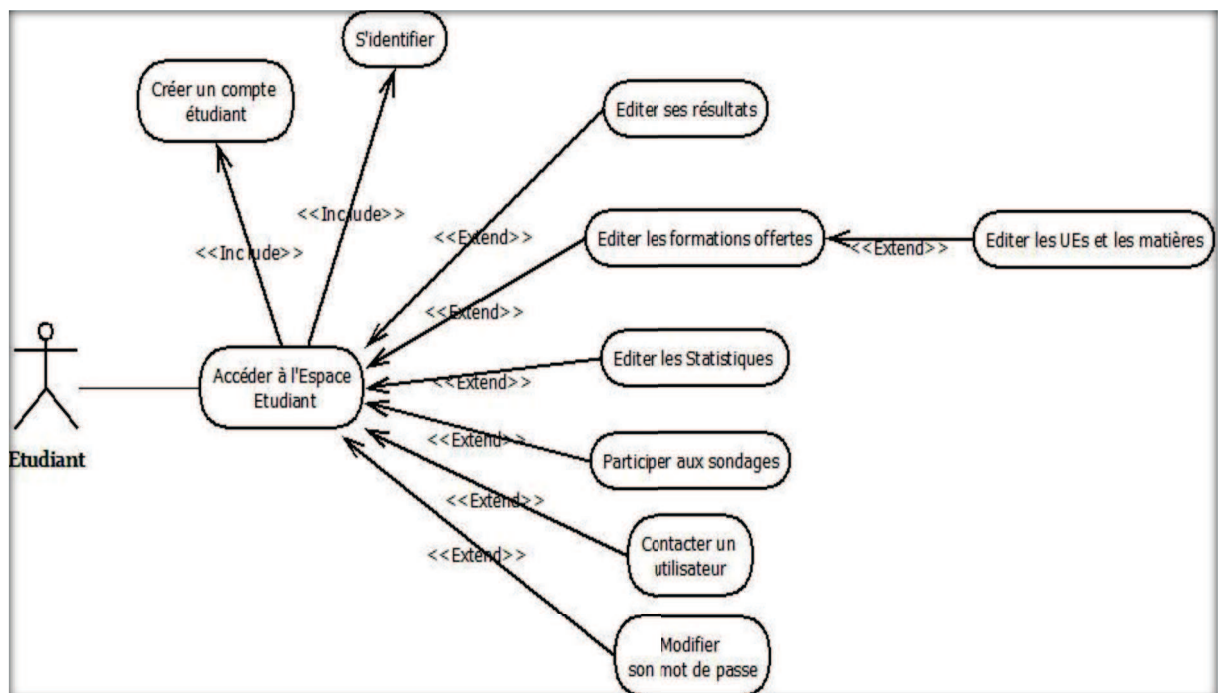


Fig.III.9 : Diagramme des cas d'utilisation Etudiant

3.5.3. Diagrammes de séquence

❖ Diagramme séquence simple Cas utilisation « identification » :

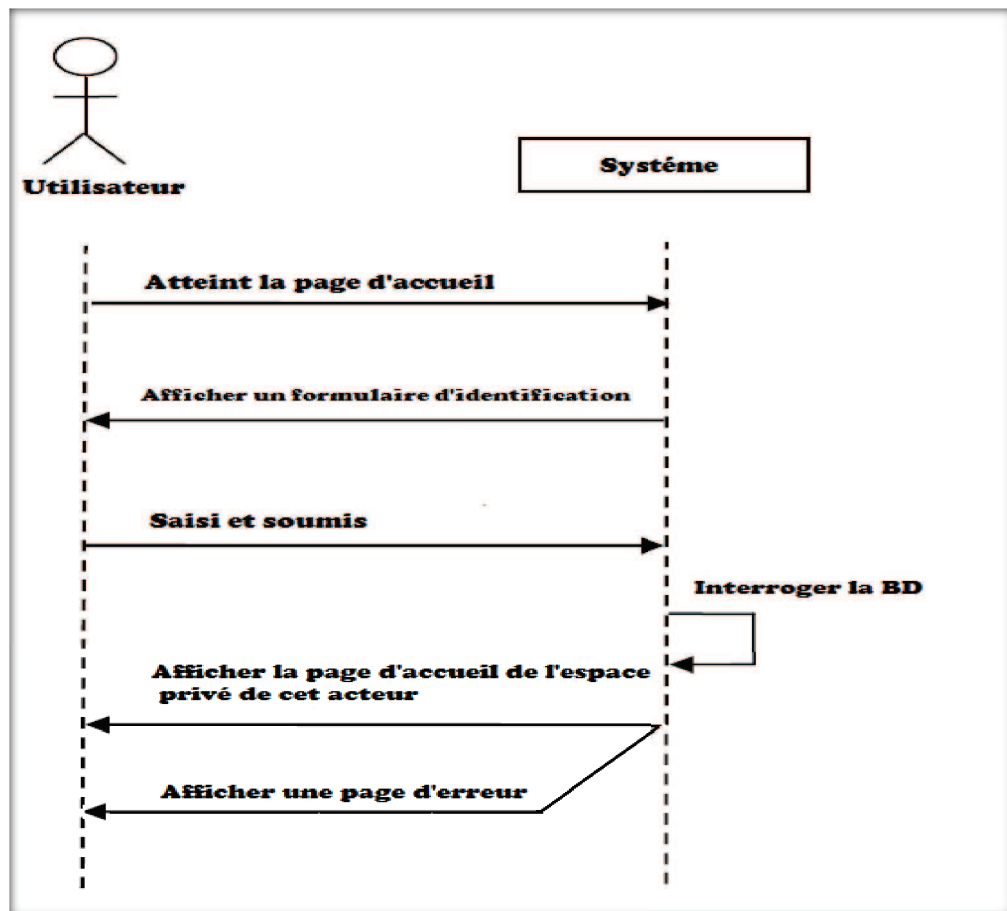


Fig.III.10 : Diagramme de séquence simple du cas d'utilisation « identification »

❖ Diagramme séquence simple Cas d'utilisation « changer mon profil » :

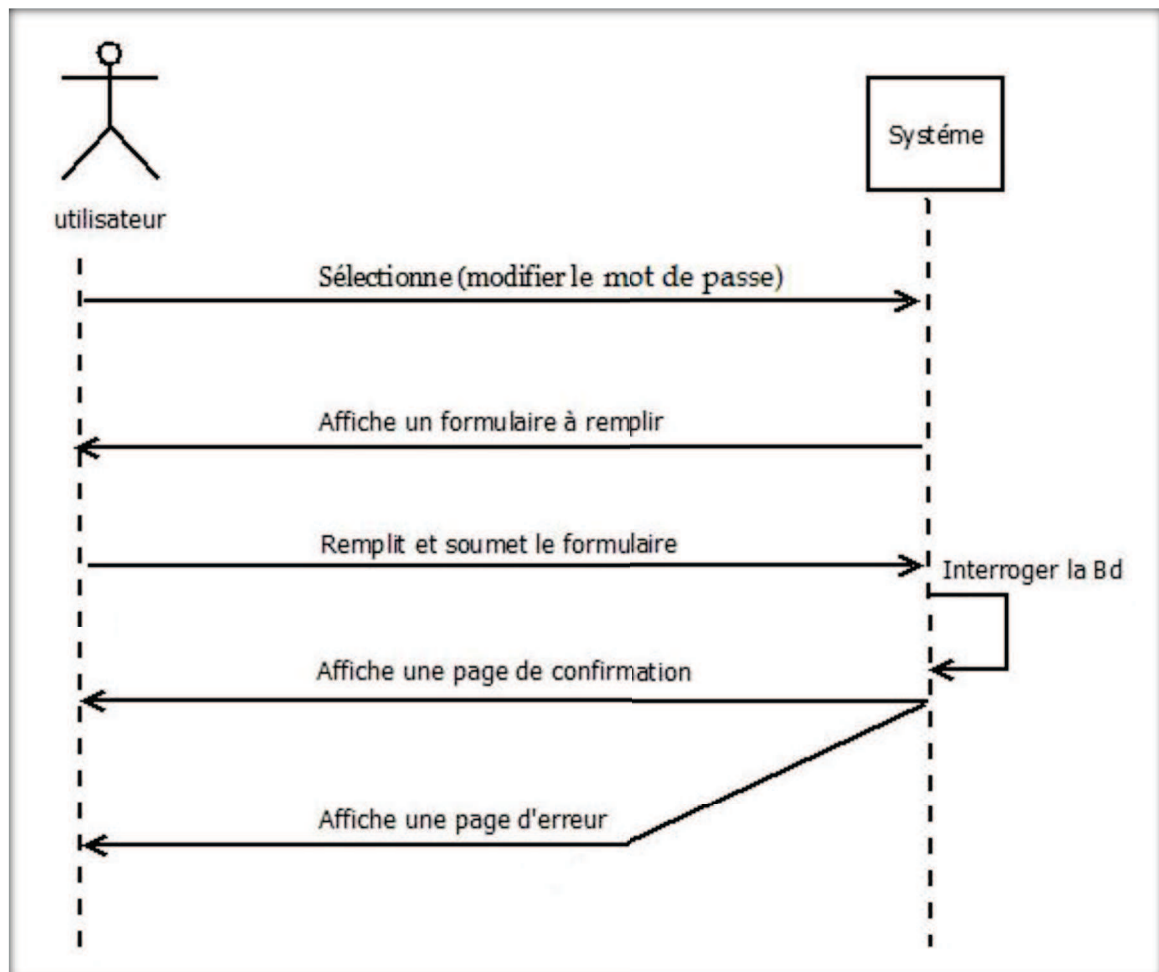


Fig.III.11 : Diagramme séquence simple du cas d'utilisation « Modifier le mot de passe »

❖ Diagramme séquence simple Cas d'utilisation « Inscrire étudiant » :

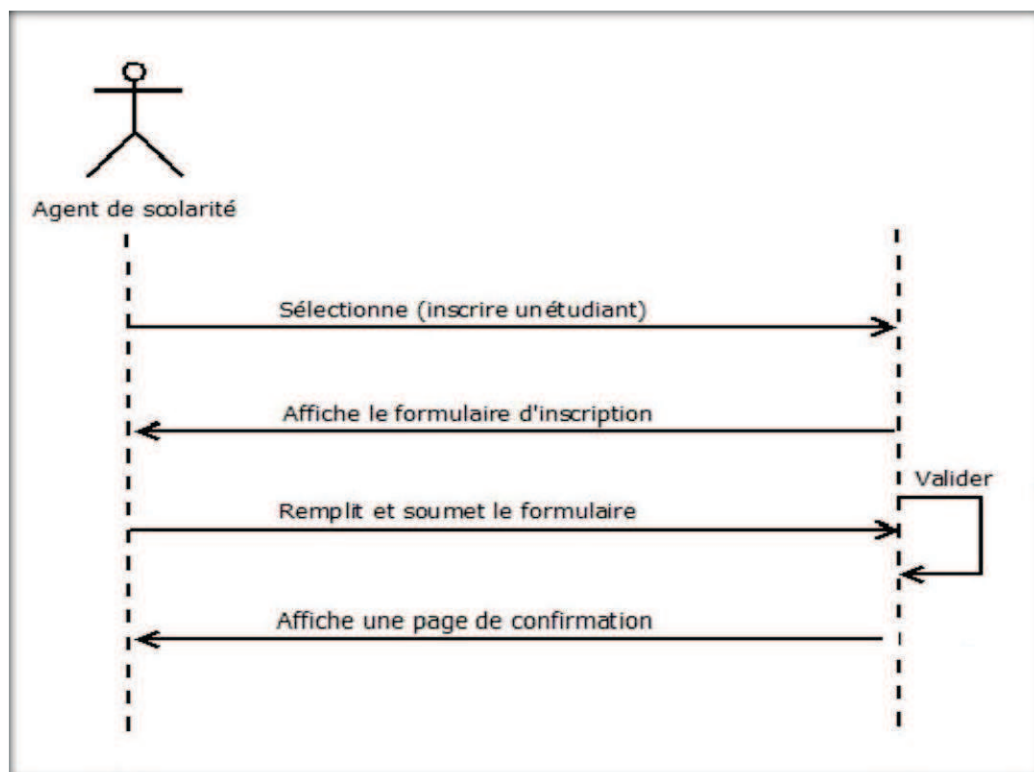


Fig.III.12 : Diagramme séquence simple du cas d'utilisation « Inscrire étudiant ».

4. Analyse

L'analyse comprend les activités qui permettent précisément d'aboutir au modèle d'analyse du système en partant des cas d'utilisations. Le modèle d'analyse est constitué par les classes et les collaborations des classes qui traduisent les comportements dynamiques.

4.1. Les diagrammes de séquence d'analyses détaillés du cas d'utilisation

4.1.1 Diagramme de séquence d'analyse détaillé du Cas d'utilisation « identification »

Ce cas d'utilisation contient les objets suivants :

Objet interface : Identification, accueil privé.

Objet contrôle : Vérification.

Objet entité : Utilisateur.

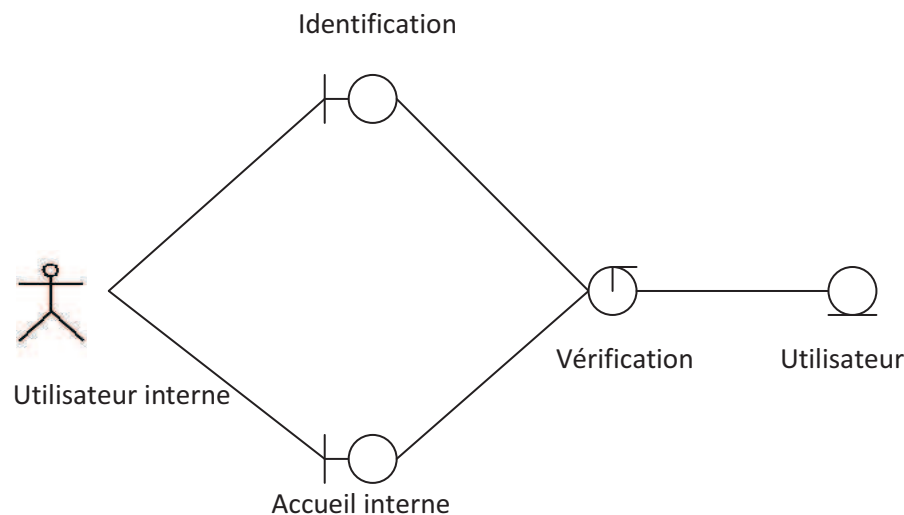


Fig.III.13 : Diagramme robustesse cas d'utilisation « Identification »

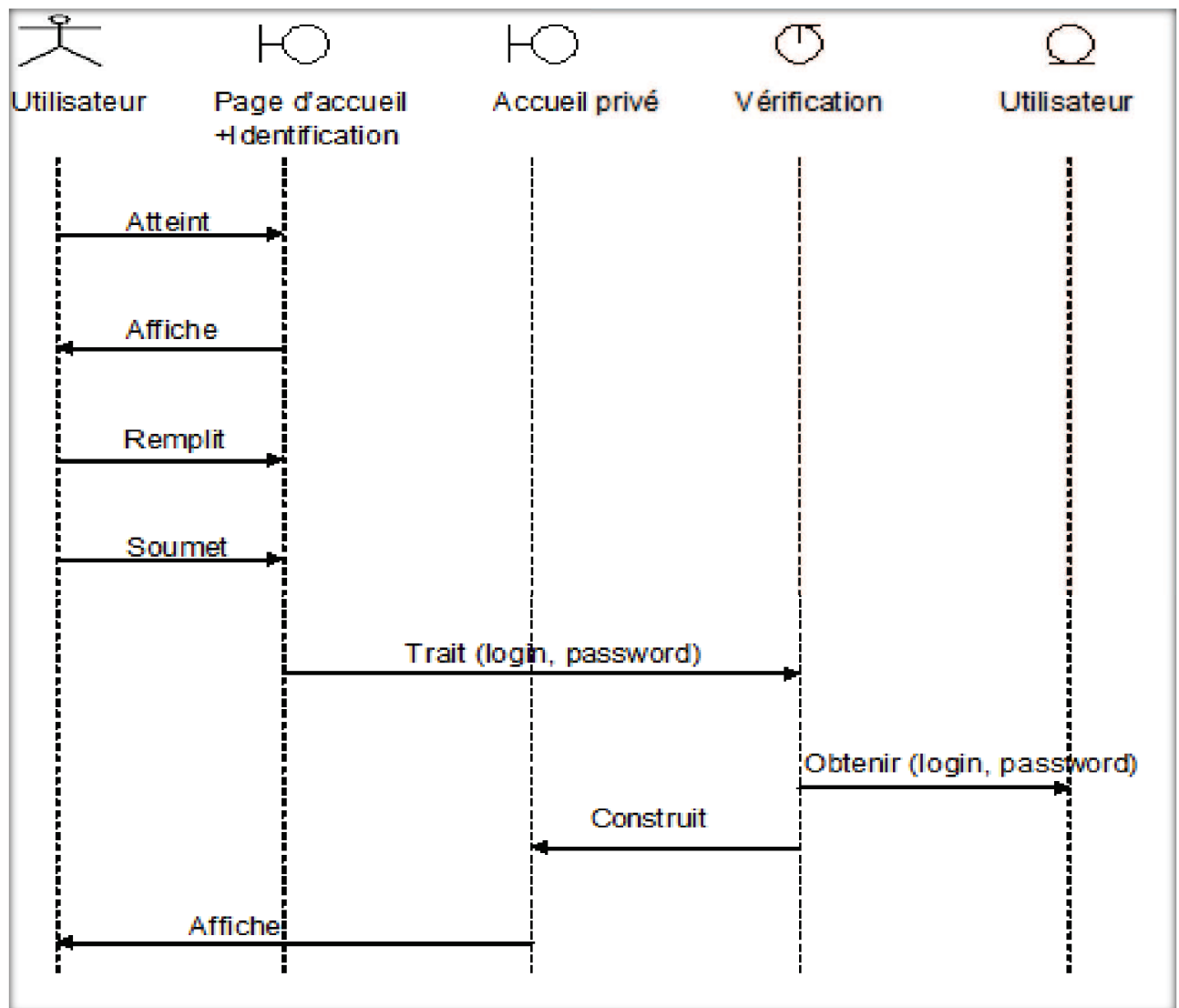


Fig.III.14 : Diagramme de séquence détaillé Cas d'utilisation « Identification »

4.1.2 Diagramme de séquence d'analyse détaillé du Cas d'utilisation « Inscrire étudiant »

Ce cas d'utilisation contient les objets suivants :

Objet interface : Page accueil privé, formulaire inscrire étudiant, page confirmation inscrire étudiant.

Objet contrôle : Inscrire étudiant.

Objet entité : Etudiant.

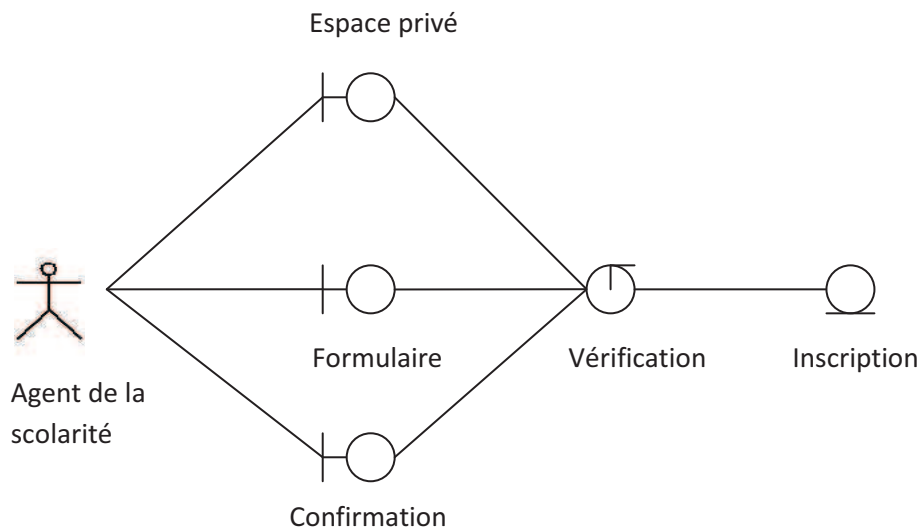


Fig.III.15 : Diagramme robustesse Cas d'utilisation « Inscrire étudiant »

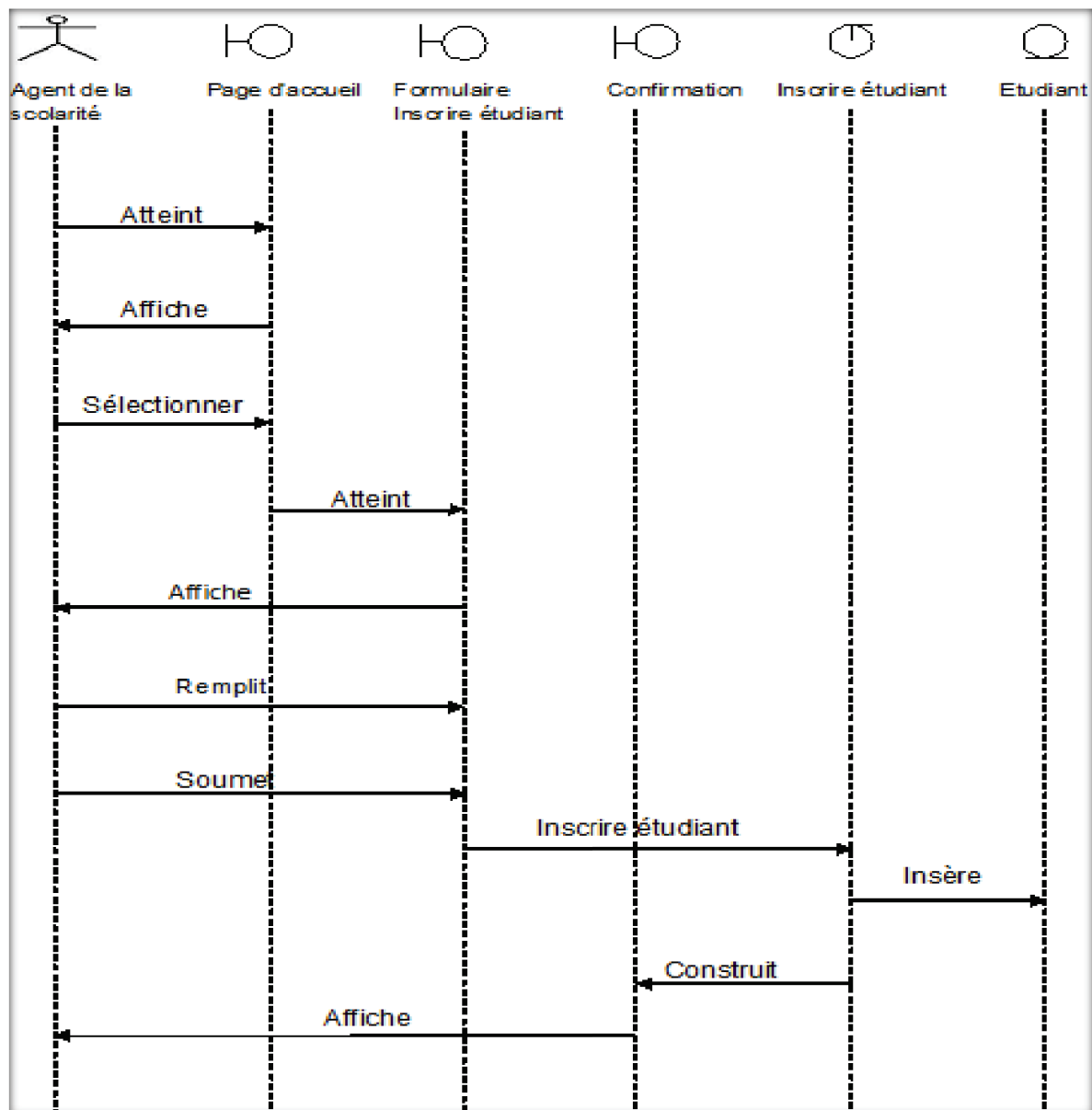


Fig.III.16 : Diagramme de séquence détaillé Cas d'utilisation « Inscrire étudiant »

4.1.3 Diagramme de séquence d'analyse détaillé du Cas d'utilisation « Ajouter Unité d'enseignement »

Ce cas d'utilisation contient les objets suivants :

Objet interface : Page accueil privé, formulaire ajouter UE/matière, page confirmation.

Objet contrôle : Ajouter UE/matière.

Objet entité : UE/matière.

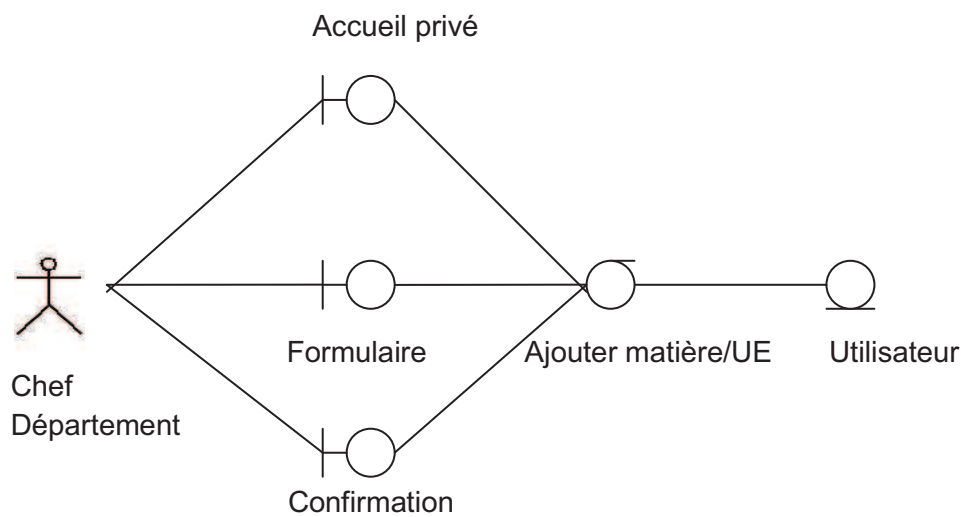


Fig.III.17 : Diagramme robustesse Cas d'utilisation « Ajouter Unité d'enseignement »

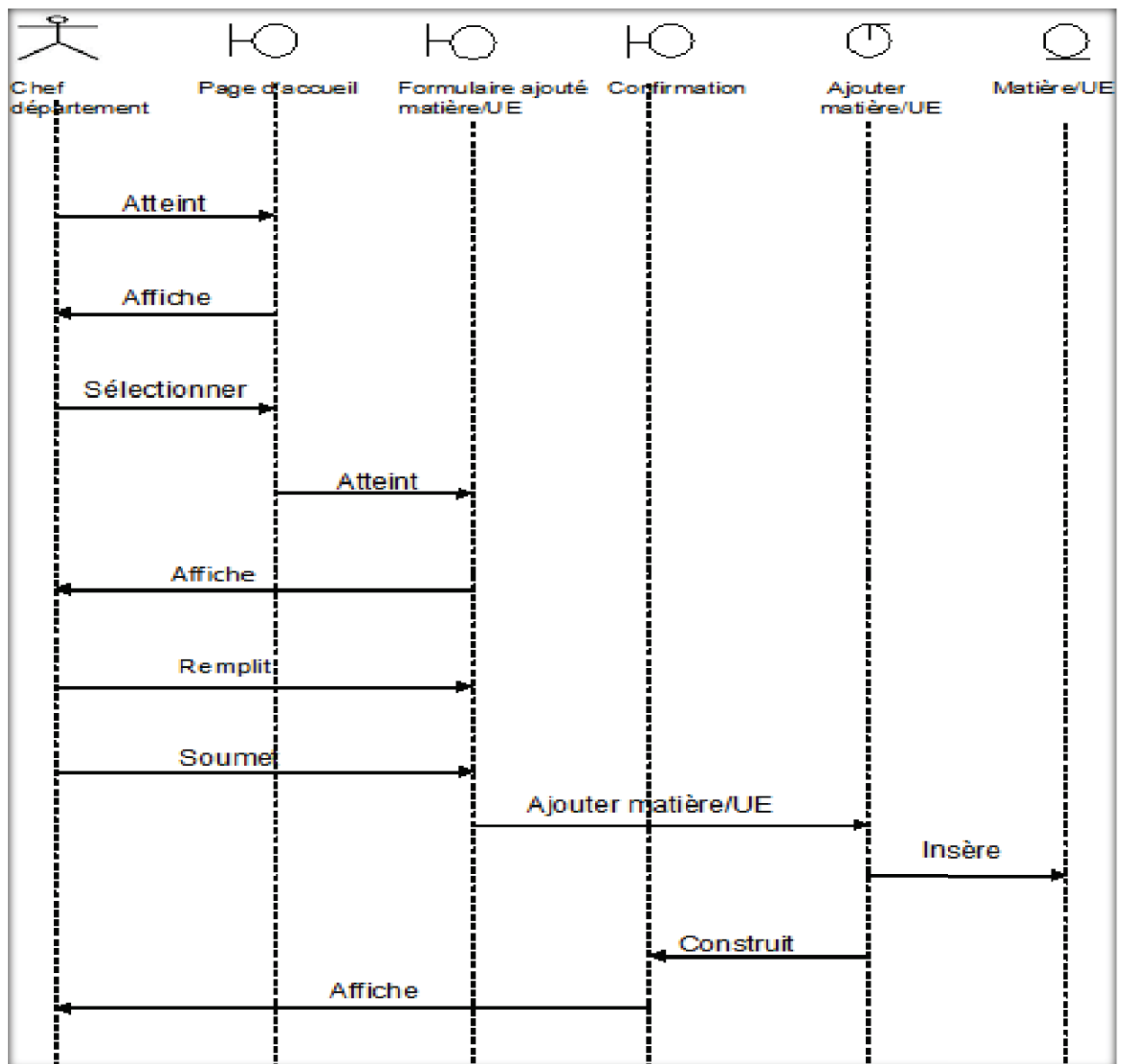


Fig.III.18 : Diagramme de séquence détaillé Cas d'utilisation « Ajouter UE/Matières »

5. Conception

La conception exprime les modules et les exécutable physiques sans aller à la réalisation concrète du système.

Dans la phase de conception, nous représenterons quelques diagrammes de Séquences de réalisation des cas d'utilisation déjà décrits et la conception de notre base de données par le modèle entité/association suivi du modèle relationnel lui correspondant.

5.1. Diagramme de séquence de réalisation des cas d'utilisation**5.1.1. Diagramme de séquence de réalisation des cas d'utilisation
« Identification »**

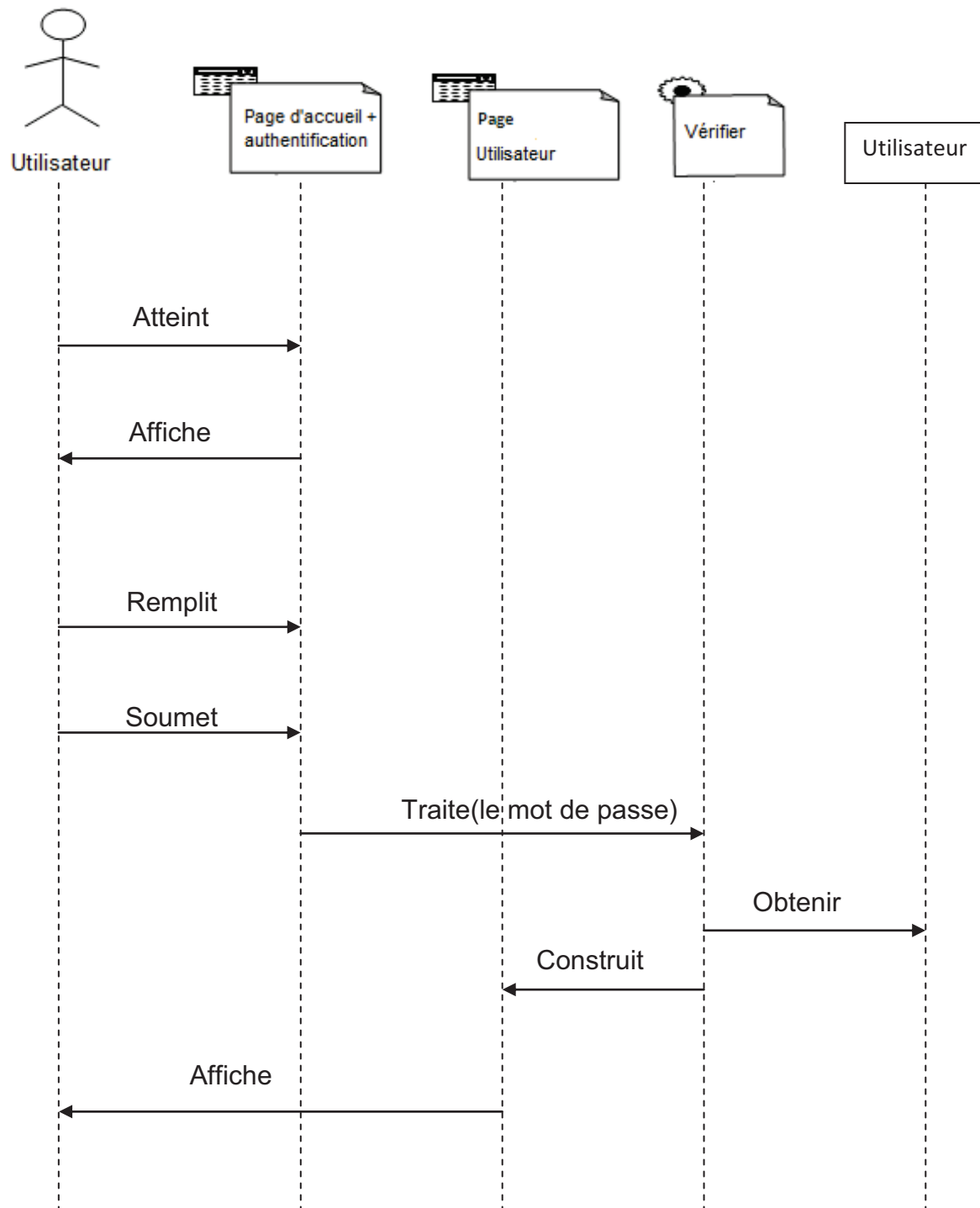


Fig.III.19 : Diagramme de séquence de réalisation des cas d'utilisation « Identification »

5.1.2. Diagramme de séquence de réalisation de Cas d'utilisation « Inscrire étudiant ».

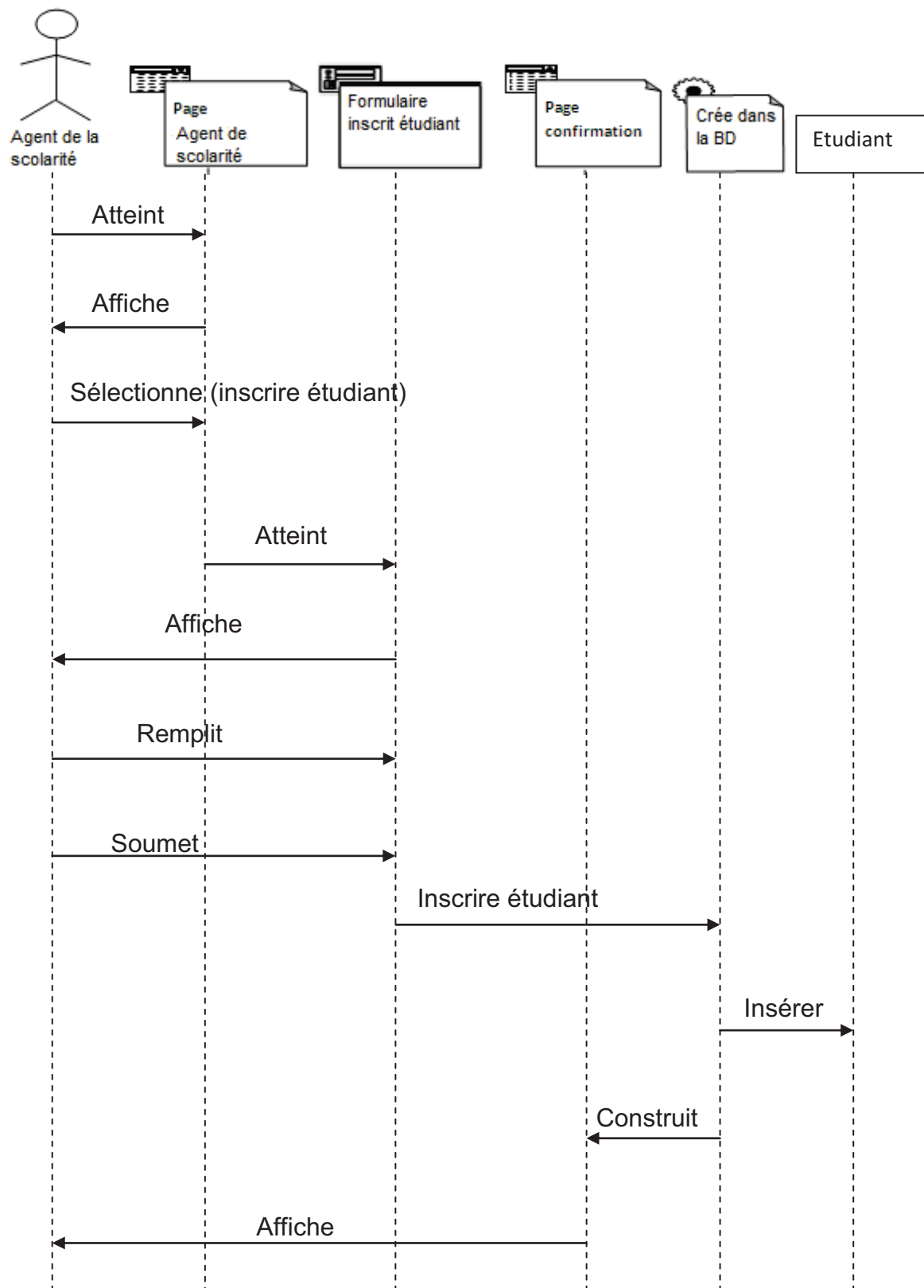


Fig.III.20 : Diagramme de séquence de réalisation de Cas d'utilisation « Inscrire étudiant »

5.1.3. Diagramme de séquence de réalisation cas d'utilisation « Ajouter formation ».

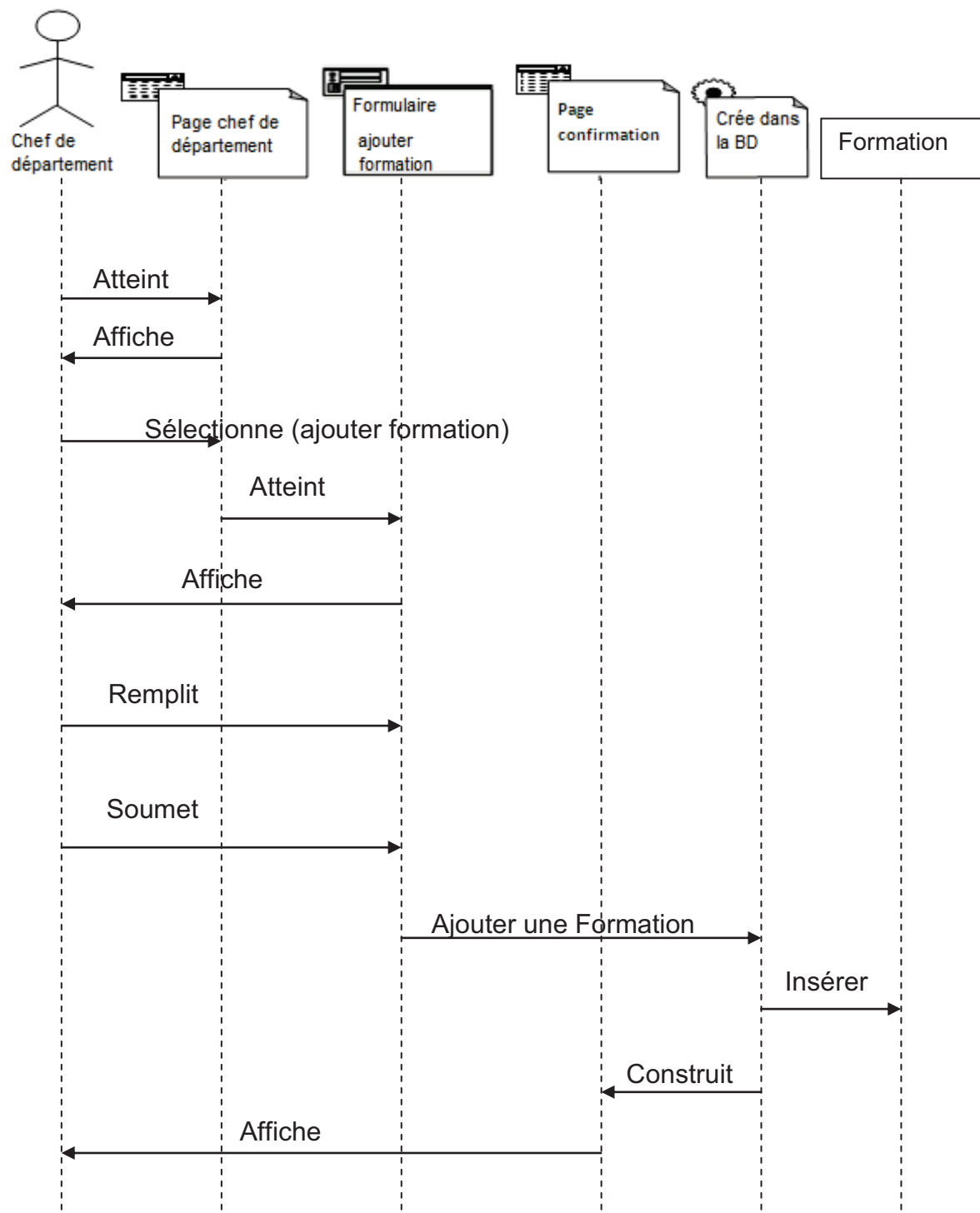


Fig.III.21 : Diagramme de séquence de réalisation cas d'utilisation « Ajouter formation »

5.1.4. Diagramme de séquence de réalisation de ca d'utilisation « rempli formulaire ».

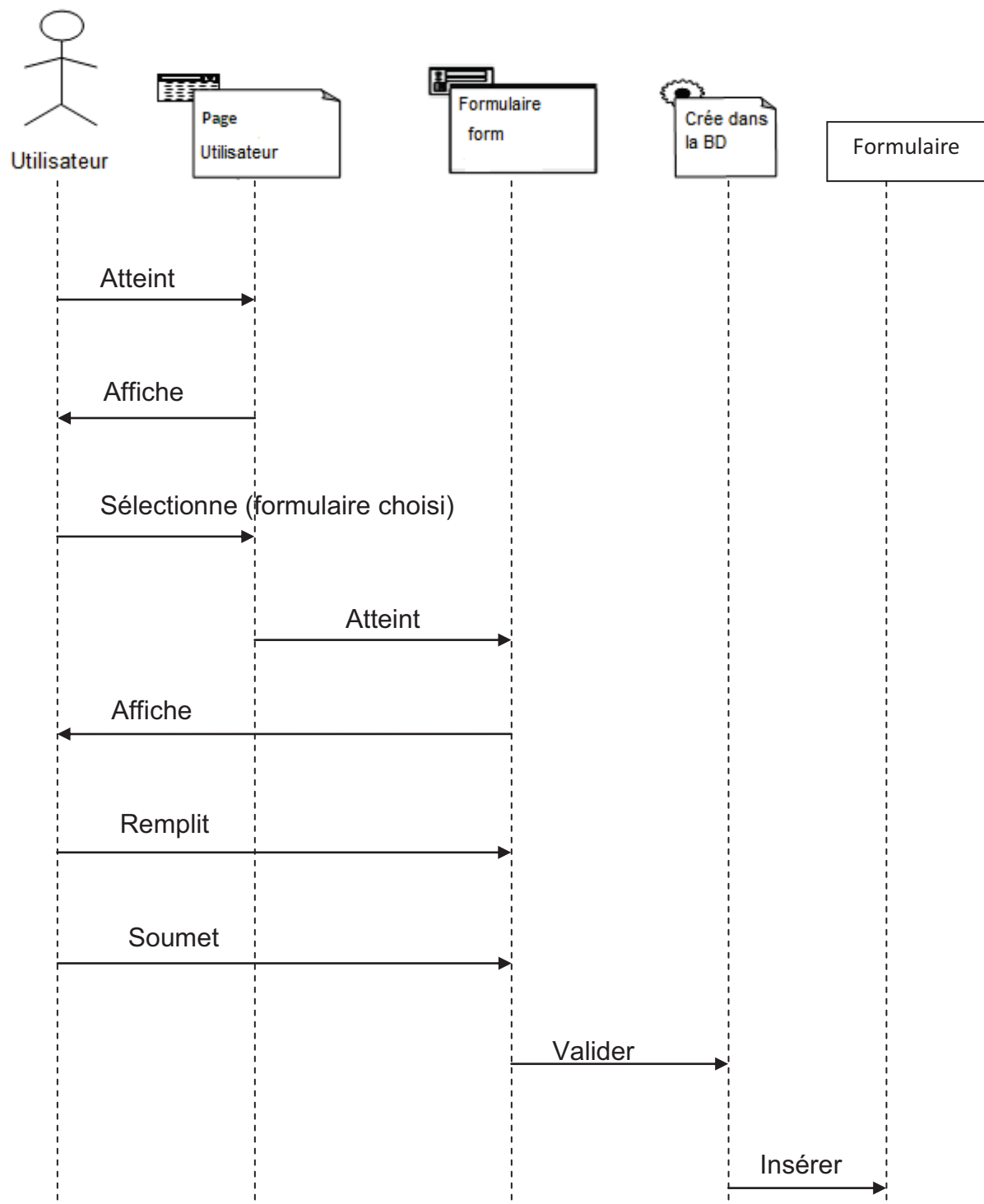


Fig.III.22 : Diagramme de séquence de réalisation de ca d'utilisation « rempli formulaire »

5.2. Les diagrammes de classes de l'application

Les diagrammes de classes expriment de manière générale la structure statique d'un système en termes de classes et de relations entre ces classes. De même qu'une classe décrit un ensemble d'objets, une association décrit un ensemble de liens ; les objets sont les instances de classes et les liens sont les instances des relations.

Un diagramme de classes n'exprime rien de particulier sur les liens d'un objet donné, mais décrit de manière abstraite les liens potentiels d'un objet vers d'autres objets c'est le diagramme pivot de l'ensemble de la modélisation d'un système, en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements.

Et voici les associations utilisées dans ces diagrammes :

- **Le lien « Link »**

C'est une association entre une page client et une autre page client ou serveur.

- **Construit « build »**

L'association « build » identifie quelques pages serveurs et elle est responsable de la création d'une page client.

- ✓ Une page serveur peut construire plusieurs pages client.
- ✓ Une page client ne peut être construite que par une page serveur.

- **Redirige « redirect »**

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

- **Le lien « submit »**

C'est un lien unidirectionnel d'une page client vers une page serveur.

Dans ce qui suit nous présentons quelques diagrammes de classes sur quelques cas d'utilisations.

5.2.1. Diagramme de classes général du cas d'utilisation «Identification»

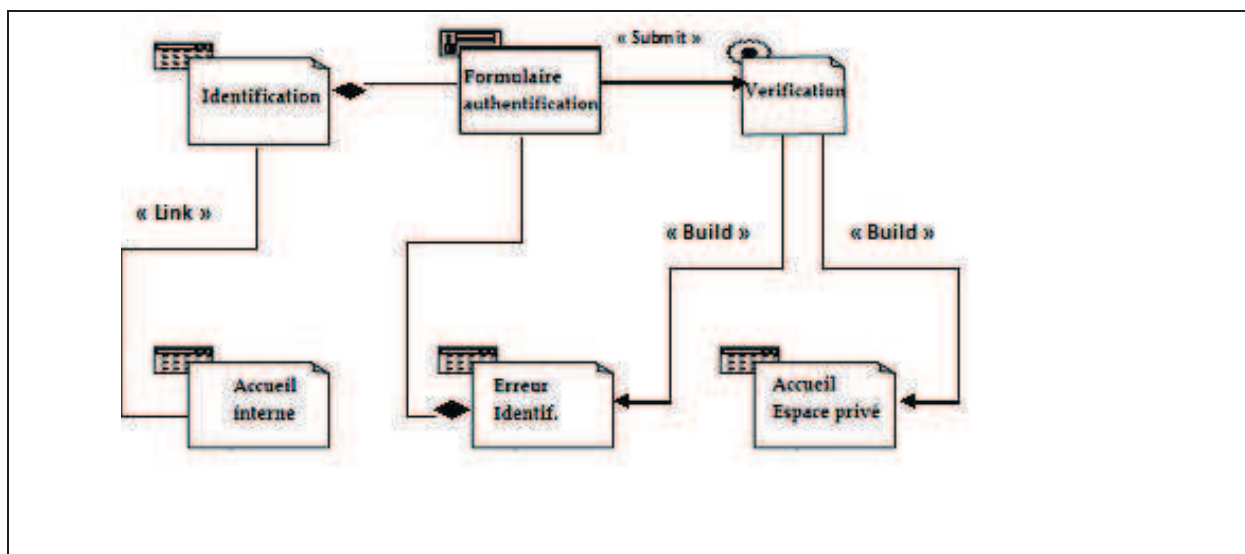


Fig.III.23 : Diagramme de classes général du cas d'utilisation «Identification»

5.2.2. Diagramme de classes général du cas d'utilisation «Inscrire étudiant»

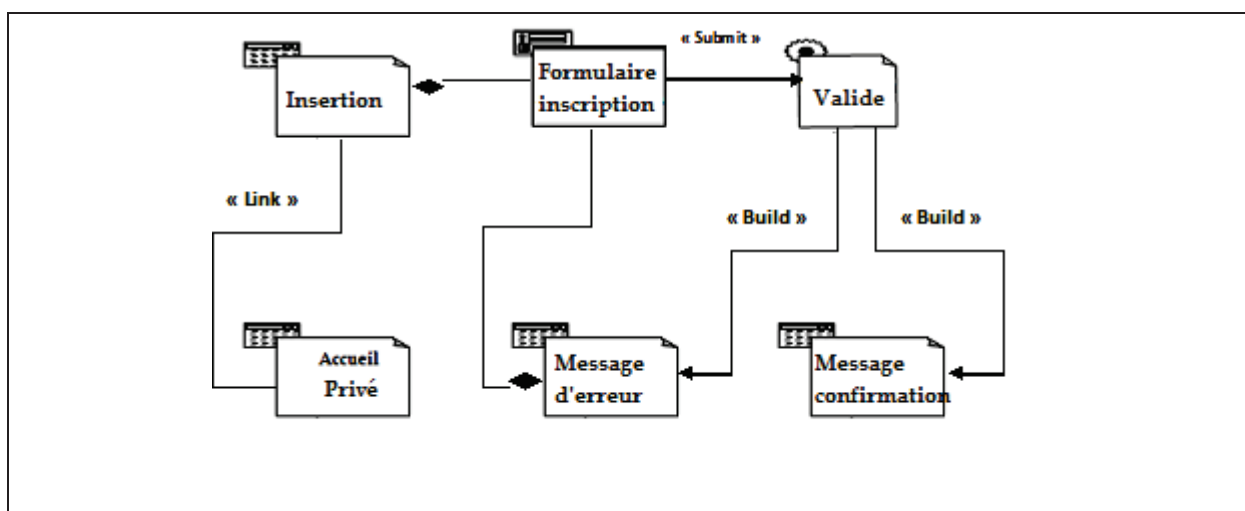


Fig.III.24 : Diagramme de classes général du cas d'utilisation «Inscrire étudiant»

5.2.3. Diagramme de classes général du cas d'utilisation «Ajouter UE/matières»

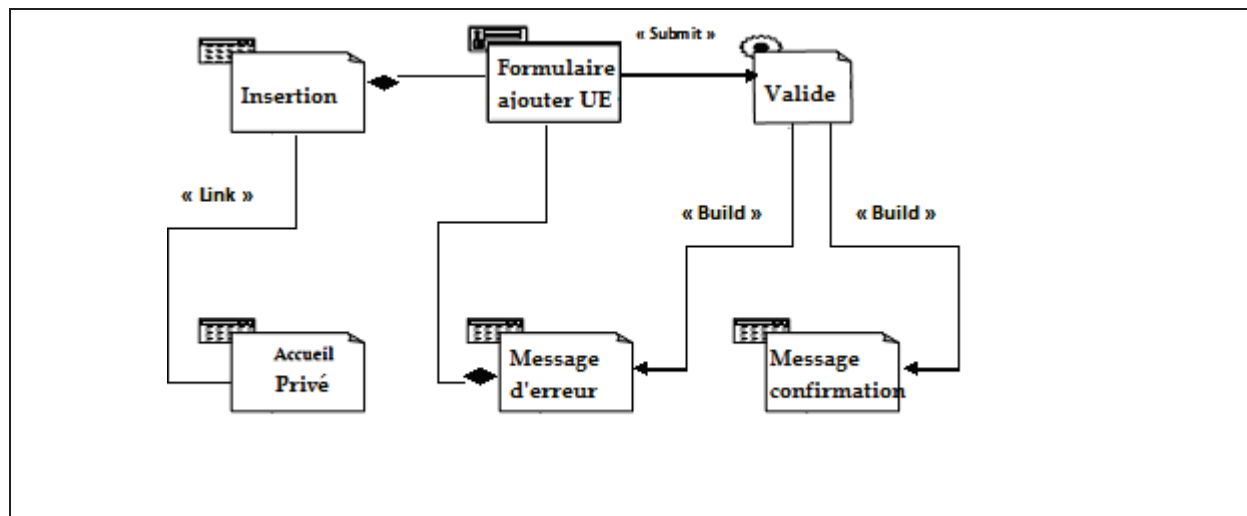


Fig.III.25 : Diagramme de classes général du cas d'utilisation «Ajouter UE/matières»

5.2.4. Diagramme de classes général du cas d'utilisation «Remplir formulaire»

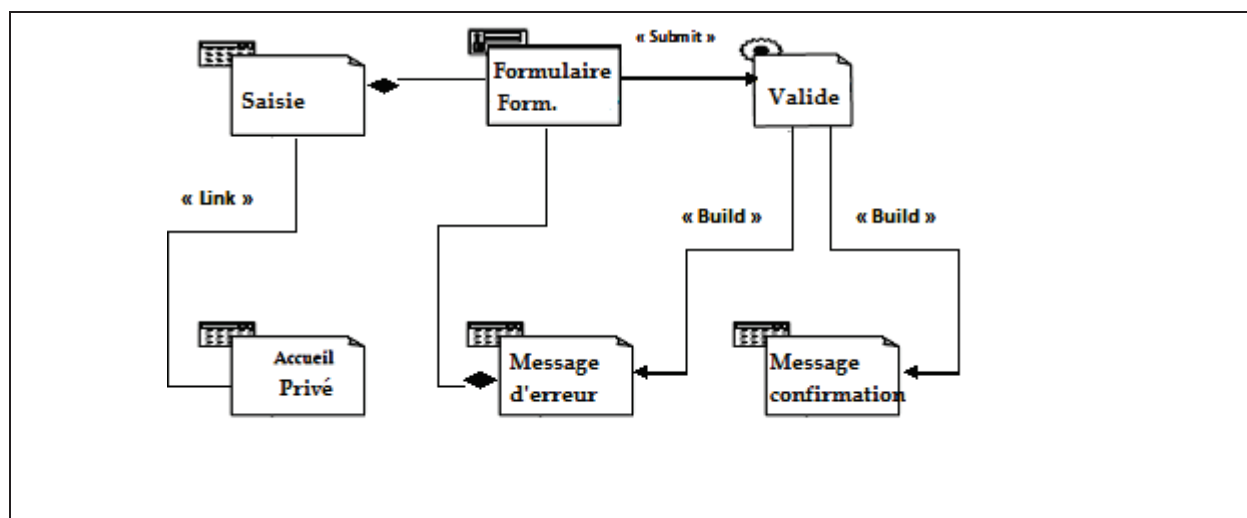


Fig.III.25 : Diagramme de classes général du cas d'utilisation «Remplir formulaire»

5.3. Diagramme de classes général

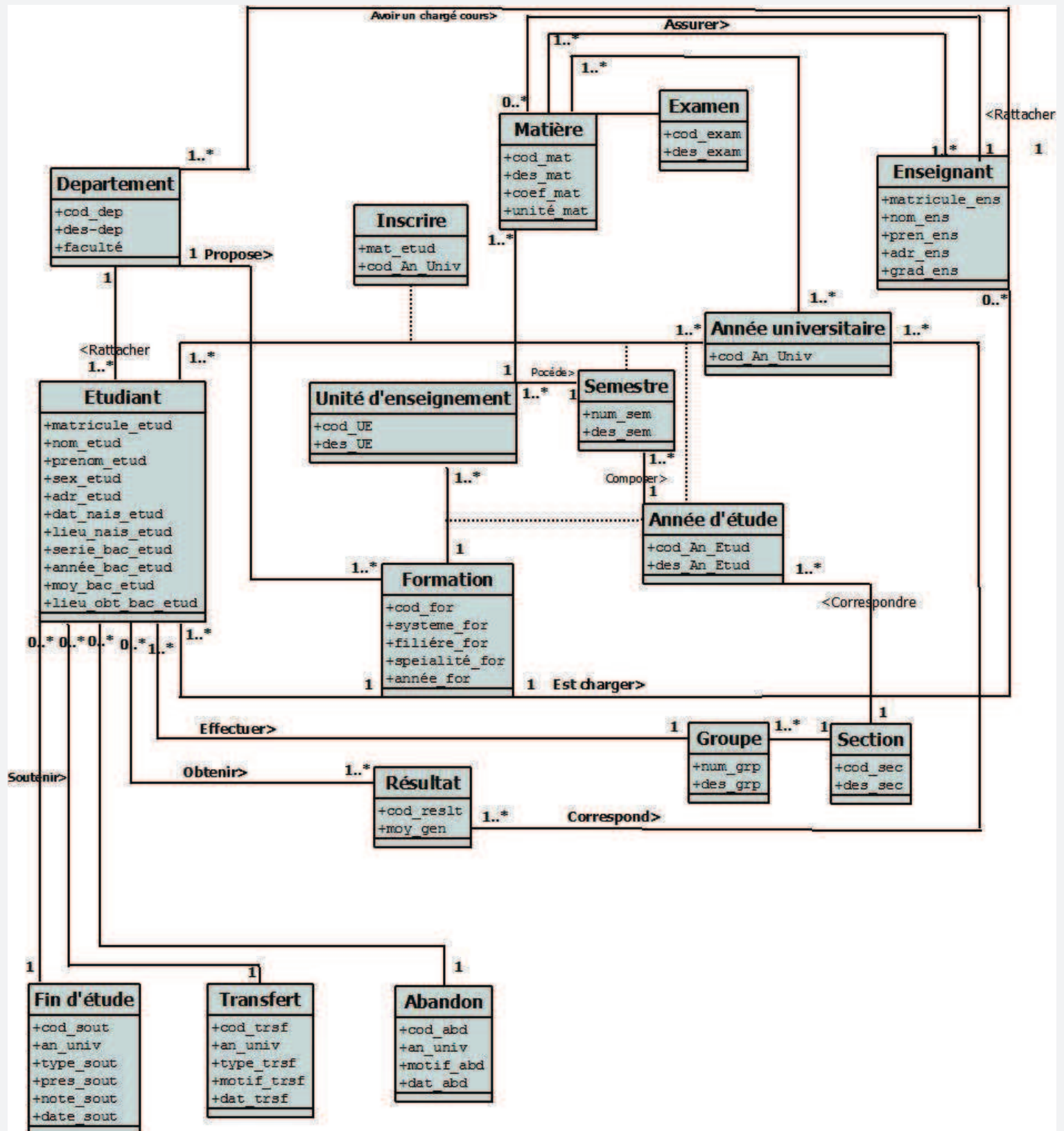


Fig.III.26 : Diagramme de classes général

6. Conception de la base de données**➤ Définition d'une base de données**

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

➤ Définition d'un SGBD

Un système de Gestion de Base de Données (SGBD) est un outil permettant d'insérer, de modifier et de recherche des données spécifiques dans une masse d'informations. C'est aussi une interface entre l'utilisateur et la mémoire secondaire permettant l'exploitation des informations stockées. Un SGBD est composé de trois couches : un système de gestion de fichier, un SGBD interne et un SGBD externe.

Il existe une multitude de types SGBD, ils se rejoignent sur un ensemble d'aspects fondamentaux. Parmi les SGBD les plus connus : Oracle, Microsoft SQL Server, MYSQL, Sybase SQL, IBM DB2.

➤ Modèle relationnel

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

❖ Organisation générale des données

Après la détermination des tables constitueront notre base de données, il nous faut définir les informations (champs) que contiendront chacune des tables.

- **Année_etud** (id_an_etud, lib_an_etud)
- **Année_univ** (id_an_univ, lib_an_univ)
- **Département** (id_dep, lib_dep)
- **Compte** (id_cpt, type_cpt, login_user, mdp_user, mat_user*, id_dep*)
- **Etudiant** (mat_etud, nom_etud, prenom_etud, sex_etud, adr_etud, dat_nais_etud, lieu_nais_etud, an_bac, moy_bac, wil_bac)
- **Etude** (mat_etud*, code_an_univ*, id_dep*, id_an_etud*, id_form*)
- **Enseignant** (mat_ens, nom_ens, prenom_ens, adr_ens, grade_ens, id_dep*)
- **Formation** (id_form, lib_form, spécialité, diplôme, mat_ens_resp*, id_dep*)
- **Unite_ens** (code_ue, lib_ue, credit_ue, sem_ue, vh_ue, id_form*, id_an_etud*)
- **Matiere** (id_mat, lib_mat, credit_mat, coef_mat, id_ue*)
- **Ens_Mat** (id_mat*, mat_ens*, id_an_univ*)
- **Section_groupe** (mat-etud*, id_an_univ*, sec, grp)
- **Resultat** (mat_etud*, id_mat*, note_emd1, note_emd2, note_tp, note_rat, note_prg, moy_mat)
- **Valider_PV_mod**(id_pv_mod, mat_mat, id_ens, id_form, id_an_univ, sem)
- **Valider_PV_sem**(id_pv, id_form, id_an_univ, sem)
- **Abandon** (mat_etud*, dat_ab, derniere_an_ins, id_an_univ*)
- **Transfers** (mat_etud*, id_an_univ*, type_trsf, univ_trsf, form_trsf, motif_trsf, dat_trsf)
- **Soutenance** (id_sout, mat_etud*, id_an_univ*, diplôme, note_sout, theme, dat_sout)
- **Absence** (mat_etud*, justif, motif, dat_abs_id_an_univ* , mat_ens*, id_mat*)

❖ **La structure complète de la base de données [III.2]**

Nous avons donc défini, de façon théorique la structure de notre base de données, voici le schéma général :

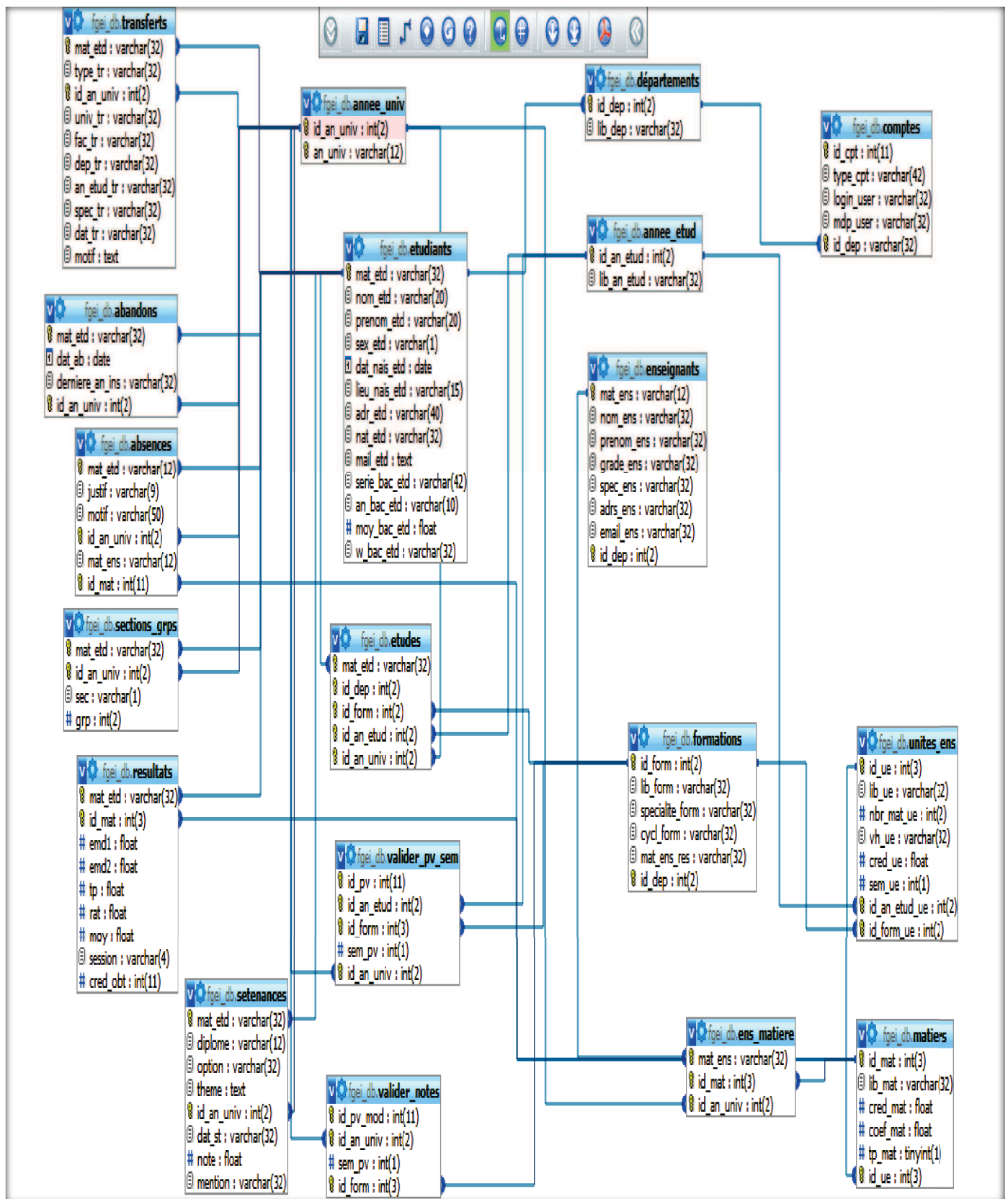


Fig.III. : Conception de bases de données.

Conclusion

Dans ce troisième chapitre, nous avons proposé une démarche de modélisation pour développer notre application, en se basant sur la méthode UML.

En premier lieu, nous avons défini les spécifications des besoins et les divers cas d'utilisations, puis la conception des diagrammes de séquences en phase d'analyse. Par la suite en phase de conception nous avons élaboré les diagrammes de classes spécifique puis générale.

Reste à définir l'environnement de développement de notre application Web, ce qui sera l'objet du chapitre suivant, qui est la réalisation et la mise en œuvre.

Chapitre IV

Réalisation et Mise en œuvre

Introduction

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

1. Architecture de l'application

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

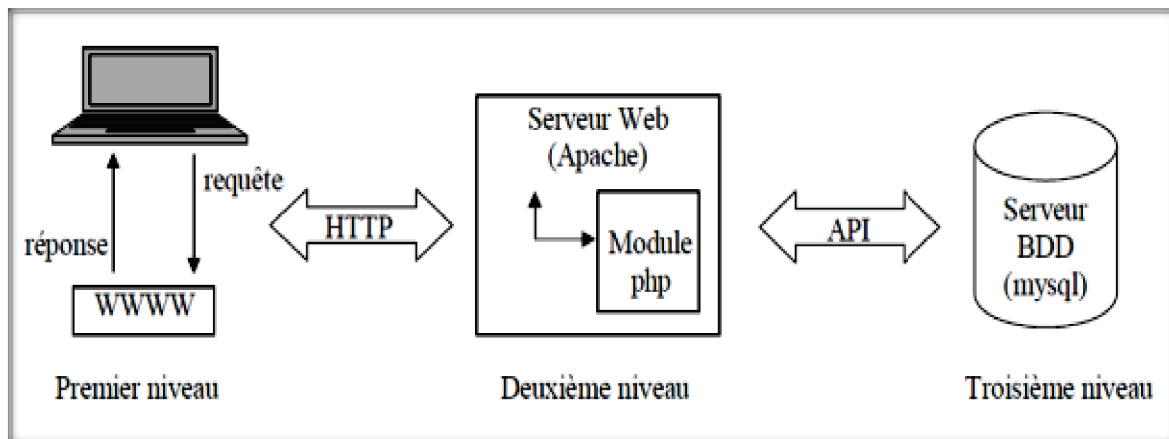


Fig. IV.1 : Architecture du déploiement de l'application [IV.1]

- **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.

2. Environnement de développement

L'implémentation du système d'information précédemment conçu a été réalisée sous le système d'exploitation Microsoft Windows 7. Cette implémentation a nécessité l'utilisation d'un ensemble d'outils ou package appelé WAMP, cet ensemble est décrit comme étant un environnement de développement contenant : un serveur http et un gestionnaire (serveur) de base de données ainsi qu'un éditeur de texte ; simples, gratuits et extrêmement efficaces.

Concrètement, le choix s'est porté sur l'environnement « XAMPP pour Windows » avec sa version 1.7.4 (janvier 2011), doté de la version d'Apache 2.1.13, MySQL

5.1.37, serveur ftp : File Zella, serveur de messagerie : Mercury. Il est à la fois simple d'utilisation, ergonomique et stable.

3. Outils de développement

3.1 Le serveur Web Apache

Le serveur Web Apache est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows.

Apache (prononcez à la française ou bien pour les puristes à l'anglophone « Apachy ») tire son nom de la façon dont il a été mis au point (« Apachy server » traduisez « un serveur rafistolé ») car il est le fruit d'une multitude de correctifs logiciels afin d'en faire une solution très sûre. En effet Apache est considéré comme sûr dans la mesure où peu de vulnérabilités le concernant sont publiées.

Tout développement de site web requiert un serveur Web qui s'occupe de traitement des requêtes des clients et l'exécution des programmes sur la machine serveur.

Notre méthode s'est basée sur Apache, du fait qu'il est le serveur Web le plus utilisé présentant un niveau de performances élevé pour des exigences matérielles modestes, en plus de sa garantie et sa robustesse.

3.2 Le serveur de base de données MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR) robuste et rapide. Une base de données permet de manipuler les informations de manière efficace, de les enregistrer, de les trier, de les lire et d'y effectuer des recherches. Le serveur MySQL contrôle l'accès aux données pour s'assurer que plusieurs utilisateurs peuvent se servir simultanément d'une même base de données pour y accéder rapidement et pour garantir que seuls les utilisateurs autorisés peuvent accéder aux données. MySQL est donc un serveur multiutilisateur et multithread. Il utilise SQL (*Structured Query Language*), le langage standard des requêtes de bases de données. MySQL est disponible depuis 1996, mais son développement remonte à 1979. Il s'agit de la base de données open-source la plus employée au monde.

Il est implémenté sur mode client serveur avec du côté serveur : le serveur MySQL, et du côté client : les différents programmes et librairies.

Mysql se caractérise par sa rapidité et sa facilité d'utilisation, il est offert avec l'outil d'administration de base de données « PhpMyadmin » par XAMPP.

4. Langages de programmation

❖ XHTML

Le langage utilisé pour la conception d'une page Web est le Hyper Text Markup Language. Il ne s'agit pas d'un langage de programmation au sens propre, mais d'un langage de description d'une page Web. Le fichier qui contient la description de cette page porte en général l'extension .htm ou .html. Il est constitué du texte et des liens aux images à afficher, répartis entre des balises (par exemple : <p>...</p>) qui déterminent la façon dont ces éléments seront présentés dans le navigateur. Certaines de ces balises permettent également de transformer un texte ou une image en lien hypertexte (<a>...). Ces hyperliens (les liens hypertextes) sont très importants dans une page Web, puisqu'ils permettent d'organiser la

navigation dans un site en reliant les pages entre elles. Les internautes peuvent passer d'une page à l'autre grâce à un simple clic sur ces liens, d'où l'expression « naviguer » ou « surfer » sur le Web.

❖ PHP

PHP est un langage de script côté serveur qui a été conçu spécifiquement pour le Web.

Le code PHP est inclus dans une page HTML et sera exécuté à chaque fois qu'un visiteur affichera la page. Le code PHP est interprété au niveau du serveur web et génère du code HTML ou toute autre donnée affichable dans le navigateur de l'utilisateur. PHP a été conçu en 1994 par **Rasmus Lerdorf**. Il a ensuite été adopté par d'autres personnes talentueuses et réécrit quatre fois avant de devenir le produit abouti que nous connaissons aujourd'hui.

PHP est un projet open-source, ce qui signifie qu'on peut se procurer son code, l'utiliser, le modifier et le redistribuer gratuitement.

PHP signifiait à l'origine *Personal Home Page*, mais ce nom a été changé en un acronyme récursif comme GNU (*Gnu's Not Unix*) : il signifie maintenant *PHP Hypertext Preprocessor*.

La dernière version principale de PHP est la version 5.

La page d'accueil de PHP est accessible à l'adresse <http://www.php.net>.

❖ SQL

C'est un langage de manipulation de base de 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 bases de données : sélection, modification et suppression d'enregistrements.
- La gestion des droits d'accès aux tables : contrôle des données et validation des Modifications.

❖ CSS

Les feuilles de styles simplifient le balisage HTML et soulagent grandement HTML des responsabilités de la représentation. Elles donnent le contrôle de la présentation des documents, informations sur les polices de caractères, alignement, couleurs, etc. L'objectif est de tendre vers la séparation du contenu et de la présentation.

Les informations de style peuvent être spécifiées pour un élément ponctuel ou pour des groupes d'éléments. Elles peuvent se trouver à l'intérieur du document HTML ou dans une feuille de style externe. Les mécanismes qui affectent une feuille de style à un document sont indépendants du langage de feuille de style.

Un fichier de feuille de style est un fichier dont l'extension est (.css). Dans ce fichier, il est possible de définir les propriétés de différentes manières. Si l'identifiant est une classe, il est précédé d'un point, si c'est un identifiant unique dans le fichier XHTML, il est précédé d'un dièse. Dans le cas des propriétés attribuées à une balise, il suffit d'utiliser le nom de la balise seul. Les propriétés de la balise sont encadrées par des accolades et définies par la syntaxe suivante :

Propriété : valeur ;

balise {
color : #FFFFFF;}

```
.classe {  
Color : #FFFFFF;}  
#id{  
Color :#FFFFFF;}
```

❖ JavaScript

Le JavaScript est un langage très largement employé sur Internet côté client, même s'il peut aussi fonctionner côté serveur. Il a été mis au point par Netscape Communications. Ses instructions sont incluses dans le code HTML des pages envoyées sur le poste client et sont traitées directement par le navigateur

A quoi sert le JavaScript ?

JavaScript est un langage qui permet d'exécuter des actions en fonction d'événements déclenchés par l'utilisateur (clic de Souris sur tel ou tel bouton par exemple). Il est ainsi possible d'ajouter des éléments dynamiques et interactifs à une page Web : pop-up, messages contextuels, affichage et disparition de menus. Ce langage permet de manipuler des objets, au sens informatique : créer des fenêtres spécifiques, contrôler les données saisies dans les formulaires, redimensionner certains objets, rediriger des liens...

A noter : c'est le navigateur Web qui se charge de l'exécution des petits bouts de programmes ou « scripts » écrits en JavaScript.

Dans ce qui suit, nous allons donner un bref aperçu sur l'ensemble des logiciels et langages cités au dessus :

5. Environnement logiciel

5.1



Fait partie du projet **Apache Friends**, créée en 2002. **XAMPP** est un ensemble de logiciels permettant de mettre en place facilement un serveur Web, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (**X** Apache **M**ySQL **P**erl **P**HP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus. Il est distribué avec différentes bibliothèques logicielles qui élargissent la palette des services de façon notable : OpenSSL, Expat (parseur XML), PNG, SQLite, zlib, ... ainsi que différents modules Perl et Tomcat. **[IV.2]**



Fig. IV.2: XAMPP for Windows

5.2. Macromedia Dreamweaver 8

Dreamweaver fut l'un des premiers éditeurs HTML de type WYSIWYG (What You see Is What You Get.), mais également l'un des premiers à intégrer un gestionnaire de site.

Il est destiné à la conception, au codage et au développement de sites, de pages et d'applications Web.

Quelque soit l'environnement de travail utilisé (codage manuel HTML ou environnement d'édition visuel), Dreamweaver propose des outils simples comparables à ceux qu'offre un logiciel de traitement de texte tel que Word (insertion de tableaux et d'images), est aussi bien utilisable par le néophyte que par le professionnelle.

Il offre trois modes de conception :

- **Mode création** : Permet à l'utilisateur d'effectuer la mise en page directement à l'aide des différents outils.
- **Mode code** : Afficher et modifier le code HTML, PHP ou autre (*c'est le mode qu'on a utilisé pour la réalisation de toutes les pages de notre appweb*).
- **Mode fractionné** : Afficher la page et son code en même temps.

6. Présentation des interfaces de l'application

Nous allons présenter dans ce qui suit quelques interfaces de notre application web, qui répondent aux critères ergonomiques cités précédemment afin de garantir une meilleure convergence entre notre système et les besoins de ses futurs utilisateurs.

Toute page de notre application se décompose en 03 zones principales :

- Une zone de navigation contenant les informations sur l'identité du l'utilisateur et de l'application.
- Une zone pour le menu, et les liens hypertextes.
- Une zone centrale affichant le contenu dynamique d'une page.

6.1. Page d'authentification (page d'accueil)

Page d'authentification des tous les utilisateurs de l'application (administrateur, chef de service, agents de scolarité, chefs de départements, enseignants, étudiants), c'est un formulaire simple clair et accessible, cette page est la page d'accueil de l'application, les utilisateurs devront s'authentifier pour accéder a leurs espaces privés, et seuls les étudiants de la FGEI peuvent créer leurs comptes à partir de cette page.

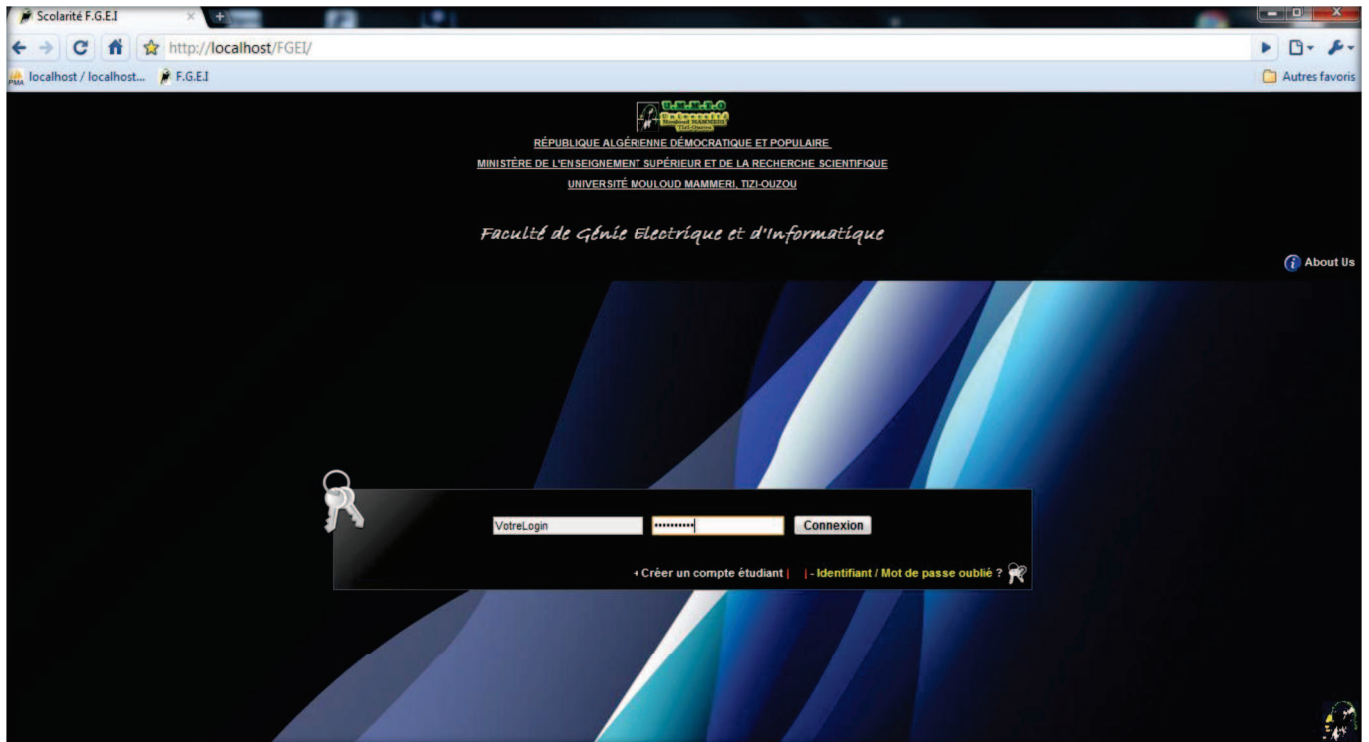


Fig. IV .3: Page d'authentification (page d'accueil)

6.2. Page d'accueil des utilisateurs après authentification (page espace privé)

Cette page peut être atteinte par tout acteur ayant un compte, dont chaque acteur peut accomplir ses taches.



Fig. IV .4: Page Espace Administrateur

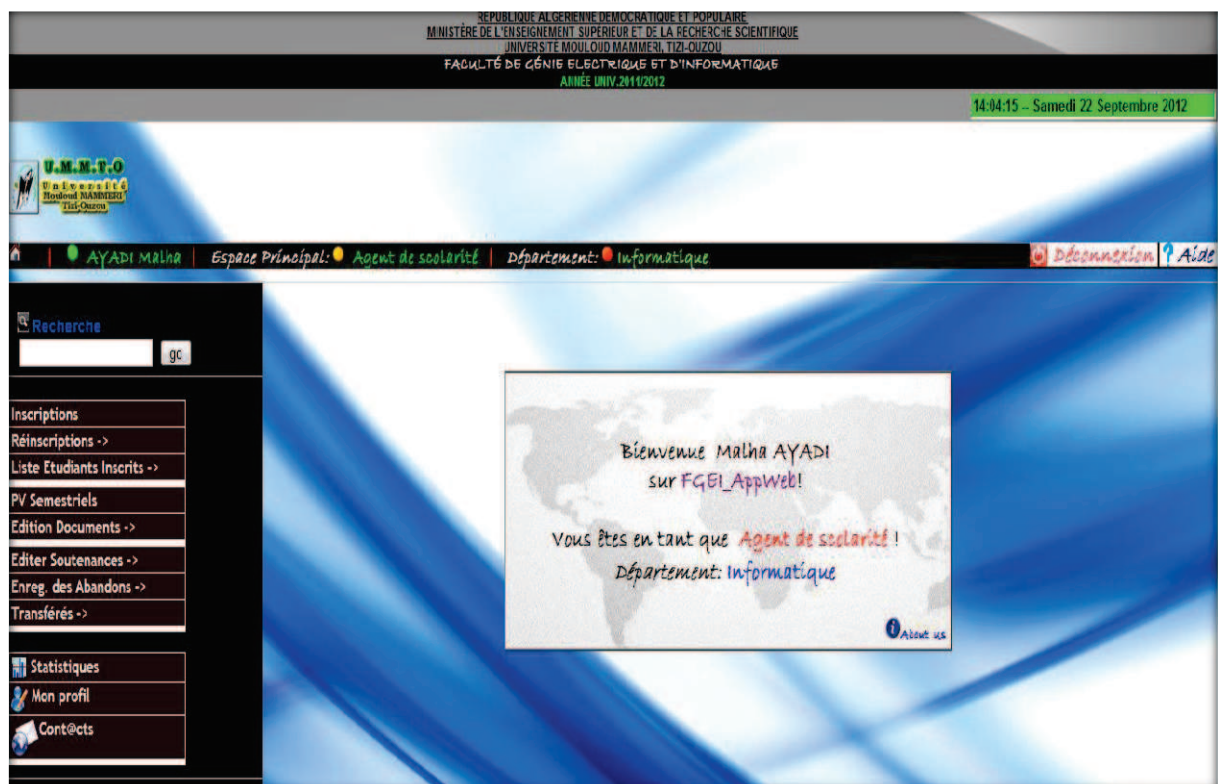


Fig. IV .5: Page Espace Agent de la scolarité

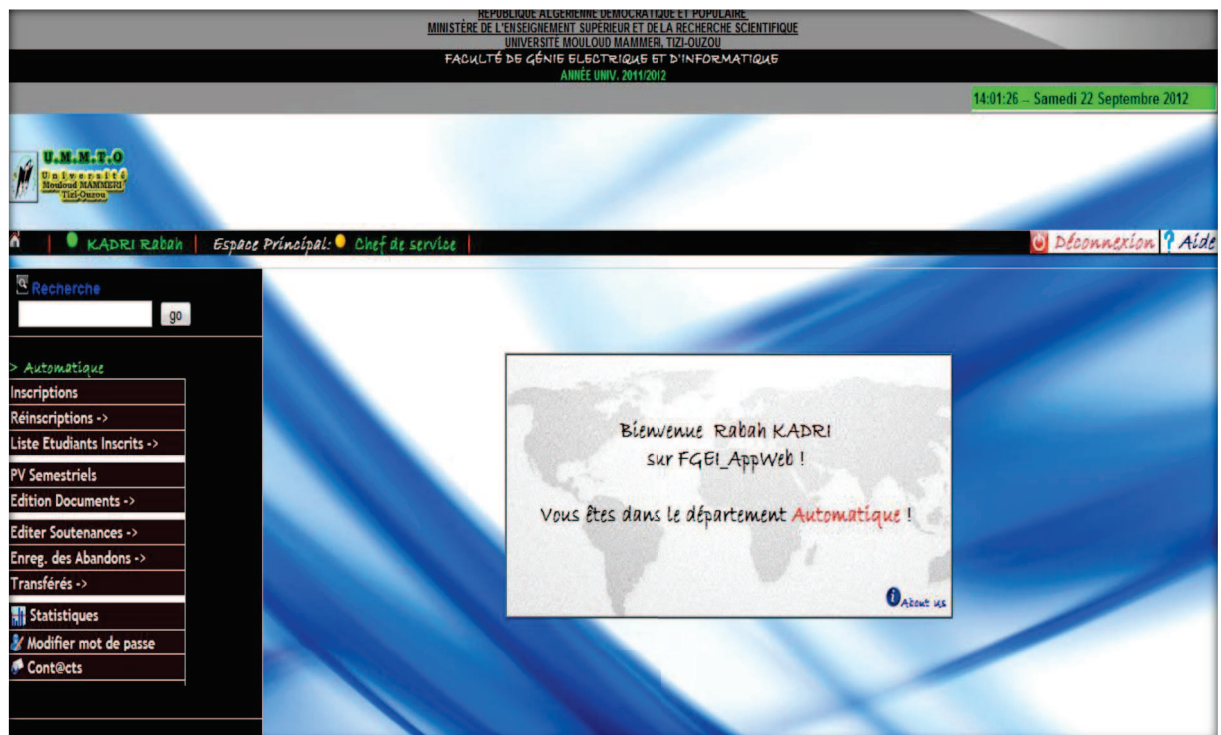


Fig. IV .6: Page Espace Chef de service de la scolarité



Fig. IV .7: Page Espace Chef de département



Fig. IV .8: Page Espace Enseignant

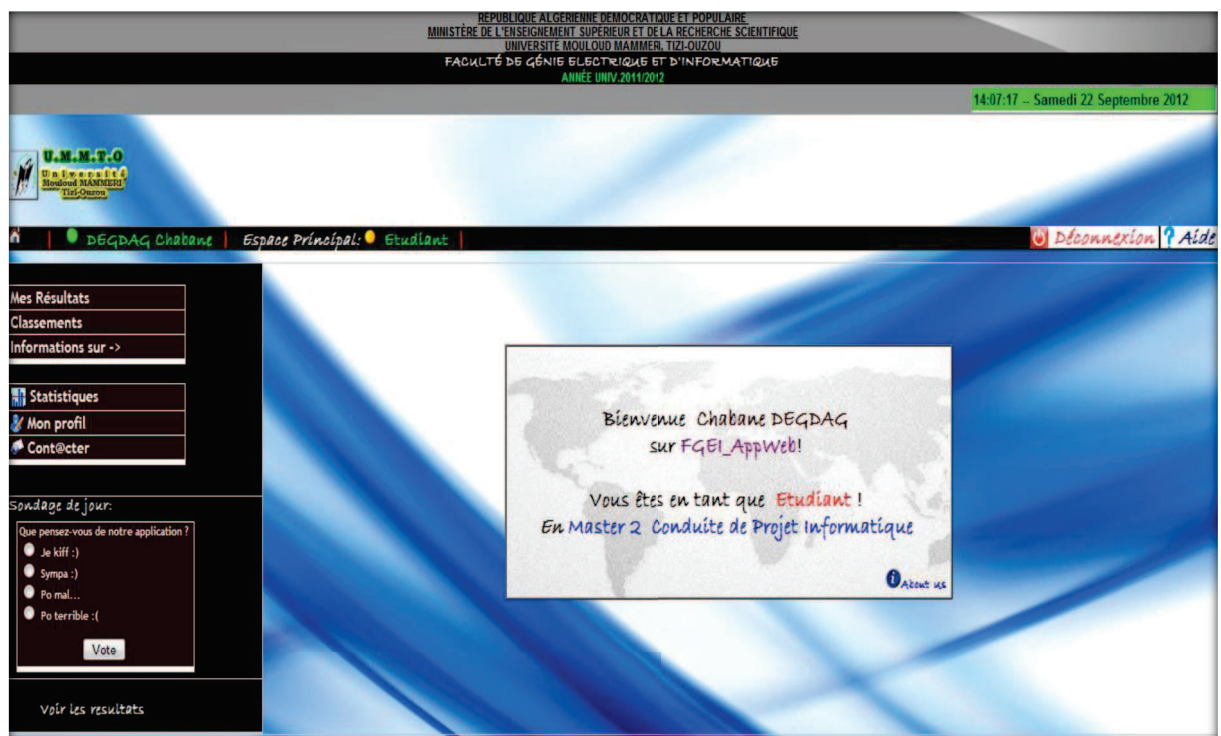


Fig. IV .9: Page Espace Etudiant

6.3. Page d'inscription des étudiants

http://localhost/FGEl/module_users/ges_etud/inscription_etudiants.php

Formulaire d'inscription des étudiants(es) en FGEl dépt. Automatique 2011/2012

Attribuer un Matricule Etudiant : 14D0307R14

Note: Le dernier matricule attribué est: 14D0307R13

Les infos. personnelle de l'étudiant(e):

Nom de famille :

Prenom :

Civilité : [Sélectionnez] ▼

Date de Naissance : [Jour] ▼ [Mois] ▼ [Année] ▼

Lieu de naissance : [Sélectionnez] ▼

Adresse personnelle complète:

Nationalité :

E-mail :

Les infos. concernant le BAC de l'étudiant(e):

Serie de BAC :

Année BAC : [Sélectionnez] ▼

Moyenne du BAC :

Wilaya d'obtention BAC : [Sélectionnez] ▼

Spécialité de la formation choisie :

Année d'etude:

6.4. Page d'édition des certificats de scolarité

Avant d'éditer un certificat de scolarité en PDF il faut entrer un matricule étudiant et choisir l'année universitaire.

- Etablir des Certificats de scolarité pour Les étudiants(es) /dép. Informatique

veuillez saisir

Matricule d'étudiant(e):

Année universitaire:

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

CERTIFICAT DE SCOLARITE

Le Recteur de l'Université Mouloud Mammeri de Tizi-Ouzou certifie que

l'étudiant(e): **DEGDAG Chabane**
Né(e) le: **1988-10-06**
à: **TIZI OUZOU**

est inscrit(e) à la faculté de Génie Electrique et d'informatique durant

l'année universitaire: **2011/2012**
Année d'études: **Master 2**
Filières: **Informatique**
Spécialité: **Conduite de Projet Informatique**

Fait à Tizi-Ouzou, le: **23 août, 2012**
P/ Le Recteur

6.5. Page d'édition des relèves de notes

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

RELEVÉ DE NOTES

Le chef de département, Informatique atteste que l'étudiant(e): Nom: **DEGDAG** Prénom: **Chabane** Né(e) le: **1988-10-06** à: **TIZI OUZOU**
Inscrit(e) en **Master 1 - Conduite de Projet Informatique** - a obtenu les résultats suivants durant l'année universitaire: **2010/2011**

SEMESTRE 1




Intitulés des unités d'enseignement et des matières constitutives	Coef.	Crdts.	MG /20	Session	Crdts.Acquis	VH
UED1 UE Decouverte 1		2				30
Séminaires à Thèmes Ouverts	2	2	9	1	0	
UEF9 UE fondamentales 9		16				600
Réseaux et NTIC	4	4	12	1	4	
Conception sur Systèmes d'exploitation	4	4	14	1	4	
Programmation Temps Réel	4	4	11	1	4	
Architectures Logicielles	4	4	12	1	4	
UEM5 UE méthodologique 5		6				100
Parcours Conduite de Projets Informatiques	6	6	13	1	6	
UET2 UE Transversale 2		6				60
Technique de Communication et d'Expression I	3	3	12	1	3	
Anglais III	3	3	12	1	3	


SEMESTRE 2

Intitulés des unités d'enseignement et des matières constitutives	Coef.	Crdts.	MG /20	Session	Crdts.Acquis	VH
UEF10 UE Fondamentale 10		25				600
Bases de Données Avancées	4	4	12	1	4	
Fondements de l'IA	5	5	12	1	5	
Architectures I	5	5	9	1	0	
Conduite de Projets Informatiques	6	6	11	1	6	
Compilation II	5	5	10	1	5	
UET3 UE Transversale 3		5				60

6.6. Page PV Modulaire


PV Modulaire Génie logiciel II
Master 2 Conduite de Projet Informatique Semestre 3
(2011/2012).

 Calculatrice |
  Version Imprimable (PDF) |
  Valider ce PV Modulaire

N°	Nom	EMD1	EMD2	TP	Rattrapage	Moyenne Module	Session	Credit Obtenu	
1	DEGDAG Chabane	12	12	/	/	12	1	5	 Modifier
2	ICHEDDADENE Aziz	13	12			12.5	Session 1	5	Enregistrer
3	GUECHOUD Saber						Session 1	5	Enregistrer
4	IRATNI Salim						Session 1	5	Enregistrer
5	CHELLI Nassim						Session 1	5	Enregistrer
6	BERCHICHE Hamza						Session 1	5	Enregistrer
7	BRAHAMI Saliha						Session 1	5	Enregistrer
8	KACI Brahim						Session 1	5	Enregistrer
9	KHELFA Amirouche						Session 1	5	Enregistrer

6.7. Page PV Semestriel

[PV Semestre 1 ✓](#) |
 [Pv Semestre 2 >](#) |
 [PV Global >>](#)

 Version Imprimable (PDF)

Pv Semestre 1 / Master 1 Conduite de Projet Informatique 2010/2011

No	Nom & Prenom	Moy.S1	Crds.Obt./30	S.	Rslt.	Relevés de Notes
1	DEGDAG Chabane	12.13	30	Session 1	Admi(e)	Releve de Notes
2	ICHEDDADENE Aziz	11.66	30	Session 1	Admi(e)	Releve de Notes
3	GUECHOUD Saber	11.46	30	Session 1	Admi(e)	Releve de Notes
4	IRATNI Salim	11.63	30	Session 1	Admi(e)	Releve de Notes
5	CHELLI Nassim	11.63	30	Session 1	Admi(e)	Releve de Notes
6	BERCHICHE Hamza	11.56	30	Session 1	Admi(e)	Releve de Notes
7	BRAHAMI Saliha	11.83	30	Session 1	Admi(e)	Releve de Notes
8	KACI Brahim	11.53	30	Session 1	Admi(e)	Releve de Notes
9	KHELFA Amirouche	11.46	30	Session 1	Admi(e)	Releve de Notes
10	OUAGUENOUNI Mokrane	13.33	30	Session 1	Admi(e)	Releve de Notes

6.8. Page résultats des étudiants

Les Résultats de DEQDAG Chabane
En Licence 3 Informatique Générale Durant 2009/2010

Résultats de Semestre 5 ✓ | Résultats de Semestre 6 >

Modules	EMD1	EMD2	TP	Rattrapage	Moyenne	Session	Credit Obt.
✓ Compilation I	12	12			12	1	6
✓ Réseaux	12	12			12	1	6
✓ Système Exploitation II	14	14			14	1	6
✓ Programmation Logique	12	12	12		12	1	4
✓ Technologies Internet	15	15			15	1	4
✓ Théorie des Graphes	13	13			13	1	4
Total des Crédit capitalisés	Moyenne Générale de semestre 5					Session	Résultat
30	12.93					Session 1	Admi(e)

6.9. Page UE et matières


Liste des unités d'enseignements et Matières constitutives / Master 2 Conduite de Projet Informatique

Semestre 3 ✓ | Semestre 4 >

Semestre 3				
Intitulés des unités d'enseignement et des matières constitutives	+			
UEF11 UE Fondamentale		+	/	X
✓ Systèmes d'Information II	/	X		
✓ Génie logiciel II	/	X		
✓ Modélisation et simulation	/	X		
UEF12 UE Fondamentale		+	/	X
✓ Sécurité Informatique II	/	X		
✓ Représentation des connaissances	/	X		
✓ Logique formelle	/	X		
UET4 UE Transversale		+	/	X
✓ Anglais V	/	X		

Notes: - Pour Editer des unités d'enseignements et des modules, Cliquer sur le lien correspondant
 - Pour Ajouter des unités d'enseignements et des modules, Cliquer sur l'icône +
 - Pour Modifier des unités d'enseignements et des modules, Cliquer sur l'icône /
 - Pour Supprimer des unités d'enseignements et des modules, Cliquer sur l'icône X

6.10. Page liste d'étudiants



Les étudiants Licence 2 Informatique Générale 2008/2009

	Section : A Groupe : 1
N°	Noms Etudiants(es)
1	DEGDAG Chabane
2	ICHEDDADENE Aziz
3	GUECHOUD Saber
4	IRATNI Salim
5	CHELLI Nassim
6	BERCHICHE Hamza
7	BRAHAMI Saliha
8	KACTI Brahim
9	KHELFA Amirouche
10	QUAGUENOUNI Mokrane

6.11. Page Modifier mot de passe

http://localhost/FGEL/divers/modif_profil.php?login=root

<< Annuler

Modification profil de DERRICHE Omar



Modifier Nom	<input type="text" value="DERRICHE"/>
Modifier Prenom	<input type="text" value="Omar"/>
Modifier E-mail	<input type="text" value="administrator@localhost.tn"/>
Modifier Login	<input type="text" value="root"/>
Ancien Mot de passe	<input type="password"/>
Nouveau Mot de passe	<input type="password"/>
Confirmer le nouveau mot de passe	<input type="password"/>

6.12. Page Gestion des utilisateurs

La Liste de tous les utilisateurs



 Actualiser la page |
  Ajouter un nouveau utilisateur |
  Version imprimable (PDF)

Trier Par:
 [Tous les utilisateurs](#) |
 [Agents de scolarité](#) |
 [Chefs de service](#) |
 [Chefs de département](#) |
 [Enseignants](#) |
 [Étudiants](#)

Cocher pour plus d'infos. sur les users:

Département ☐ - E-mail ☒

Matricule	Nom	Prenom	Login	Statut	E-mail	Modifier	Supprimer
ADMIN01	DERRICHE	Omar	root	Administrateur	administrator@localhost.to		
CS1	KADRI	Rabah	kadri	Chef de service	Kadri-rabah@localhost.to		
CDINFO1	HAMEG	S.	hameg	Chef département	Hameg-samir@localhost.to		
CDAUT1	BENSIDHOUM	Outahar	bensidhoum	Chef département	bensidhoum@localhost.to		
CDEL1	ZIANI	Rabah	ziani	Chef département	ziani_rabah@localhost.to		
CDETH1	KHELIFACUI	Mohend	khelifaoui	Chef département	khelifaoui@localhost.to		
AS01	AYADI	Malha	ayadi	Agent de scolarité	ayadi_malha@localhost.to		
AS02	LATROCUS	Omar	latrous	Agent de scolarité	latrous@localhost.to		
AS03	BOUDINE	Kahina	boudine	Agent de scolarité	boudine@localhost.to		
AS04	MAYOUF	Karim	mayouf	Agent de scolarité	mayouf_karim@localhost.to		
MEINF1	LALAM	Lalam	lalam	Enseignant	lalam@localhost.to		
MEINF2	AIT ELHADJ	F.	aitelhadj1	Enseignant	aitelhadj_f@localhost.to		
MEINF3	HAMITI	Hamiti	hamiti	Enseignant	hamiti@localhost.to		
MEINF5	ANALYSE	Numérique	an	Enseignant	a_n@localhost.to		
MEINF5	HAMADOUCHE	DJ.	proba	Enseignant	proba_dj@localhost.to		
MEINF6	KHEMLICHE	S.	khemliche	Enseignant	khemliche_salem@localhost.to		

Conclusion

Nous avons abordé dans ce dernier chapitre les différents outils de développement et les langages de programmations utilisés, ainsi que quelques fonctionnalités de notre application.

CONCLUSION GENERALE

Conclusion Générale

L'information joue un rôle important dans les entreprises et dans l'administration, son acheminement dans les plus brefs délais est une préoccupation majeure des responsables. L'implémentation d'une application informatique est l'un des moyens les plus efficaces qui permettent une grande fluidité de l'information dans l'entreprise.

Pour atteindre nos objectifs, nous avons touché à plusieurs domaines (Réseaux Informatique, Système d'information, Génie logiciel, Base de données, Développement Web, Sécurité Informatique, ...) qui nous ont permis d'acquérir des connaissances considérables durant toute la phase de réalisation de notre projet. Le produit final de notre projet d'étude c'est une application portable, sécurisée, fonctionnelle et paramétrable. En effet notre application peut être installée sur différentes plateformes (Windows, UNIX, LINUX, MAC), et elle est supportée par la plus part des navigateurs Web, elle est sécurisée grâce au système de session et le cryptage des mots de passe des utilisateurs. Notre application est fonctionnelle dans le cadre où elle offre les services de la gestion de la scolarité de la FGEI en temps réel et en grandes performances. Elle est paramétrable car elle est conçue d'une manière qu'elle soit opérationnelle pour toutes les facultés de l'UMMTO.

Nous espérons avoir atteint nos objectifs et que ce travail sera à l'origine d'un aboutissement à l'utilisation réelle de notre application au niveau de notre faculté.

Index bibliographique

- [(I.1)]** Microsoft Encarta 2009.
- [(I.2)]** PUJOLLE LES RESEAUX, EDITION 2008.
- [(I.3)]** BTS CGO 2 – Le client serveur cours F. Redonnet.
- [(I-4)]** Site Internet : <http://commentcamarche.com>

- [(II.1)]** Documents récupérés au niveau de rectorat de l'UMMTO.
- [(II.2)]** Site Internet : <http://www.usthb.dz/lmd.doc>
- [(II.3)]** Documents récupérés au niveau de FGEI.
- [(II.4)]** Mémoire d'ingénieur : conception et réalisation d'une application web de gestion de la scolarité (appliquée au système LMD) dans le cadre de la mise en place d'un intranet à la FGEI. Réalisé par DJAGHLOULI Nabila & ZERROUKI Djamila en 2007.

- [(III.1)]** Jim Conallen Concevoir des applications web avec UML
- [(III-2)]** XAMPP for windows: *phpMyAdmin*.

- [(IV.1)]** G. A. LEIERER, R. STOLL, "Le grand livre PHP 4 et MySQL". Micro application, 2000.
- [(IV.2)]** Site Internet : <http://www.apachefriends.org>

Annexe I

L'extension d'UML pour le web

L'extension d'UML pour le web

Un stéréotype : est une extension du vocabulaire d'UML, il permet d'associer une nouvelle signification à un élément du modèle. On le représente par une chaîne de caractères entre guillemets (« »).

Une étiquette : est une extension des propriétés d'un élément, elle permet la description d'une nouvelle propriété d'un élément du modèle. On la représente par une chaîne de caractères entre chevrons (<>).

Une contrainte : est une extension de la sémantique d'UML. Elle édicte une règle que le modèle doit vérifier pour être qualifié de « bien formé ». Elle est représentée par une chaîne de caractères entre accolades ({}).

II.1. Stéréotypes :

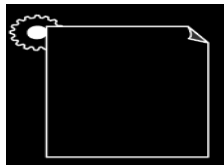
II.1.1. Une classe :

Page serveur « serveur page » :

Description : une page serveur représente une page qui possède des scripts exécutés par le serveur. Ces scripts interagissent avec des ressources serveur, telles que les bases de données.

Les opérations de l'objet représentent les fonctions dans le script et ses attributs, représentent les variables qui sont visibles dans la portée de la page.

Icône :



Contraintes : la page serveur ne peut avoir de relation qu'avec les objets sur serveur.

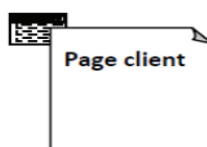
Étiquette : moteur de script pouvant être un langage ou le moteur qui doit être utilisé pour exécuter ou interpréter cette page.

Page client « client page » :

Description :

Une instance d'une page client est une page web formatée en HTML, un mélange de données, de représentation et même de logique. Les fonctions d'une page client correspondent aux fonctions des scripts de la page web, quant à ses attributs, ils correspondent aux variables déclarées dans les scripts et qui sont accessibles à toutes les fonctions et la page client. Les pages clients peuvent avoir des associations avec d'autres pages client ou serveur.

Icône :



Contraintes : aucune

Étiquette :

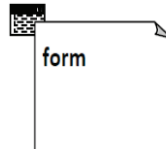
Titre : titre de la page tel qu'il est affiché dans le navigateur.

Base : URL de base pour référencer les URL relatives.

Corps : ensemble des attributs de la balise <body> qui définit les caractéristiques par du texte de l'arrière plan.

Formulaire «form» :**Description :**

Une classe stéréotype « form » est un ensemble de champ de saisie faisant partie d'une page client.

 Icône :

Étiquette : les méthodes Post ou Get pour soumettre les données à l'URL de l'attribut action de la balise <form>.

II.1.2. Une association :**Lien « link » :****Description :**

Un lien est un pointeur d'une page client vers autre page. Dans un diagramme de classe, un lien est une association entre une page client et une page visiteur. A un lien correspond une balise HTML ancre.

Icône : aucune.

Contrainte : aucune. **Étiquette :**

Paramètres : c'est la liste de noms de paramètres qui doivent être passés avec la demande de la page liée.

Lient cible « Target Link » :**Description :**

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

Icône : aucune.

Contraintes : aucune.

Étiquettes : en plus de la liste des paramètres, le nom de la cible ou la page vers laquelle pointe le lien doit être affiché.

Soumet «Submit» :**Description :**

Une association de soumission se trouve toujours entre un formulaire et une page serveur : les formulaires soumettent les valeurs de leurs champs au serveur pour qu'il les traite. Le serveur web traite la page serveur, qui accepte les informations de formulaire et les utilise.

Icône : aucune.

Contraintes : aucune.

Étiquette :

Paramètre : la liste de nom de paramètres qui doivent être passées avec la

demande de la page liée.

Construit « Build » :**Description :**

«Build » est une relation particulière faisant le pont entre les pages clients et les pages serveur.

L'association «build »indique quelle page serveur est responsable de la création d'une page client. C'est une relation orienté, puis que la page client n'a pas connaissance de la page serveur qu'il a construite.

Une page serveur peut construire plusieurs pages clients, en revanche une page client n'est construite que par une seule page serveur.

Icône : aucune.

Étiquettes : aucune.

Redirige «redirect » :**Description :**

Une relation de ce type est une relation unidirectionnelle avec une autre page web, pouvant être dirigée à partir d'une page client ou serveur vers une autre page client ou serveur.

Si la relation part d'une page serveur, le traitement de la requête de la page peut se poursuivre par l'autre page.

Icône : aucune.

Contraintes : Aucune.

Étiquette :

Délai : délai que doit observer la page client avant de se rediriger vers la page de destination.