

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE
SCIENTIFIQUE

Université Mouloud Mammeri Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique
Département d'Informatique



MEMOIRE

De fin d'études

En vue de l'obtention du diplôme

De Master en informatique.

**Automatisation du dispositif d'aide à l'insertion
professionnelle (AGSA)**

Cas : Agence Nationale de l'emploi (ANEM)

Proposé et dirigé par :

Mr. TALBI Said.

Réalisé par :

Mr. LAOUFI Mohamed Amine.

Mr. AICHOUR Amirouche.

Liste des Figures

Figure 1.1 Classification des réseaux informatiques	12
Figure 1.2 Topologie en bus	13
Figure 1.3 Topologie en anneau	13
Figure 1.4 Topologie en étoile.....	14
Figure 1.5 Classification selon l'étendue géographique	14
Tableau 1.1 Types de câbles et leurs débit	16
Figure 1.6 Modèle OSI.....	17
Figure 1.7 Processus d'encapsulation et de décapsulation	18
Tableau 1.2 Technologie Ethernet	20
Figure 1.8 Adressage IP	22
Tableau 1.3 Classes d'adressage IP	23
Figure 1.9 entête Ipv4 et Ipv6.....	23
Tableau 1.4 Quelques ports réservés	25
Figure 1.10 modèle TCP/IP.....	27
Figure 1.11 Architecture client / serveur.....	30
Figure 1.12 Architecture à 2 niveaux.....	31
Figure 1.13 Architecture à 3 niveaux.....	32
Figure 2.1 Organigramme d'une agence nationale de l'emploi.....	37
Tableau 2.1 Description des flux	38
Figure 2.2 Diagramme des flux	38
Tableau 2.2 liste des postes de travail.....	41
Tableau 2.3 codification de la liste des documents	43
Tableau 2.3.1 Feuille de presence.....	43
Tableau 2.3.2 Liste des candidats en formation	44
Tableau 2.3.3 Procès verbale d'ouverture d'une session	45
Tableau 2.3.4 Contrat de formation pour les jeunes âgés de 16 à 20 ans.....	46
Tableau 2.3.5 Carte bleue.....	47
Tableau 2.3.6 Fiche de pointage	48
Tableau 2.3.7 Lettre d'orientation	48
Tableau 2.4 liste des documents circulant au sein des postes de travail	49
Figure 3.1 Schéma de la démarche suivit.....	54
Figure 3.2. Diagramme de contexte.	55
Tableau 3.1 Liste des tâches et scénarios	57
Figure.3.2. Tableau représentant les différents cas d'utilisation	58
Figure.3.3. Diagramme de cas d'utilisation pour l'acteur Conseiller d'orientation.....	59
Figure.3.4. Diagramme de cas d'utilisation pour l'acteur Candidat	60
Figure. 3.5: Diagramme de séquence «l'authentification».....	62
Figure. 3.6 Diagramme de séquence «inscription d'un stagiaire»	63
Figure. 3.7 Diagramme de séquence «établissement d'un contrat».....	64
Figure. 3.8 Diagramme de séquence «mise à jour»	65
Figure 3.8 Diagramme d'activité du cas d'utilisation « authentification »	66
Figure 3.9 Diagramme d'activité du cas d'utilisation « inscription »	67
Figure 3.10 Diagramme d'activité du cas d'utilisation « Etablissement d'un contrat »	68
Figure 3.11 Diagramme d'activité du cas d'utilisation « Mise à jour »	69
Figure 3.12 Diagramme de classe du cas d'utilisation « authentification »	70
Figure 3.12 Diagramme d'activité du cas d'utilisation « Inscription »	71

Figure 3.13 Diagramme d'activité du cas d'utilisation « Etablissement d'un contrat »	72
Figure 3.14 Diagramme d'activité du cas d'utilisation « mise à jour »	73
Figure 3.15 Diagramme de classe globale.....	75
Tableau 4.1.1 Table candidat.....	90
Tableau 4.1.2 Table agence.....	90
Tableau 4.1.3 Table centre de formation.....	90
Tableau 4.1.4 Table Formation.....	91
Tableau 4.1.5 Table session	91
Tableau 4.1.6 Table avoir.....	91
Tableau 4.1.7 Table appartenir	91
Tableau 4.1.8 Table langue	91
Tableau 4.1.9 Table maitriser	92
Tableau 4.1.10 Table proposer	92
Tableau 4.1.11 Table accueil.....	92
Tableau 4.1.12 Table contrat.....	93
Tableau 4.1.13 Table candidat.....	93
Tableau 4.1.14 Table pointage	93
Tableau 4.1.15 Table daïra	94
Tableau 4.1.16 Table préinscription	94
Tableau 4.1.17 Table RDV	94
Figure 4.4 Interface d'accueil d' AGSA	95
Figure 4.5 interface Gestion des candidats.....	96
Figure 4.6 Interface Gestion des formations	97
Figure 4.7 Interface d'accueil de l'application web.....	98
Figure 4.8 interface d'inscription en ligne	99
Figure 4.9 Interface de l'espace candidat.....	100

Liste des Tableaux

Tableau 1.1 Types de câbles et leurs débit.....	16
Tableau 1.2 Technologie Ethernet.....	20
Tableau 1.3 Classes d'adressage IP	23
Tableau 1.4 Quelques ports réservés	25
Tableau 2.1 Description des flux	38
Tableau 2.2 liste des postes de travail.....	41
Tableau 2.3 codification de la liste des documents.....	43
Tableau 2.3.1 Feuille de presence.....	43
Tableau 2.3.2 Liste des candidats en formation	44
Tableau 2.3.3 Procès verbale d'ouverture d'une session	45
Tableau 2.3.4 Contrat de formation pour les jeunes âgés de 16 à 20 ans.....	46

Tableau 2.3.5 Carte bleue.....	47
Tableau 2.3.6 Fiche de pointage	48
Tableau 2.3.7 Lettre d'orientation	48
Tableau 2.4 liste des documents circulant au sein des postes de travail	49
Tableau 3.1 Liste des tâches et scénarios	57
Tableau 4.1.1 Table candidat.....	90
Tableau 4.1.2 Table agence	90
Tableau 4.1.3 Table centre de formation.....	90
Tableau 4.1.4 Table Formation.....	91
Tableau 4.1.5 Table session	91
Tableau 4.1.6 Table avoir.....	91
Tableau 4.1.7 Table appartenir	91
Tableau 4.1.8 Table langue	91
Tableau 4.1.9 Table maitriser	92
Tableau 4.1.10 Table proposer	92
Tableau 4.1.11 Table accueil.....	92
Tableau 4.1.12 Table contrat	93
Tableau 4.1.13 Table candidat.....	93
Tableau 4.1.14 Table pointage	93
Tableau 4.1.15 Table daïra	94
Tableau 4.1.16 Table préinscription	94
Tableau 4.1.17 Table RDV	94

Sommaire

Liste des Figures	2
Liste des Tableaux	3
Introduction Générale	8
Chapitre 1 : Réseaux Informatiques	10
1.1 Introduction.....	11
1.2 Taxinomie des réseaux informatique.....	11
1.2.1 Classification selon la topologie.....	12
1.2.2 Classification selon l'étendue géographique	14
1.2.3 Classification selon le média de transmission	15
1.3 Couches protocolaires du modèle OSI	16
1.3.1 Couche physique	18
1.3.2 Couche liaison de données	19
1.3.3 Couche réseau	21
1.3.4 Couche transport.....	23
1.3.5 Couche session	25
1.3.6 Couche présentation	26
1.3.7 Couche application	26
1.4 Couches protocolaires du modèle TCP/IP	26
1.4.1 Couche accès réseau	27
1.4.2 Couche internet	27
1.4.3 Couche transport.....	27
1.4.4 Couche application	28
1.5 Modèle OSI versus Modèle TCP/IP	29
1.6 Architecture client/serveur.....	29
1.6.2 Serveur.....	31
1.6.3 Niveaux d'architecture client/serveur	31
1.7 Web	32
1.7.1 Présentation.....	32
1.7.2 Page web	33
1.7.3 Site web	33
1.7.4 Site web statique.....	33
1.7.4 Applications web	33
1.8 Conclusion	33

Chapitre 2 : Etude préalable	34
2.1 Introduction.....	35
2.2 Organisme d'accueil	35
2.2.1 Présentation de l'agence nationale de l'emploi Tizi-Ouzou.....	35
2.2.2 Missions de l'agence nationale de l'emploi Tizi-Ouzou.....	35
2.2.4 Organisation de l'Agence Nationale de l'Emploi Tizi-Ouzou	36
2.2.5 Dispositif d'aide à l'insertion des jeunes demandeurs d'emploi.....	37
2.3 Champ d'étude.....	37
2.3.1 Diagramme des flux.....	37
2.3.2 Service demandeur d'emploi	39
2.3.3 Service dispositif d'aide à l'insertion professionnelle	39
2.3.4 Problématiques et expression des besoins	39
2.4 Analyse préliminaire	40
2.4.1 Etude des postes de travail.....	40
2.4.2 Etude des documents	42
2.4.3 Circulations des documents au sein des postes de travail	49
2.4.5 Architecture réseau proposée	49
2.5 Conclusion	49
Chapitre 3 : Analyse et Conception.....	50
3.1 Introduction.....	51
3.2 Unified Modeling Language UML	51
3.2.1 Présentation.....	51
3.2.2 Diagrammes structurels	51
3.2.3 Diagrammes comportementaux.....	52
3.2.4 Diagrammes d'interaction	53
3.3 Démarche suivie.....	53
3.3.1 Expression des besoins	54
3.3.2 Identification des acteurs	54
3.3.3 Choix des diagrammes UML	54
3.3.4 Analyse	55
3.3.5 Conception	55
3.4 Analyse.....	55
3.4.1 Diagramme de contexte	55
3.4.2 Tâches et scénarios	56
3.4.2 Diagramme des cas d'utilisation pour les tâches	57
3.5 Conception.....	61

3.5.1 Diagrammes de séquence.....	61
3.5.2 Diagramme d'activité.....	66
3.5.3 Diagrammes de classe.....	70
3.5.4 Diagramme classe global	74
3.5.5 Modèle logique de données.....	76
3.6 Conclusion	77
Chapitre 4 : Mise en œuvre.....	78
4.1 Introduction.....	79
4.2 Quelques Langages de programmation	79
4.2.1 Python.....	79
4.2.2 Java.....	79
4.2.3 C++.....	80
4.2.4 Delphi	80
4.3 Quelques outils de développement web	81
4.3.1 Eclipse	81
4.3.2 Netbeans	82
4.3.3 Dreamweaver	82
4.3.5 Choix d'outils de développement web	82
4.4 JSP.....	83
4.4.1 Serveur Web Apache tomcat.....	85
4.5 Quelques système de gestion de bases de données	85
4.5.1 Oracle DATABASE	85
4.5.2 Firebird	86
4.5.3 MySQL	86
4.5.4 PostgreSQL	86
4.5.5 Microsoft SQL Server.....	86
3.5.6 Langages d'interrogation de bases de données utilisés.....	88
4.6 Modèle physique de données	89
4.6.1 Candidat	89
4.6.2 Agence	90
4.6.3 Centre de formation	90
4.6.4 Formation.....	90
4.6.5 Session	91
4.6.6 Avoir.....	91
4.6.7 Appartenir	91
4.6.8 Langue	91

4.6.9 Maitriser	92
4.6.10 Proposer	92
4.6.11 Organisme accueil	92
4.6.12 Contrat	92
4.6.13 Etat.....	93
4.6.14 Pointage	93
4.6.15 Daira	94
4.6.16 Préinscription	94
4.6.17 RDV.....	94
4.7 Quelques interfaces de l'AGSA	95
4.7.1 Interface D'accueil	95
4.7.2 Interface Gestion des candidats	96
4.7.3 Interface gestion des formations.....	97
4.8 Quelques interfaces de l'application Web	98
4.8.4 Interface Accueil	98
4.8.5 Interface l'inscription en ligne	99
4.8.6 Interface espace candidat	100
4.9 Conclusion	100
Conclusion générale	101
Références	102

Introduction Générale

Toute gestion responsable d'une organisation suppose une maîtrise des informations relatives à son fonctionnement en vue de prendre des décisions qui s'imposent pour la bonne marche de cette première.

En effet, gérer une organisation sous-entend le fait d'avoir la capacité d'exécuter ou de faire exécuter en temps opportun le traitement d'un ensemble des tâches très souvent complexes et répétitives en vue d'aboutir à des résultats satisfaisants. Ce traitement, s'il est fait seulement par l'homme, a de nombreuses chances de donner lieu à des résultats incohérents et même biaisés du fait de l'erreur humaine.

Pour pallier à ces sérieuses difficultés propres à l'homme : l'erreur et la faute, il arrive que l'on fasse recours à des machines automatiques programmables. Celles-ci ont l'avantage de mémoriser un ensemble d'instructions conçues une seule fois par l'homme et vérifiées par lui pour se rassurer de leur exactitude, et de les exécuter chaque fois que le besoin se ferait sentir en tenant compte des données actualisées mises à sa disposition. C'est bien là la noble vocation de l'ère informatique.

Par ailleurs, l'informatique ayant envahi tous les domaines de la vie, l'on ne sait pas de nos jours, passer outre ses outils quoique encore effrayant dans quelques rares cas des allergiques chroniques et / ou des nostalgiques des anciennes pratiques manuelles.

En outre, l'avènement des autoroutes d'informations a rendu quasi indispensable l'usage de l'outil informatique de pointe pour une gestion moderne, fiable et rapide de toutes les informations, condition sine qua non d'une gestion efficace des organisations qui se veulent modernes.

Enfin, nous pensons que la gestion d'une agence nationale de l'emploi nécessite un traitement par des machines automatiques de ses nombreuses informations en vue d'un meilleur rendement et donc d'une bonne gestion.

C'est dans cette optique que le concessionnaire l'Agence National de l'emploi envisage de s'investir afin d'exploiter les biens de la technologie en son profit. Pour cela, un travail nous a été confié, consistant en une conception et réalisation d'une application pour gérer les demandes

d'emploi des personnes âgés de 16 à 20 ans dans le cadre du dispositif d'aide à l'insertion professionnel , cette application aura un double objectif : d'une part permettre aux demandeurs où qu'ils soient de consulter les formations disponibles, préinscrire, de prendre un rendez-vous pour une inscription et maintenance du contact avec l'agence . Et d'une autre part permettre aux conseillers d'orientation une gestion et un suivi efficace des dossiers des demandeurs.

Nous avons nommés notre application « AGSA » qui est l'acronyme de : « Application de Gestion des Stagiaire ANEM ».

Pour tenter de répondre à ces objectifs nous avons scindé ce mémoire d'étude en quatre chapitres :

- **Chapitre1:** Présente les généralités sur le web et les réseaux informatiques.
- **Le chapitre2 :** Intitulé Etudes préalable qui présente l'Agence National de l'emploi de Tizi-Ouzou.
- **Le chapitre3 :** Est consacrée à l'analyse et la conception de l'application.
- **Le chapitre4 :** Nous exposerons sous forme d'un résumé le choix de l'environnement de développement utilisé ainsi que les différents outils.

Chapitre 1 : Réseaux Informatiques

1.1 Introduction

A l'origine, un réseau était un rassemblement de personnes ou d'objets. De nos jours on entend par réseau, les réseaux d'entreprises, qui connectent différentes machines afin de pouvoir les faire communiquer entre elles. Que ce soit pour le partage de fichiers ou l'envoi de messages, la plupart des entreprises sont aujourd'hui dotées d'un réseau afin d'être plus efficaces (il est quand même plus simple de transférer un fichier par Internet que de l'envoyer sur CD par la poste).

Dans ce chapitre nous allons étudier les différentes méthodes de classer les réseaux informatiques, qui sont classées selon la topologie, l'étendue géographique, le média de transmission.

1.2 Taxinomie des réseaux informatique

Nous pouvons classer les réseaux selon la topologie, l'étendue géographique, le média de transmission, comme nous le montrons dans la figure ci-dessous.

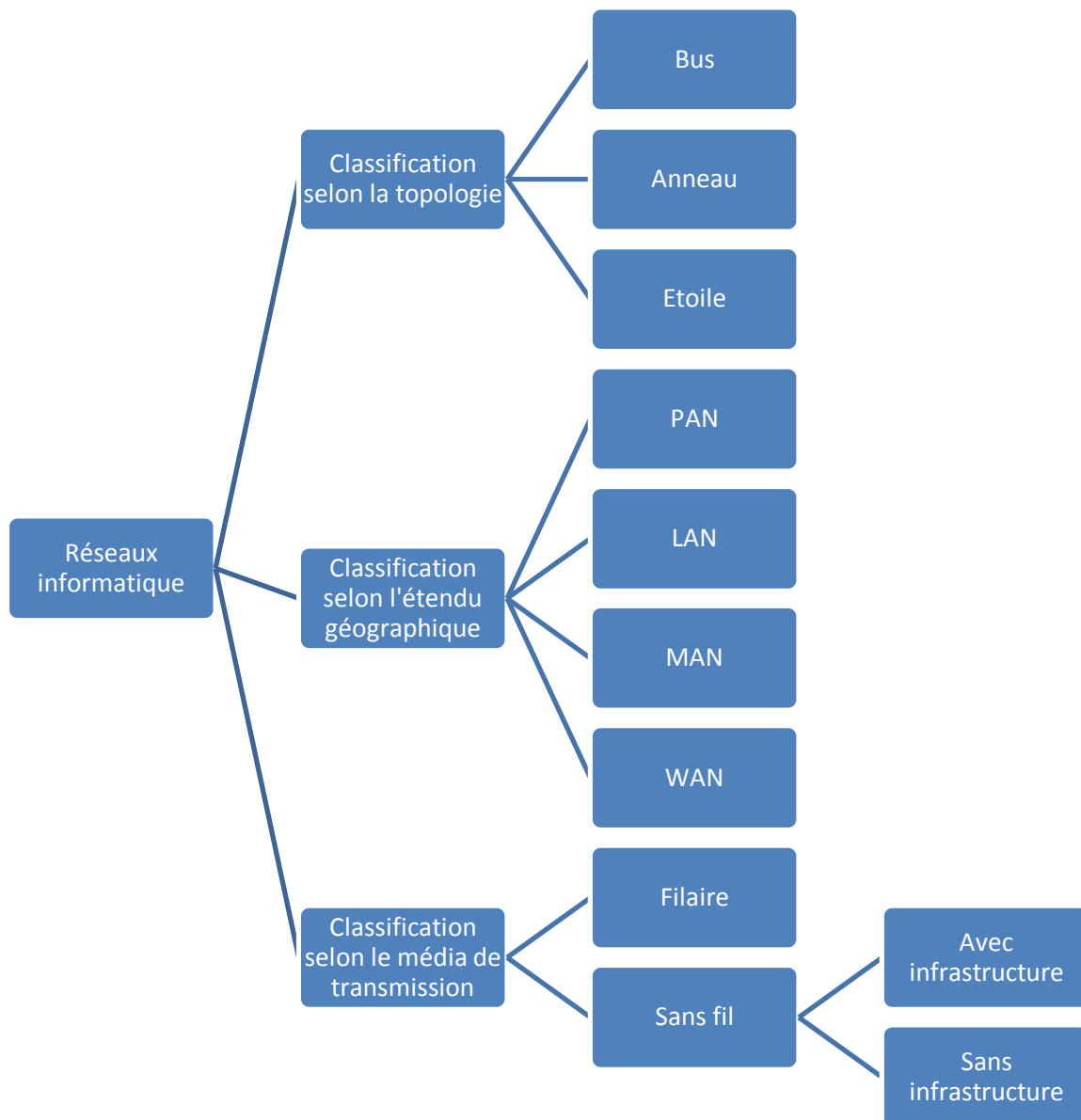


Figure 1.1 Classification des réseaux informatiques

1.2.1 Classification selon la topologie

Un réseau informatique est constitué d'ordinateurs reliés entre eux grâce au matériel (câblage, cartes réseau, ainsi que d'autres équipements permettant d'assurer la bonne circulation des données). L'arrangement physique de ces éléments est appelé topologie physique

Topologie en bus

Dans une topologie en bus tous les ordinateurs sont reliés à une même ligne de transmission par l'intermédiaire de câble, généralement coaxial. Le mot "bus" désigne la ligne physique qui relie les machines du réseau.

Cette topologie a pour avantages d'être facile à mettre en œuvre et de fonctionner facilement, par contre elle est extrêmement vulnérable étant donné que si l'une des connexions est défectueuse, c'est l'ensemble du réseau qui est affecté.

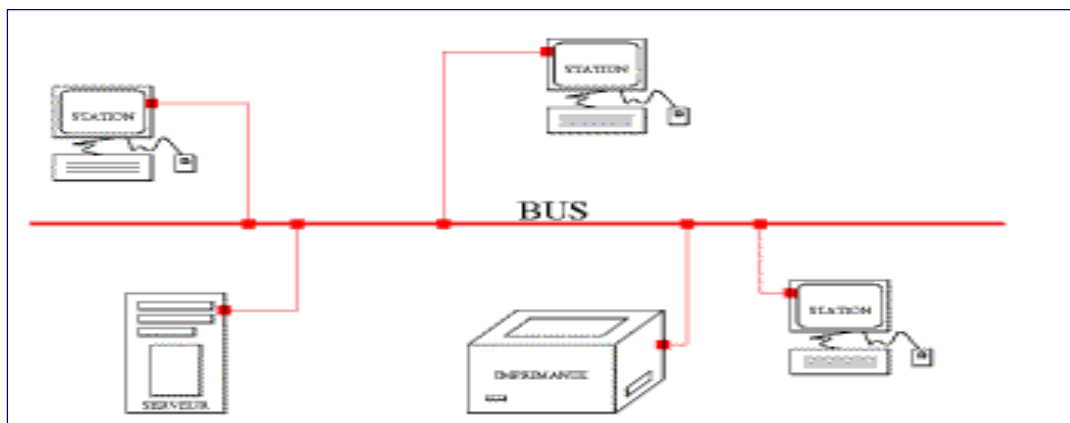


Figure 1.2 Topologie en bus

Topologie en anneau

Dans un réseau en topologie en anneau, les ordinateurs communiquent chacun à leur tour, nous avons donc une boucle d'ordinateurs sur laquelle chacun d'entre eux accède à la communication dès que le média de transmission se libère.

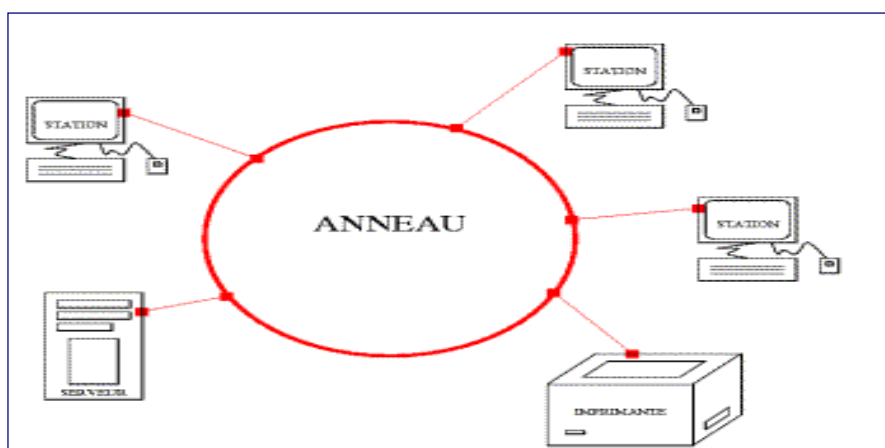


Figure 1.3 Topologie en anneau

Topologie en étoile

Dans une topologie en étoile, les ordinateurs du réseau sont reliés à un système matériel appelé Hub ou concentrateur.

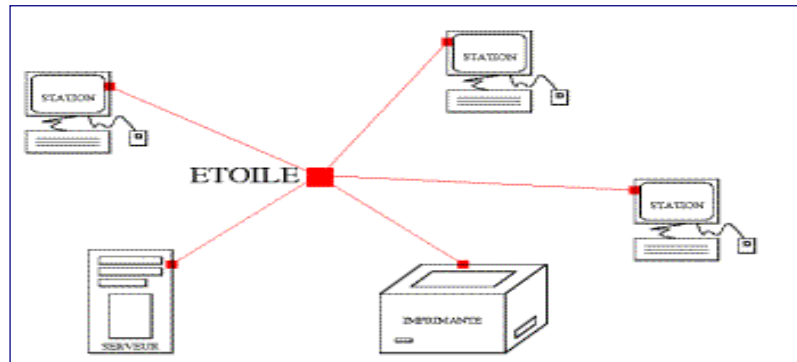


Figure 1.4 Topologie en étoile

Contrairement aux réseaux construits sur une topologie en bus, ces réseaux sont moins vulnérables aux pannes car nous pouvons aisément retirer une des connexions en la débranchant du concentrateur sans pour autant paralyser le reste du réseau.

1.2.2 Classification selon l'étendue géographique

On distingue différents types de réseaux, selon leur taille (nombre de machines), leur vitesse de transfert des données ainsi que leur localisation. On peut distinguer quatre types de réseaux :

- PAN (*Personal Area Network*)
- LAN (*Local Area Network*)
- Man (*Métropolitain Area Network*)
- WAN (*Wide Area Network*)

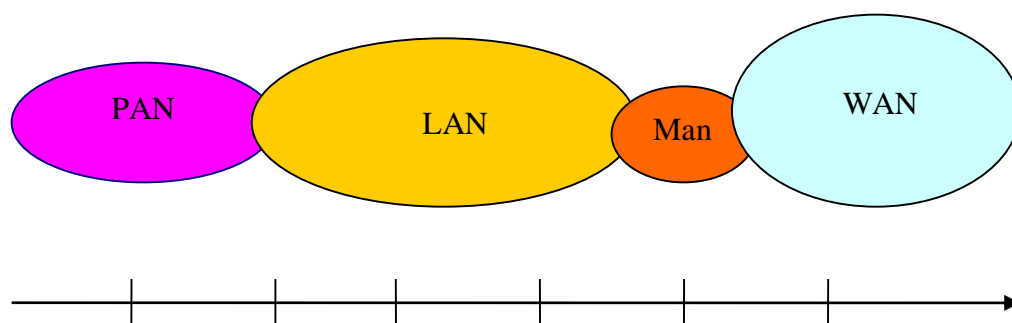


Figure 1.5 Classification selon l'étendue géographique

Réseaux personnels

Les PAN (*Personal Area Network*), la plus petite taille de réseau. Ces réseaux personnels interconnectent sur quelques mètres les équipements personnels tels que : téléphone portable, organiseur etc.... d'un même utilisateur

Réseaux locaux

LAN (*Local Area Network*). Il s'agit d'un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux dans une petite aire géographique par un réseau, souvent à l'aide d'une même technologie (la plus répandue étant Ethernet).

Réseaux métropolitains

Les MAN (*Métropolitan Area Network*) interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants. Un MAN est formé de commutateurs ou de routeurs interconnectés par des liens hauts débits (en général en fibre optique).

Réseaux étendus

Un WAN (*Wide Area Network* ou réseau étendu) permet de communiquer à l'échelle d'un pays, ou de la planète entière, les infrastructures physiques pouvant être terrestres (infrastructures au niveau sol), ou spatiales à l'aide de satellites de télécommunications c'est le cas de l'Internet.

1.2.3 Classification selon le média de transmission

Réseaux filaire

Nous avons distingués plusieurs types de connexions : Câble coaxial, Paires torsadées, Fibre optique. Ce tableau récapitule les différents types de câbles ainsi que leur débit :

Technologie	Type de câble	Débit théorique	Longueur Max	Connecteur	Coût
10 Base 2 (Thinnet)	Coaxial	10 Mbits/s	200 m	BNC	Peu cher
10 Base 5 (Thicknet)	Coaxial	100 Mbits/s	500 m	BNC	Peu cher

10 Base T	UTP cat 5	10 Mbits/s	100 m	RJ45	Bon marché
100 Base TX	UTP cat 5	100 Mbits/s	100 m	RJ45	Bon marché
10 Base FL	Fibre optique	10 Mbits/s	2000 m	SC	Elevé
100 Base FX	Fibre optique	100 Mbits/s	400 m	SC	Elevé

Tableau 1.1 Types de câbles et leurs débit

Réseaux sans fil

Est un réseau informatique ou numérisé qui connecte différents postes ou systèmes entre eux par ondes radio. Il peut être associé à un réseau de télécommunications pour réaliser des interconnexions entre nœuds. Nous pouvons citer quelques types : Bluetooth, wifi, wimax, GSM (*Global System for Mobile Communication*), GPRS (*General Packet Radio Service*), UMTS (*Universal Mobile Telecommunication System*).

Parmi ces types, nous pouvons distinguer deux familles :

Avec infrastructure

Le mode BSS est le mode généralement utilisé. Le mode BSS est également appelé mode infrastructure. Dans ce mode, plusieurs points d'accès sans fils sont connectés à un réseau câblé. Chaque réseau sans fil possède son propre nom. Ce nom est ce que l'on appelle le *Service set identifier* "SSID" du réseau.

Sans infrastructure

Le mode IBSS, également appelé mode "ad-hoc", est conçu pour les connexions point à point. Il existe en fait deux types de mode ad-hoc. Le premier est le mode IBSS, également appelé mode ad-hoc ou IEEE ad-hoc. Ce mode est défini par les normes IEEE 802.11. Le deuxième mode est appelé ad-hoc démo ou encore mode ad-hoc Lucent (et parfois, ce qui prête à confusion, mode ad-hoc). C'est l'ancien mode ad-hoc pré-standard 802.11 et ne devrait être utilisé qu'avec d'anciennes installations.

1.3 Couches protocolaires du modèle OSI

La première évolution des réseaux informatiques a été des plus anarchiques, chaque constructeur développant sa propre technologie. Le résultat fut une quasi-impossibilité de connecter différents réseaux entre eux. Pour pallier à ce problème d'interconnexions, l'ISO

(*International Standards Organisation*) décida de mettre en place un modèle de référence théorique décrivant le fonctionnement des communications réseaux.

Ainsi fût créé le modèle OSI, à partir des structures réseau prédominantes de l'époque : DECNet (*Digital Equipment Corporation's Networking développé par digital*) et SNA (*System Network Architecture* développé par IBM). Ce modèle a permis aux différents constructeurs de concevoir des réseaux interconnectables.

Le modèle OSI est un modèle conceptuel. Il a pour but d'analyser la communication en découpant les différentes étapes en 7 couches, chacune de ces couches remplissant une tâche bien spécifique.

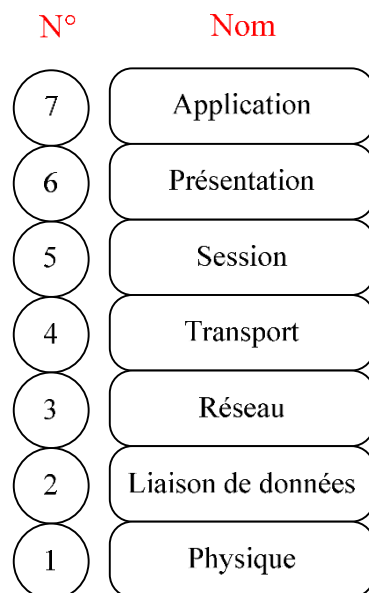


Figure 1.6 Modèle OSI

Pour communiquer entre les couches et entre les hôtes d'un réseau, OSI a recourt au principe d'encapsulation (voir figure 1.7), qui est un processus de conditionnement des données consistant à ajouter un en-tête de protocole déterminé avant que les données ne soient transmises à la couche inférieure :

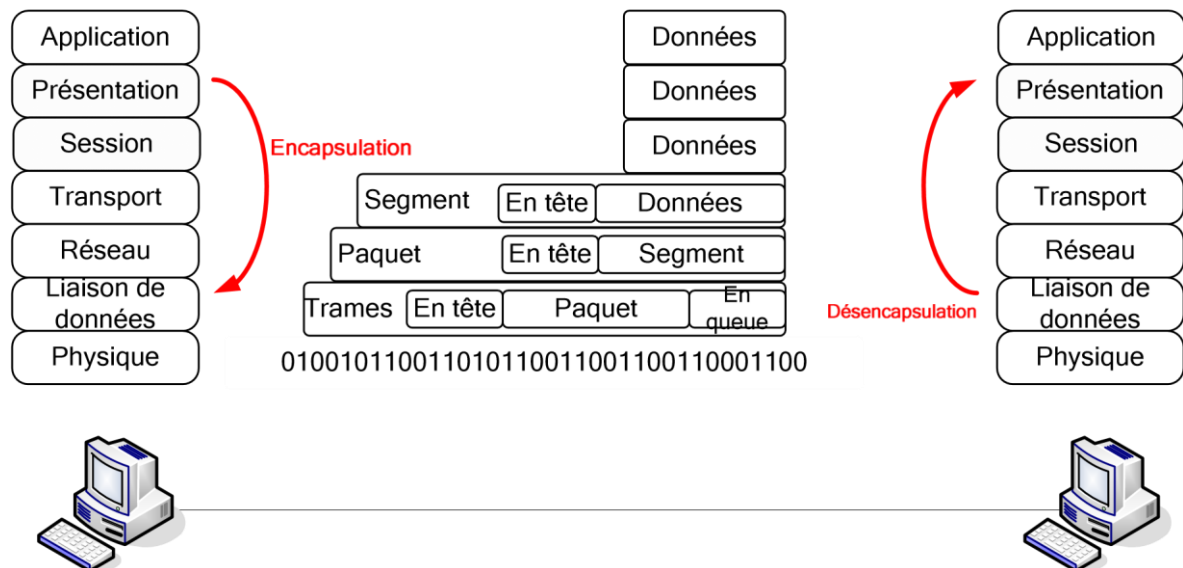


Figure 1.7 Processus d'encapsulation et de décapsulation

1.3.1 Couche physique

Dans le domaine des réseaux informatiques, la couche physique est la première couche du modèle OSI. Elle est chargée de la transmission effective des signaux électriques ou optiques entre les interlocuteurs. Son service est généralement limité à l'émission et la réception d'un bit ou d'un train de bits continu (notamment pour les supports synchrones comme la fibre optique). Cette couche est chargée de la conversion entre bits et signaux électriques ou optiques. On appelle cette conversion la modulation. Elle est en pratique toujours réalisée par un circuit électronique spécifique.

Le service de cette couche est approximativement défini par

- la norme ISO 7498-1
- précisée par ISO 10022
- précisée par la recommandation X211 du CCITT.

Dans le cas des réseaux Ethernet, les données sont transmises sur la couche physique par un PHYceiver.

Modulation

Transmission en bande de base

Ce type de transmission dit en bande de base est principalement réservé aux réseaux locaux. Le codeur en bande de base transforme les bits de données en un signal électrique numérique. Ce signal se présente comme une suite de niveaux de tension dont les amplitudes sont choisies parmi un nombre fini de possibilités.

Transmission à large bande

Lorsque la longueur de la liaison dépasse quelques centaines de mètres, les informations ne peuvent plus être transmises sur le support de liaison sans transformation du signal numérique en signal analogique. Dans une transmission à large bande, les données modulent une onde porteuse sinusoïdale dont la fréquence est bien plus élevée que le rythme de transmission des données. Le codage est alors appelé modulation et le décodage est nommé démodulation, d'où le nom de l'appareil modem.

Quelques encodages

- Codages en bande de base :
 - BHDn
 - Biphase ou Manchester
 - Bipolaire simple ou d'ordre 2
 - Manchester différentiel
 - Miller
 - MLT-3
 - HDB3
 - NRZ
 - NRZI

1.3.2 Couche liaison de données

Dans le domaine des réseaux informatiques, la **couche de liaison de données** est la seconde couche des sept couches du modèle OSI. La couche de liaison de données est la couche de protocole qui transfère les données entre les nœuds adjacents d'un réseau étendu (WAN) ou entre des nœuds sur le même segment d'un réseau local (LAN). La couche de liaison de données fournit les moyens fonctionnels et procéduraux pour le transfert de données entre des entités d'un réseau et, dans certains cas, les moyens de détecter et potentiellement corriger les erreurs qui peuvent survenir au niveau de la couche physique.

Parmi les protocoles s'exécutant au niveau de cette couche, nous citons : Ethernet pour les réseaux locaux (multi-nœuds), le protocole point à point (PPP), HDLC (*High-Level Data Link Control*) et ADCCP (*Advanced Data Communication Control Procedures*) pour des connexions points à points (double nœud) sont des exemples de protocoles de liaison de données.

Cette couche est divisée en deux sous-couches :

- La sous-couche MAC (*Média Access Control*) est chargée du contrôle de l'accès au média. C'est au niveau de cette couche que l'on retrouve les adresses de liaison de donnée (MAC, DLCI) la technologie Ethernet et la méthode d'accès CSMA/CD.
- La sous-couche LLC (*Layer Link Control*) s'occupe de la gestion des communications entre les stations et interagit avec la couche réseau (norme IEEE 802.2).

Norme	Appellation	Débit	Média utilisé
802.3	Ethernet	10 Mbps	Coaxial / UTP / fibre optique
802.3u	Fast Ethernet	100 Mbps	UTP / Fibre optique
802.3z	Gigabit Ethernet	1000 Mbps	Fibre optique
802.3ab	Gigabit Ethernet	1000 Mbps	Câble UTP
802.3ae	10 Gigabit Ethernet	10 000 Mbps	Fibre Optique

Tableau 1.2 Technologie Ethernet

Quelques protocoles de la couche liaison de données

- ARCnet
- ATM
- Cisco Discovery Protocol (CDP)
- Controller Area Network (CAN)
- Econet
- Ethernet
- Ethernet Automatic Protection Switching (EAPS)
- Fiber Distributed Data Interface (FDDI)
- Frame Relay
- High-Level Data Link Control (HDLC)
- IEEE 802.2 (fournit des fonctionnalités LLC aux couches MAC IEEE 802)
- IEEE 802.11 wireless LAN
- Link Access Procedures, D channel (LAPD)
- LocalTalk
- Multiprotocol Label Switching (MPLS)
- Point-to-Point Protocol (PPP)
- Spanning tree protocol
- StarLan
- Token ring

1.3.3 Couche réseau

Cette couche gère l'adressage de niveau trois, la sélection du chemin et l'acheminement des paquets au travers du réseau. Ses apports fonctionnels principaux sont donc:

- **ROUTAGE** : Déterminer un chemin permettant de relier les 2 machines distantes suivant des critères bien précis;
- **ACHÉMINEMENT** : Retransmission d'un PDU (*Protocol Data Unit* ou *Unité de données de protocole*) dont la destination n'est pas locale pour le rapprocher de sa destination finale.
- **Contrôle des flux** : Représente un asservissement du débit binaire de la source vers le puits. Quand une machine qui a un débit montant supérieur au débit descendant de la destination, la source diminue son débit pour ne pas submerger le puits de requêtes (obligeant parfois, vu que le puits ne peut pas les traiter, la réémission de ces dernières).

Cette couche est donc la seule à être directement concernée par la topologie du réseau. C'est aussi la dernière couche supportée par toutes les machines du réseau pour le transport des données utilisateur : les couches supérieures sont réalisées uniquement dans les machines d'extrémité. Le PDU (*Protocol Data Unit*) de cette couche est souvent appelé « paquet ». La fonction de « relayage » (terme OSI) est parfois appelée « acheminement ».

Quelques protocoles de la couche réseau :

- Border Gateway Protocol
- AppleTalk
- Connectionless Network Protocol (CLNP): un protocole OSI dont le service est inspiré de celui d'IP.
- Controller Area Network (CAN)
- Internet Control Message Protocol (ICMP): protocole de support d'IP, notamment pour la notification des erreurs
- Internet Protocol (IP)
- Internetwork packet exchange (IPX)
- Wireless Distribution System (WDS)
- X.25

Adressage IP :

Une adresse IP est une adresse 32 bits notée sous forme de 4 nombres décimaux séparés par des points. On distingue en fait deux parties dans l'adresse IP :

- Une partie désignant le réseau (on l'appelle netID)
- Une partie désignant les hôtes (on l'appelle host-ID)

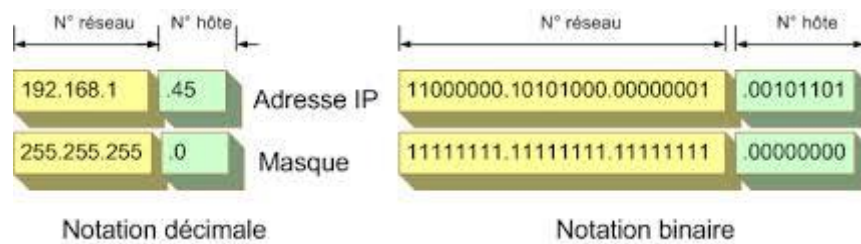


Figure 1.8 Adressage IP

Classes d'adresse IP

L'organisme chargé d'attribuer les adresses IP publiques est l'InterNIC (*Internet Network Information Center*).

On appelle « Bits de poids fort », les premiers bits de l'octet le plus à gauche. Les adresses IP sont réparties en plusieurs classes, en fonction des bits qui les composent :

Classe	Bits de départ	Début	Fin	Notation CIDR	Masque de sous-réseau par défaut
Classe A	0	0.0.0.0	127.255.255.255	/8	255.0.0.0
Classe B	10	128.0.0.0	191.255.255.255	/16	255.255.0.0
Classe C	110	192.0.0.0	223.255.255.255	/24	255.255.255.0
Classe D (multicast)	1110	224.0.0.0	239.255.255.255	/8	non défini

Classe E (réservée)	1111	240.0.0.0	255.255.255.255		non défini
----------------------------	------	-----------	-----------------	--	------------

Tableau 1.3 Classes d'adressage IP

IPv4 et IPv6

IPv4 est le protocole le plus couramment utilisé sur Internet tout comme sur les réseaux privés. IPv6 est son successeur. IPv4 utilise des adresses codées sur 32 bits (soit en théorie 4 294 967 296 adresses possibles) tandis qu'IPv6 les code sur 128 bits (soit en théorie $3,4 \times 10^{38}$ adresses possibles).

Le premier champ d'un paquet d'un protocole IP est composé de 4 bits qui indiquent la version du protocole utilisé. La valeur 0100 (4 en binaire) est utilisée pour IPv4, 0110 (6 en binaire) pour IPv6. La valeur 0101 (5 en binaire) est utilisée pour le protocole Internet Stream Protocol, la valeur 0111 (7 en binaire) pour TP/IX ([RFC 1475](#)), 1000 (8 en binaire) pour PIP ([RFC 1621](#)) et 1001 (9 en binaire) pour TUBA (« *TCP and UDP with Bigger Addresses* », [RFC 1347](#))

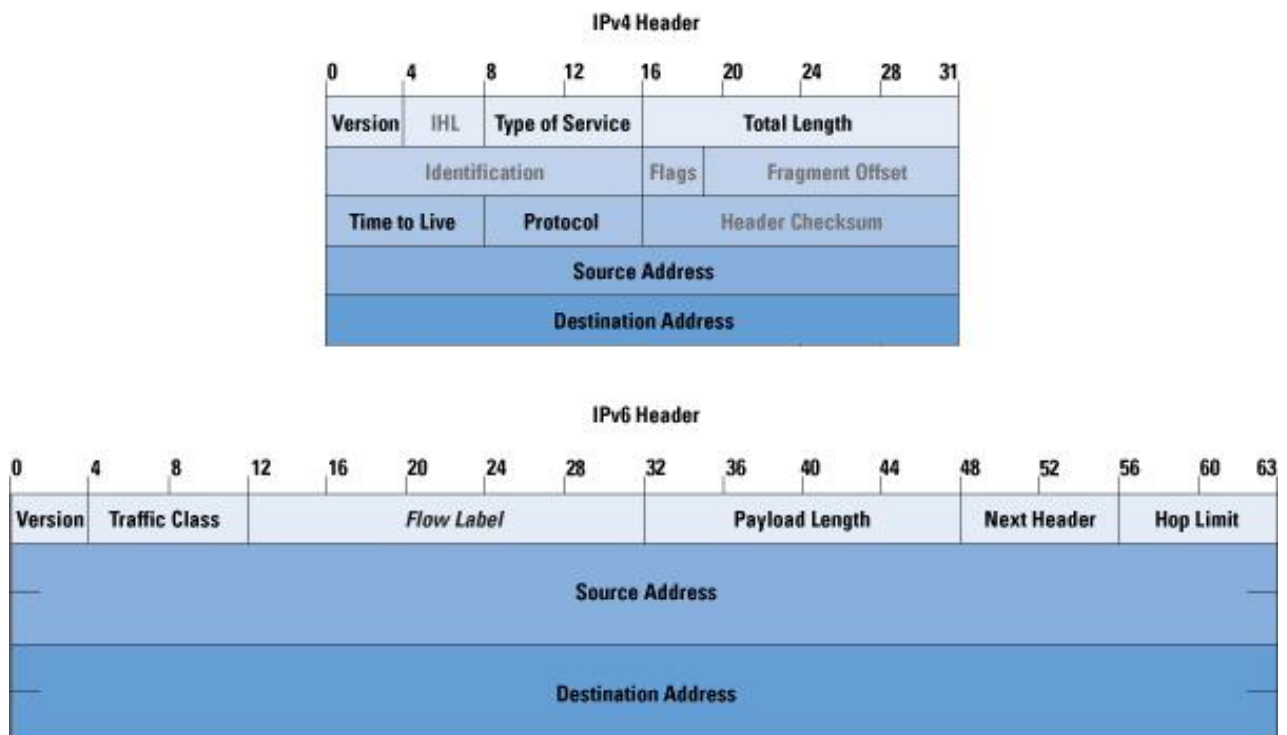


Figure 1.9 entête Ipv4 et Ipv6

1.3.4 Couche transport

La couche transport gère les communications de bout en bout entre processus. Cette couche est souvent la plus haute couche dont on se préoccupe de la correction des erreurs (exception

connue dans le monde IP : utilisation de DNS sur UDP). C'est-à-dire que le service de niveau transport consiste généralement en un service en mode connecté offrant le transfert de messages ou d'octets bruts garantis sans corruption, pertes, ré ordonnancement, duplication. En particulier, c'est le service offert par les protocoles TCP (dans le monde IP.

En fonction des protocoles, le PDU est appelé « message », « paquet » ou, uniquement dans le cas TCP, « fragment » (ou « segment » en Internet).

On peut aussi parler du multiplexage au niveau de la couche transport.

TCP (Transmission Control Protocol)

Est un protocole de transport fiable, en mode connecté, documenté dans la RFC 793 de l'IETF. Dans le modèle Internet, aussi appelé modèle TCP/IP, TCP est situé au niveau de la couche transport (entre la couche réseau et la couche session). Les applications transmettent des flux de données sur une connexion réseau, et TCP découpe le flux d'octets en *segments*, dont la taille dépend de la MTU (*Minimum Transfert Unit*) du réseau sous-jacent (couche liaison de données).

TCP a été développé en 1973, puis adopté pour Arpanet en 1983 comme remplacement de NCP (RFC 801).

UDP (User Datagram Protocol)

Est un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport de la pile de protocole TCP/IP : dans l'adaptation approximative de cette dernière au modèle OSI, il appartiendrait à la couche 4, comme TCP. Il est détaillé dans la RFC 768

TCP versus UDP

TCP découpe le message en petites séquences nommées *segments* qui sont numérotées puis envoyés à l'hôte de destination qui à son tour va le reconstituer à l'aide des numéros de séquence des paquets avant l'envoi de l'acquittement. Quant au protocole UDP, celui-ci permet de transmettre des paquets de données sans établir préalablement une connexion entre les deux entités communicantes, dans ce cas-là, les paquets peuvent emprunter des chemins différents, et l'acquittement n'est pas assuré dans ce mode.

Notion de port

Correspondant à la couche de transport du modèle OSI, la notion de **port** logiciel permet, sur un ordinateur donné, de distinguer différents interlocuteurs. Ces interlocuteurs sont des programmes informatiques qui, selon les cas, écoutent ou émettent des informations sur ces ports. Un port est distingué par son numéro. Nous montrons dans la figure qui suit quelques ports :

Décimal	Mot-clé	Description
0		Réservé
1-4		Non attribué
5	RJE	Soumission de travaux à distance
7	ECHO	Écho
9	DISCARD	Élimination
11	USERS	Utilisateurs actifs
13	DAYTIME	Heure du jour
15	NETSTAT	Who is Up ou NETSTAT
17	QUOTE	Citation du jour
19	CHARGEN	Générateur de caractères
20	FTP-DATA	Protocole FTP (données)
21	FTP	Protocole FTP
23	TELNET	Connexion en mode terminal
25	SMTP	Protocole SMTP (Simple Mail Transfer Protocol)
37	TIME	Heure du jour
39	RLP	Protocole RLP (Resource Location Protocol)

Tableau 1.4 Quelques ports réservés

1.3.5 Couche session

Les 2 services originaux de la couche session sont la synchronisation des communications (n'importe quel intervenant peut émettre à tout moment) et la gestion des « transactions ». Un service cependant a été rajouté, c'est un mécanisme de correction des erreurs de traitement par restauration d'un état antérieur connu.

Les services de transport sont des services de communication point à point, c'est-à-dire avec deux interlocuteurs. Mais le modèle OSI doit aussi convenir aux communications multipoints. Deux genres de communications multipoints sont explicitement mentionnées dans la norme : les communications en étoile où une session est un ensemble de communications point à point avec un interlocuteur engagé dans tous les échanges ; et la diffusion où tous les interlocuteurs

reçoivent tous les messages. Des mécanismes de synchronisation sont alors requis pour savoir par exemple qui répond à quoi. C'est le rôle des protocoles de cette couche.

Quelques protocoles de la couche session : AppleTalk, NetBIOS, RPC.

1.3.6 Couche présentation

La couche présentation est chargée du codage des données applicatives. Les couches 1 à 5 transportent des octets bruts sans se préoccuper de leur signification. Mais ce qui doit être transporté en pratique, c'est du texte, des nombres et parfois des structures de données arbitrairement complexes. Un protocole de routage par exemple doit transporter un graphe représentant au moins partiellement la topologie du réseau. Le rôle de la couche présentation est donc de convertir les données applicatives manipulées par les programmes en chaînes d'octets effectivement transportées par le réseau.

1.3.7 Couche application

La couche application est surtout, du point de vue du modèle, le point d'accès aux services réseaux. Comme le modèle n'a pas pour rôle de spécifier les applications, il ne spécifie pas de service à ce niveau.

La couche d'application représente des données pour l'utilisateur ainsi que du codage et un contrôle du dialogue : des mécanismes de communication offerts aux applications de l'utilisateur.

Quelques protocoles de la couche application

HTTP, HTTPS, Telnet, SMTP, IMAP, SSH, POP3, SNMP.....

1.4 Couches protocolaires du modèle TCP/IP

La forme actuelle de TCP/IP résulte du rôle historique que ce système de protocoles a joué dans le parachèvement de ce qui allait devenir Internet. À l'instar des nombreux développements de ces dernières années, Internet est issu des recherches lancées par le DOD (*Department Of Defense*), département de la défense américaine [PE09].

Ses fonctions essentielles ne devaient en aucun cas se trouver en un seul point, ce qui le rendrait trop vulnérable. C'est alors que fut mis en place le projet ARPANet (*Advanced Research Projects Agency Network du DOD*), qui allait devenir par la suite le système d'interconnexion de réseau qui régit ce que l'on appelle aujourd'hui Internet : TCP/IP.

TCP/IP est un modèle comprenant 4 couches :

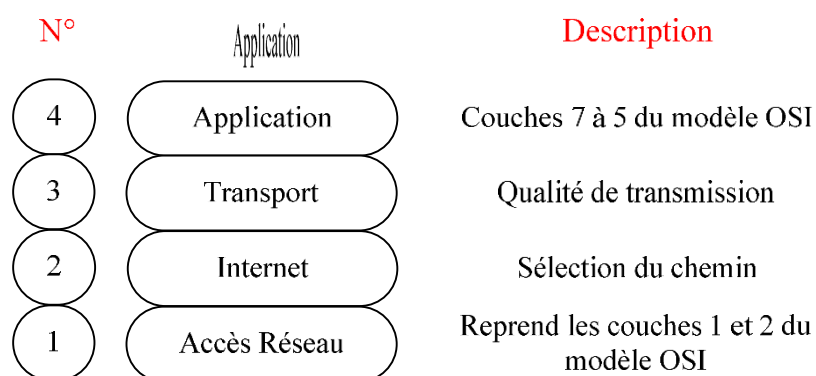


Figure 1.10 modèle TCP/IP

1.4.1 Couche accès réseau

Cette couche est assez "étrange". En effet, elle semble "regrouper" les couches physiques et liaison de données du modèle OSI. En fait, cette couche n'a pas vraiment été spécifiée ; la seule contrainte de cette couche, c'est de permettre un hôte d'envoyer des paquets IP sur le réseau. L'implémentation de cette couche est laissée libre. De manière plus concrète, cette implémentation est typique de la technologie utilisée sur le réseau local. Par exemple, beaucoup de réseaux locaux utilisent Ethernet ; Ethernet est une implémentation de la couche hôte-réseau.

1.4.2 Couche internet

Cette couche est la clé de voûte de l'architecture. Cette couche réalise l'interconnexion des réseaux (hétérogènes) distants sans connexion. Son rôle est de permettre l'injection de paquets dans n'importe quel réseau et l'acheminement de ces paquets indépendamment les uns des autres jusqu'à destination. Comme aucune connexion n'est établie au préalable, les paquets peuvent arriver dans le désordre ; le contrôle de l'ordre de remise est éventuellement la tâche des couches supérieures. Du fait du rôle imminent de cette couche dans l'acheminement des paquets, le point critique de cette couche est le routage. C'est en ce sens que l'on peut se permettre de comparer cette couche avec la couche réseau du modèle OSI. La couche internet possède une implémentation officielle : le protocole IP (Internet Protocol). Remarquons que le nom de la couche ("internet") est écrit avec un i minuscule, pour la simple et bonne raison que le mot internet est pris ici au sens large (littéralement, "interconnexion de réseaux"), même si l'Internet utilise cette couche.

1.4.3 Couche transport

Son rôle est le même que celui de la couche transport du modèle OSI : permettre à des entités paires de soutenir une conversation. Officiellement, cette couche n'a que deux implémentations

: le protocole TCP (Transmission Control Protocol) et le protocole UDP (User Datagram Protocol). TCP est un protocole fiable, orienté connexion, qui permet l'acheminement sans erreur de paquets issus d'une machine d'un internet à une autre machine du même internet. Son rôle est de fragmenter le message à transmettre de manière à pouvoir le faire passer sur la couche internet. A l'inverse, sur la machine destination, TCP replace dans l'ordre les fragments transmis sur la couche internet pour reconstruire le message initial. TCP s'occupe également du contrôle de flux de la connexion. UDP est en revanche un protocole plus simple que TCP : il est non fiable et sans connexion. Son utilisation présuppose que l'on n'a pas besoin ni du contrôle de flux, ni de la conservation de l'ordre de remise des paquets. Par exemple, on l'utilise lorsque la couche application se charge de la remise en ordre des messages. On se souvient que dans le modèle OSI, plusieurs couches ont à charge la vérification de l'ordre de remise des messages. C'est là un avantage du modèle TCP/IP sur le modèle OSI, mais nous y reviendrons plus tard. Une autre utilisation d'UDP : la transmission de la voix. En effet, l'inversion de 2 phonèmes ne gêne en rien la compréhension du message final. De manière plus générale, UDP intervient lorsque le temps de remise des paquets est prédominant.

1.4.4 Couche application

Contrairement au modèle OSI, c'est la couche immédiatement supérieure à la couche transport, tout simplement parce que les couches présentation et session sont apparues inutiles. On s'est en effet aperçu avec l'usage que les logiciels réseau n'utilisent que très rarement ces 2 couches, et finalement, le modèle OSI dépouillé de ces 2 couches ressemble fortement au modèle TCP/IP. Cette couche contient tous les protocoles de haut niveau, comme par exemple Telnet, TFTP (trivial File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), HTTP (HyperText Transfer Protocol). Le point important pour cette couche est le choix du protocole de transport à utiliser. Par exemple, TFTP (surtout utilisé sur réseaux locaux) utilisera UDP, car on part du principe que les liaisons physiques sont suffisamment fiables et les temps de transmission suffisamment courts pour qu'il n'y ait pas d'inversion de paquets à l'arrivée. Ce choix rend TFTP plus rapide que le protocole FTP qui utilise TCP. A l'inverse, SMTP utilise TCP, car pour la remise du courrier électronique, on veut que tous les messages parviennent intégralement et sans erreurs.

1.5 Modèle OSI versus Modèle TCP/IP

Tout d'abord, les points communs. Les modèles OSI et TCP/IP sont tous les deux fondés sur le concept de pile de protocoles indépendants. Ensuite, les fonctionnalités des couches sont globalement les mêmes.

Au niveau des différences, on peut remarquer la chose suivante : le modèle OSI faisait clairement la différence entre 3 concepts principaux, alors que ce n'est plus tout à fait le cas pour le modèle TCP/IP. Ces 3 concepts sont les concepts de services, interfaces et protocoles. En effet, TCP/IP fait peu la distinction entre ces concepts, et ce malgré les efforts des concepteurs pour se rapprocher de l'OSI. Cela est dû au fait que pour le modèle TCP/IP, ce sont les protocoles qui sont d'abord apparus. Le modèle ne fait finalement que donner une justification théorique aux protocoles, sans les rendre véritablement indépendants les uns des autres.

Enfin, la dernière grande différence est liée au mode de connexion. Certes, les modes orienté connexion et sans connexion sont disponibles dans les deux modèles mais pas à la même couche : pour le modèle OSI, ils ne sont disponibles qu'au niveau de la couche réseau (au niveau de la couche transport, seul le mode orienté connexion n'est disponible), alors qu'ils ne sont disponibles qu'au niveau de la couche transport pour le modèle TCP/IP (la couche internet n'offre que le mode sans connexion). Le modèle TCP/IP a donc cet avantage par rapport au modèle OSI : les applications (qui utilisent directement la couche transport) ont véritablement le choix entre les deux modes de connexion.

1.6 Architecture client/serveur

L'environnement client-serveur désigne un mode de communication à travers un réseau entre plusieurs programmes ou logiciels : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'ordinateur sur lequel est exécuté le logiciel client, et le serveur, l'ordinateur sur lequel est exécuté le logiciel serveur.

En général, les serveurs sont des ordinateurs dédiés au logiciel serveur qu'ils abritent, et dotés de capacités supérieures à celles des ordinateurs personnels en termes de puissance de calcul, d'entrées-sorties et de connexions réseau. Les clients sont souvent des ordinateurs personnels ou des appareils individuels (téléphone, tablette), mais pas systématiquement. Un serveur peut répondre aux requêtes d'un grand nombre de clients.

Il existe une grande variété de logiciels serveurs et de logiciels clients en fonction des besoins à servir : un serveur web publie des pages web demandées par des navigateurs web ; un serveur de messagerie électronique envoie des mails à des clients de messagerie ; un serveur de fichiers permet de stocker et consulter des fichiers sur le réseau ; un serveur de données à communiquer des données stockées dans une base de données, etc.

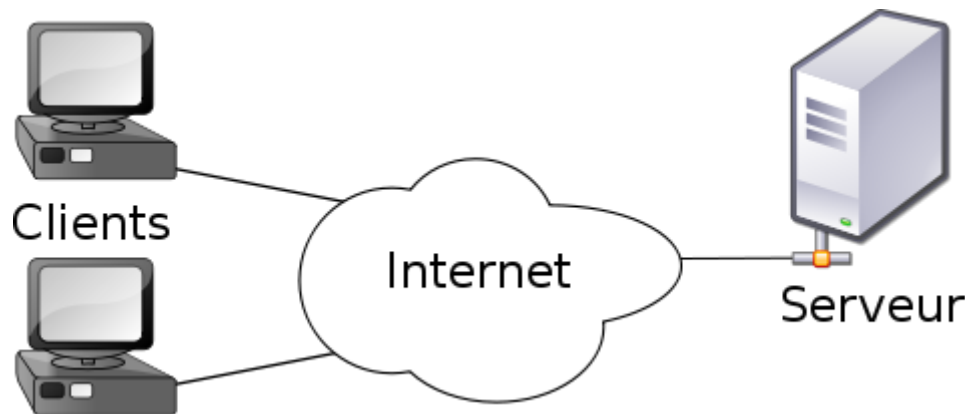


Figure 1.11 Architecture client / serveur

Client léger

Le poste client accède à une application située sur un ordinateur dit « serveur » via une interface et un navigateur Web. L'application fonctionne entièrement sur le serveur, le poste client reçoit la réponse « toute faite » à la demande (requête) qu'il a formulée.

Client Lourd

Le poste client doit comporter un système d'exploitation capable d'exécuter en local une partie des traitements. Le traitement de la réponse à la requête du client utilisateur va mettre en œuvre un travail combiné entre l'ordinateur serveur et le poste client.

Client riche

Une interface graphique plus évoluée permet de mettre en œuvre des fonctionnalités comparables à celles d'un client "lourd". Les traitements sont effectués majoritairement sur le serveur, la réponse "semi-finie" étant envoyée au poste client, où le client "riche" est capable de la finaliser et de la présenter.

1.6.2 Serveur

Dans un réseau informatique, un serveur est à la fois un ensemble de logiciels et l'ordinateur les hébergeant dont le rôle est de répondre de manière automatique à des demandes envoyées par des clients ordinateur et logiciel via le réseau.

Les serveurs sont d'usage courant dans les centres de traitement de données, les entreprises, les institutions, et le réseau Internet, où ils sont souvent un point central et sont utilisés simultanément par de nombreux utilisateurs pour stocker, partager et échanger des informations. Les différents usagers opèrent à partir d'un client: ordinateur personnel, poste de travail, ou terminal. Le World Wide Web, la messagerie électronique et le partage de fichiers sont quelques applications informatiques qui font usage de serveurs.

1.6.3 Niveaux d'architecture client/serveur

Architecture à 2 niveaux

Ce type d'architecture (2-tier en anglais) caractérise les environnements client-serveur où le poste client demande une ressource au serveur qui la fournit à partir de ses propres ressources.

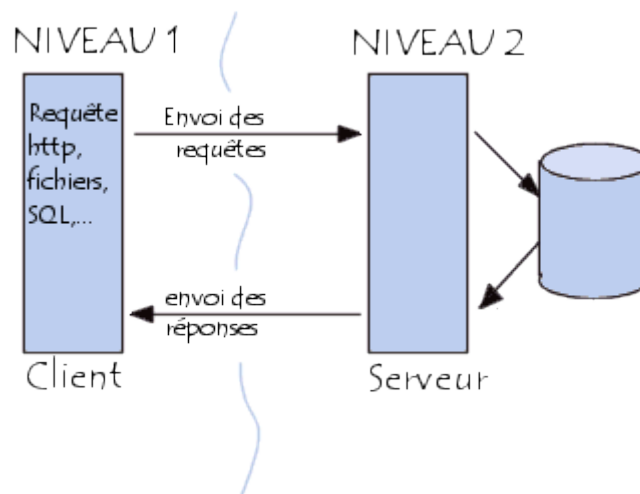


Figure 1.12 Architecture à 2 niveaux

Architecture à 3 niveaux

Dans cette architecture (3-tier en anglais), aussi nommée trois tiers en français, un niveau supplémentaire est ajouté :

- Un client (l'ordinateur demandeur de ressources) équipé d'une interface utilisateur (généralement un navigateur web) chargée de la présentation.
- Un serveur d'application (appelé middleware) qui fournit la ressource, mais en faisant appel à un autre serveur.
- Un serveur de données qui fournit au serveur d'application les données requises pour répondre au client.

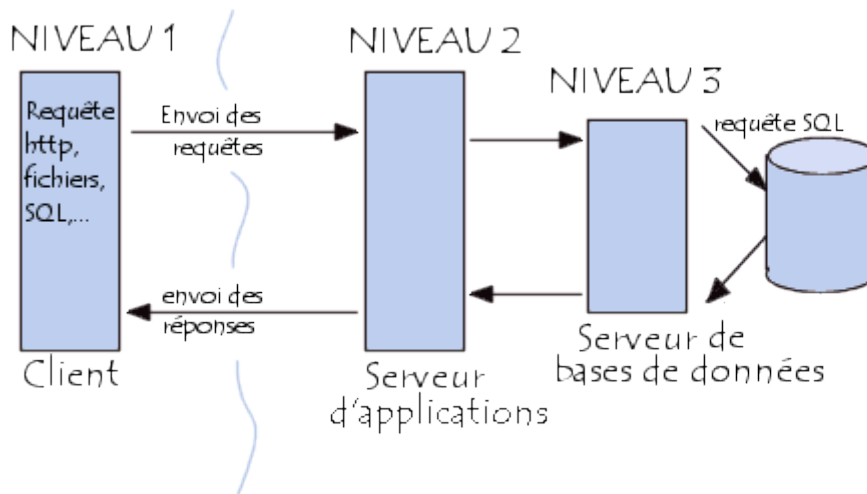


Figure 1.13 Architecture à 3 niveaux

Architecture à n niveaux

L'architecture 3 niveaux permet de spécialiser les serveurs dans une tâche précise : avantage de flexibilité, de sécurité et de performance. L'architecture peut être étendue sur un nombre de niveaux plus important : on parle dans ce cas d'architecture à N niveaux (ou multi-tiers).

1.7 Web

1.7.1 Présentation

Le Web ou le World Wide Web (WWW) est un système hypertexte qui rend au public une lecture plus dynamique, fonctionnant sur l'Internet. Le principe du web repose sur l'utilisation des hyperliens qui lient les pages web entre elles, pour naviguer entre des documents multimédias (pages de textes enrichies de sons, graphiques, images fixes et animées, vidéos,...) grâce à un logiciel appelé navigateur. Ce dernier permet d'effectuer une demande par l'intermédiaire d'une adresse (URL) et de lire le résultat de cette recherche.

1.7.2 Page web

Est une ressource du *World Wide Web* conçue pour être consultée par des visiteurs à l'aide d'un navigateur et une adresse web. Techniquement une page web est souvent constituée d'un document en *Hypertext Markup Language* (HTML ou XHTML) et d'images. Cependant, tous types de ressources ou d'assemblage de ressources, textuelles, visuelles, sonores, logicielles, peuvent constituer une page web.

1.7.3 Site web

Est un ensemble de pages web liées entre elles et mises en ligne à une adresse web. Un site web est habituellement architecturé autour d'une page centrale, appelée « page d'accueil » et proposant des liens vers un ensemble de pages hébergées sur le même serveur, parfois ces liens sont dit « externes », c'est-à-dire de pages hébergées par un autre serveur. Il existe deux types de site web : statique et dynamique.

1.7.4 Site web statique

Un site web statique est un site où chacune des pages est créée en HTML. Un ordinateur qui se connecte au serveur, demande une page, celle-ci lui est directement servie (elle est stockée toute prête sur le serveur).

1.7.4 Applications web

Un site Web dynamique est un site dont les pages sont générées dynamiquement à la demande. Le contenu est obtenu (par exemple) en combinant l'utilisation d'un langage de scripts ou de programmation et une base de données. Il s'agit souvent de PHP pour le langage et MySQL pour la base de données.

1.8 Conclusion

Dans ce chapitre nous avons abordé tous les aspects des réseaux informatiques. Ensuite nous avons parlé d'Internet et ses différents services principalement le WEB sur lequel on a donné une vue globale sur ses différentes technologies, du côté client et du côté serveur. Avant de savoir quelles solutions technologiques apportées, il faut d'abord identifier les besoins réels du système étudié et les évolutions possibles, pour cela nous avons jugé utile de présenter l'entreprise qui sera l'objet du prochain chapitre.

Chapitre 2 : Etude préalable

2.1 Introduction

Dans ce chapitre, nous allons nous intéresser au domaine de notre projet de fin d'étude portant sur l'automatisation du dispositif d'aide à l'insertion des jeunes demandeur d'emploi sans qualification. Pour se faire, nous commençons par une présentation de l'organisme d'accueil, puis nous nous fixons sur le champ de notre étude avant de passer à l'analyse préliminaire et de conclure le chapitre.

2.2 Organisme d'accueil

Dans le cadre de notre projet de fin d'étude, nous avons eu la chance d'effectuer notre stage au sein de l'Agence Nationale de l'Emploi de Tizi-Ouzou (ANEM), ce qui nous a permis d'acquérir des compétences professionnelles mettant en œuvre les acquis de notre formation universitaire. Dans cette section, sont présentées l'agence nationale de l'emploi, sa mission et enfin son organisation.

2.2.1 Présentation de l'agence nationale de l'emploi Tizi-Ouzou

En 1919, l'Organisation Internationale du Travail (OIT) recommande aux états membres, la mise en place du service public de l'emploi (SPE) à travers la convention N°02 sur le chômage. Cette mission est mise en œuvre par l'Agence Nationale de l'Emploi en succession à l'Office Nationale de la Main d'œuvre (ONAMO). Elle est instituée par décret exécutif portant le numéro 06-77 du 18 Février fixant les missions, l'organisation et le fonctionnement de l'Agence Nationale de L'Emploi.

2.2.2 Missions de l'agence nationale de l'emploi Tizi-Ouzou

L'Agence Nationale de l'Emploi a pour mission d'organiser et d'assurer la connaissance de la situation et de l'évolution du marché national de l'emploi et de la main d'œuvre et de garantir à tout demandeur et à toute entreprise un service de recrutement efficace et personnalisé. A ce titre, l'ANEM est chargée de :

- Mettre en place un système d'information permettant de renseigner de manière précise, régulière et fiable sur les fluctuations du marché de l'emploi et de la main d'œuvre.
- Procéder a toute analyse et expertise en matière d'emploi et de main d'œuvre.
- Entreprendre toutes études et enquêtes liées à l'accomplissement de sa mission.
- Développer et normaliser les instruments et outils permettant le développement de la fonction d'observation du marché de l'emploi.

- Recueillir et mettre en relation l'offre et la demande de travail
- Assurer l'accueil l'information, l'orientation et le placement des demandeurs d'emploi
- Procéder à la prospection et la collecte d'offre d'emploi auprès des organismes employeurs.
- Organiser la compensation des offres et des demandes d'emploi au niveau national régional et local.
- Favoriser la mobilité géographique et professionnelle des demandeurs d'emploi.

2.2.4 Organisation de l'Agence Nationale de l'Emploi Tizi-Ouzou

La forme d'organisation d'une agence permettant de concourir efficacement et de manière uniforme, à la prise en charge des activités qui lui échoient repose sur le schéma qui suit :

- Zone libre accès ;
- Service demandeurs ;
- Service employeurs ;
- Service dispositifs ;
- Service traitement de l'information ;
- Secrétariat et appui logistique.

Organigramme d'une agence locale de l'emploi

La figure suivante illustre l'organigramme d'une agence locale de l'emploi

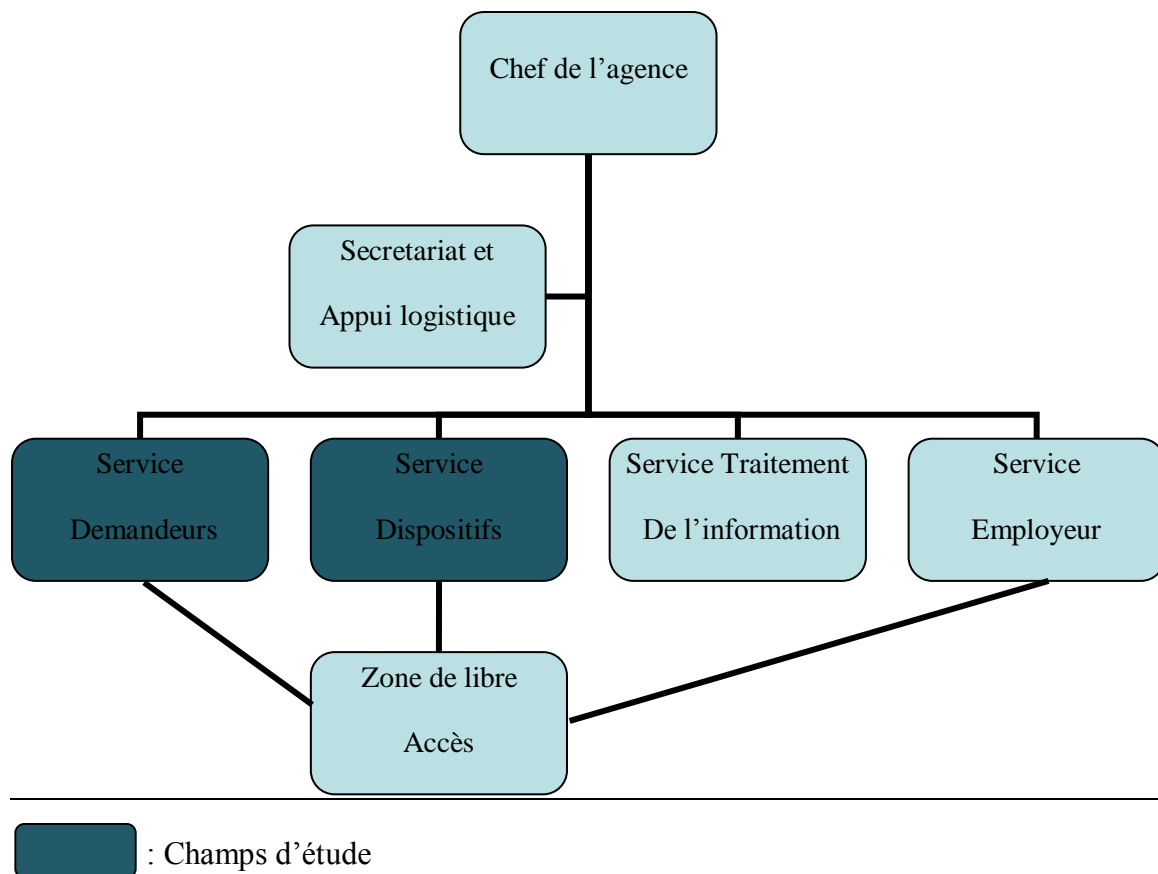


Figure 2.1 Organigramme d'une agence nationale de l'emploi

2.2.5 Dispositif d'aide à l'insertion des jeunes demandeurs d'emploi

C'est un dispositif mis en place par l'ANEM pour former et insérer les jeunes demandeurs sans aucune qualification dans les métiers déficitaires.

2.3 Champ d'étude

Les services demandeurs et dispositifs se chargent principalement de l'accueil, informations sur les différents dispositifs et formation et l'orientation des demandeurs d'emploi

2.3.1 Diagramme des flux

Le diagramme des flux est une représentation simple de l'organisation d'une entreprise, il fait apparaître les différents échanges entre les différents acteurs du domaine étudié ainsi qu'avec leur environnement.

Concepts de base

- **Acteur** : un acteur représente une unité active, c'est un agent capable d'échanger des informations avec d'autres agents. Il peut être interne ou externe. Il est interne s'il appartient au domaine d'étude et externe dans le cas contraire.

- **Flux d'information** : il représente un échange de données entre deux acteurs dans le cadre du système d'information concerné.

Description des flux

Numéro	Désignation des flux
01	Demande d'inscription
02	Carte Bleue
03	Procès-verbal d'ouverture de la section
04	Liste étudiant
05	Contrat non signé, procès verbale et la liste des étudiants
06	Validation contrat
07	Copie contrat
08	Fiche de pointage
09	Fiche de pointage
10	Versement de la prime

Tableau 2.1 Description des flux

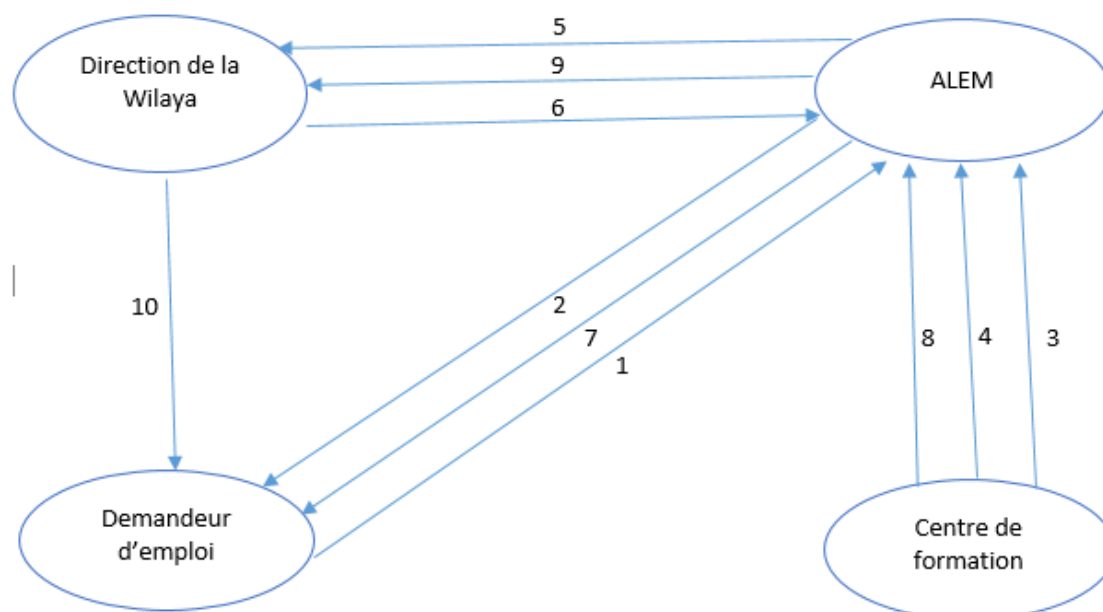


Figure 2.2 Diagramme des flux

2.3.2 Service demandeur d'emploi

Ressources humaines

Nous avons recensé dans ce service trois agents qui effectuent tous les mêmes tâches, qui sont l'accueil, inscription et orientation des demandeurs d'emploi.

Ressources matérielles

Ce service dispose de trois ordinateurs de bureau doté d'un système d'exploitation Windows 7 et d'une licence antivirus Kaspersky, une imprimante et trois onduleurs.

2.3.3 Service dispositif d'aide à l'insertion professionnelle

Ressources humaines

Nous avons recensé dans ce service trois agents qui effectuent tous les mêmes tâches. Ces tâches sont : informer les demandeurs d'emploi sur les différents dispositifs mis en place et la gestion des offres d'emploi.

Ressources matérielles

Ce service dispose de trois ordinateurs de bureau doté d'un système d'exploitation Windows 7 et d'une licence antivirus Kaspersky, deux imprimantes dont une multifonction et trois onduleurs.

2.3.4 Problématiques et expression des besoins

Problématiques

Afin de gérer leurs différentes candidatures, l'agence ne possède aucune application. En effet, toutes les candidatures des jeunes âgés entre 16 et 20 ans s'effectuent manuellement. Cette gestion manuelle reste cependant inadaptée, puisqu'elle ne permet pas d'établir un véritable suivi efficace et entraîne un risque de perte de données. Tout dépôt de candidature au sein de cet organisme doit avoir lieu de manière unique seulement. Or, en ayant recours à cette procédure manuelle, on ne peut savoir si une candidature donnée existait déjà précédemment ou non.

Expression des besoins

Pour résoudre les problèmes cités précédemment, il est indispensable de mettre en place une base de données propre au dispositif d'aide à l'insertion professionnelle au profit des

demandeurs d'emploi sans aucune qualification. A cette base de données, une application permettant de manipuler les informations contenues dans cette base de façon fiable et efficace.

2.4 Analyse préliminaire

C'est la mise en revue des besoins du client, une évaluation de ses opérations et un survol de ses ressources informatiques actuelles est essentielle pour une bonne compréhension de la situation et des besoins du client. L'analyse préliminaire peut prendre plusieurs formes selon les besoins et la complexité du dossier.

2.4.1 Etude des postes de travail

Dans cette partie nous allons définir les différents postes de travail participant à la gestion des stagiaires.

Définition d'un poste de travail

Un poste est une composante organisationnelle situé sur un site. Il est constitué d'un ou plusieurs agents ayants un certain nombre de tâches à exécuter et des informations à collecter ou à transmettre. Dans ces postes il s'agit d'énumérer les postes directement concernés par le champ d'étude.

Identification des acteurs

Un acteur représente une entité qui joue un rôle d'interagir directement avec le système.

Pendant le stage que nous avons effectué au niveau de l'agence nationale de l'emploi ANEM, nous avons procédé à l'identification des acteurs qui seront les futurs utilisateurs de AGSA (application de gestion des stagiaires de l'ANEM).les différents acteurs sont énumérés si dessous :

- **Conseiller d'orientation** : Il gère les candidats, sessions, centre de formations et organismes d'accueil.

Nous avons mis en place une extension web à la disposition du candidat pour effectuer une préinscription et suivre l'avancement de son dossier.

Liste des postes de travail

Notre domaine d'étude contient un seul poste de travail qui est :

- Conseiller en orientation

Fiche d'étude du poste N°01		
Code du poste : 01 Désignation du poste : Conseillé en orientation Service de rattachement : Service demandeurs. Responsabilité : inscription des demandeurs emploi; Acteurs : 03		
Tâche du poste		
Désignation	Fréquence	Délai
<ul style="list-style-type: none"> – accueillir les nouveaux candidats. – vérification du dossier d'inscription. – inscription des nouveaux candidats. – Etablissement de contrats. – Délivrance des différentes attestations. – suivi du dossier d'inscription 	à chaque opération	Variable

Tableau 2.2 liste des postes de travail

Identification des acteurs

Un acteur représente une entité qui joue un rôle d'interagir directement avec le système. Pendant le stage que nous avons effectué au niveau de l'agence nationale de l'emploi ANEM, nous avons procédé à l'identification des acteurs qui seront les futurs utilisateurs de AGSA (application de gestion des stagiaires de l'ANEM).les différents acteurs sont énumérés si dessous :

- **Conseiller d'orientation** : Il gère les candidats, sessions, centre de formations et organismes d'accueil.

Nous avons mis en place une extension web à la disposition du candidat pour effectuer une préinscription et suivre l'avancement de son dossier.

2.4.2 Etude des documents

L'étude des documents permet de recenser tous les documents manipulés et de les analyser. Une fiche d'étude sera utilisée à cette fin qui comportera toutes les caractéristiques du document : code, désignation, rôle, nombres d'exemplaires.

Chaque information sera décrite par sa taille, son type et des observations s'il y a lieu. Le type de rubriques peut être :

A : Alphabétique, N : Numérique, AN : Alphanumérique, DATE : date

L'objectif de cette étude est de :

- Déceler l'origine et la destination de chaque document ;
- Voir si les documents répondent aux besoins pour lesquels ils ont été créés.

Parmi les documents, on distingue :

- **Document interne de position** : on appelle document interne de position, tout document utilisé dans le service même où il a été créé.
- **Document interne de liaison** : on appelle document interne de liaison, tout document transitant entre les services de l'organisme.
- **Document externe** : on appelle document externe, tout document provenant de l'environnement externe de l'organisme ou diffusé vers celui-ci.

Liste des documents

Nous avons recensé les documents suivants :

Code	Désignation	Nombre d'exemplaire
FPR	Feuille de présence	
LC	Liste de candidats	
PVOS	Procès-verbal d'ouverture de la section	
CFJ	Contrat de formation pour les jeunes âgés de 16 à 20 ans Inscrit au niveau de l'ANEM comme demandeurs d'emplois	
FP	Fiche de pointage	
CB	Carte bleu	
LO	Lettre d'orientation	

Tableau 2.3 codification de la liste des documents

A présent nous présentons la fiche d'étude de chaque document

Fiche d'étude du document N°1			
Code document : FPR Désignation : feuille de présence Rôle : jour de présence de tous les stagiaires Poste émetteur : Centre de formation Poste récepteur : Conseiller d'orientation Nombre exemplaire : 01			
Description des données			
Désignation	Type	Taille	Observation
Stagiaire	N	8	JJ/MM/AAAA
- Numéro	A	20	
- Nom	A	20	
- Prénom	DATE	10	
- Date de naissance	AN	10	
- Numéro de compte ccp	AN	10	
- Nombre de jours de travail	AN	30	
- Observation	AN	50	
Total		158	

Tableau 2.3.1 Feuille de presence

Fiche d'étude du document N°2			
Code document : LC Désignation : Liste des candidats en formation Rôle : Donner le nombre de candidat dans session Poste émetteur : Centre de formation Poste récepteur : Conseiller d'orientation Nombre exemplaire : 01			
Description des données			
Désignation	Type	Taille	Observation

- session	AN	10	
- Wilaya	AN	20	
- Numéro du candidat	N	6	
- Secteur	A	20	
- Spécialité	A	30	
- Effectif masculin	N	6	
- Effectif féminin	N	6	
- Total	N	10	
Total		108	

Tableau 2.3.2 Liste des candidats en formation

Fiche d'étude du document N°03			
Code document : PVOS			
Désignation : Procès verbale d'ouverture de la section			
Rôle : l'ouverture d'une nouvelle session			
Poste émetteur : C F P A			
Poste récepteur : Conseiller d'orientation			
Nombre exemplaire : 01			
Description des données			
Désignation	Type	Taille	Observation
- Numéro	N	08	
- Date d'effet	Date	10	JJ/MM/AAAA
- Spécialité	A	10	
- Code la section	AN	10	
- Niveau de qualification	A	15	
- Début de formation	Date	10	JJ/MM/AAAA
- Fin de formation	Date	10	JJ/MM/AAAA
- Nombre d'apprenants	N	03	
• Dont les filles	N	03	
• Dont les handicapés	N	03	
• Dont les étrangers	N	03	
- Listes des apprenants			

• Numéro d'inscription	AN	08	JJ/MM/AAAA
• Nom et prénom	A	20	
• Date de naissance	Date	10	
• Lieu de naissance	AN	30	
• Niveau scolaire	A	15	
• Prénom du père	A	15	
• Nom et prénom de la mère	A	30	
• Adresse	A	40	
• Observation	A	60	
Total		313	

Tableau 2.3.3 Procès verbale d'ouverture d'une session

Fiche d'étude du document N°04			
Code document : CFJI			
Désignation : Contrat de formation pour les jeunes âgés de 16 à 20 ans Inscrit au niveau de l'ANEM comme demandeurs d'emplois			
Rôle : installer les étudiants dans des centres de formations			
Poste émetteur : Conseiller d'orientation			
Poste récepteur : AWEM			
Nombre exemplaire : 03			
Description des données			
Désignation	Type	Taille	Observation
-Numéro décrit exécutif	AN	10	JJ/MM/AAAA
-Date décrit exécutif	AN	20	
-Date de la formation professionnelle	Date	10	
	A	20	
-Directeur de l'emploi de la wilaya	A	20	
-Centre de formation C.F.P.A	N	20	
-Directeur du C.F.P.A			
Stagiaire	N	20	JJ/MM/AAAA
-Nom et prénom	Date	10	

-Date de naissance	AN	30	
-Adresse	A	10	
-Daïra			
Total		170	

Tableau 2.3.4 Contrat de formation pour les jeunes âgés de 16 à 20 ans

Fiche d'étude du document N°05			
Code document : CB			
Désignation : Carte bleue			
Rôle : Carte d'inscription			
Poste émetteur : Conseiller d'orientation			
Poste récepteur : Candidat			
Nombre exemplaire : 02			
Description des données			
Désignation	Type	Taille	Observation
Demandeur :			
- Date d'inscription	Date	10	JJ/MM/AAAA
- Numéro d'inscription	AN	08	
- Nom	A	10	
- Prénom	A	10	
- Adresse	AN	30	
- Numéro de téléphone	N	10	
- Adresse électronique	AN	30	
- Situation familial	A	10	
- Services National	A	20	
- Nombre d'enfants	N	02	
- Numéro de sécurité social	AN	10	
- Handicap	A	20	
- Langues maîtrisées	A	10	
- Diplômes	A	10	
- Spécialités	A	10	
- Filière	A	10	

- Durée d'étude	AN	08	JJ/MM/AAAA
- Date de fin d'étude	Date	10	
- Observation	AN	50	
ANEM :			
- Wilaya	A	10	JJ/MM/AAAA
- Numéro	N	08	
- Agence	A	10	
- Date d'inscription	Date	10	
- Nom	A	10	
- Prénom	A	10	
- Adresse	AN	30	
- Fonction	A	15	
- Diplômes et spécialisation	A	20	
Total		403	

Tableau 2.3.5 Carte bleue

Fiche d'étude du document N°06			
Code document : FP			
Désignation : Fiche de pointage			
Rôle : L'assiduité d'un stagiaire			
Poste émetteur : C F P A			
Poste récepteur : Conseiller d'orientation			
Nombre exemplaire : 01			
Description des données			
Désignation	Type	Taille	Observation
- Wilaya	A	10	
- Organisme d'accueil	A	10	
- Adresse	AN	20	
- Numéro de téléphone	N	10	
- Numéro de fax	N	10	
- Email	AN	20	
Fiche pointage			

- Nom	A	10	
- Prénom	A	10	
- Mois	A	10	
- Poste occupé	A	15	
- Année	N	04	
- Jour	N	02	
- Heure d'entrée	AN	10	
- Heure de sortie	AN	10	
Total		151	

Tableau 2.3.6 Fiche de pointage

Fiche d'étude du document N°07			
Code document : LO			
Désignation : Lettre d'orientation			
Rôle : orienter un candidat vers un centre de formation			
Poste émetteur : ANEM			
Poste récepteur : Centre de formation			
Nombre exemplaire : 01			
Description des données			
Désignation	Type	Taille	Observation
- Nom de l'ALEM	A	30	
- Numéro de téléphone	N	10	
- Numéro de la lettre	AN	10	
- Date d'orientation	DATE	10	
-Nom du CFPA	A	30	
-Nom du candidat	A	20	
-Métier	A	30	
-Motifs	A	30	
-Date du lancement de formation	DATE	10	
Total		180	

Tableau 2.3.7 Lettre d'orientation

2.4.3 Circulations des documents au sein des postes de travail

Dans ce qui suit, nous présentons les documents qui sont manipulés par les différents acteurs

Documents entrants	
Au poste	Désignation du document
Conseiller d'orientation	<ul style="list-style-type: none"> - dossier d'inscription. - fiche de pointage. - liste des étudiants - PV d'ouverture de session - Lettre d'orientation
Documents sortant	
Désignation	Vers le poste
Lettre d'orientation	C F P A
PV d'ouverture session	AWEM
Carte bleue	Stagiaire
Dossiers d'inscriptions	AWEM
Contrat	AWEM

Tableau 2.4 liste des documents circulant au sein des postes de travail

2.4.5 Architecture réseau proposée

L'architecture réseau adéquate à notre application est l'architecture à 2-tier, parce que l'AGSA devra faire appel deux serveurs (serveur d'application et celui de gestion de base de données relationnelles).

2.5 Conclusion

A travers ce chapitre, nous avons introduit une brève présentation de l'organisme d'accueil et avons situé notre champ d'étude puis élaboré une étude de l'existant et sa problématique générale. Enfin nous allons proposer des solutions aux insuffisances constatées; entre autre la mise en place d'une application pour le suivie de la gestion des dossiers d'inscription; dans le chapitre suivant nous décrivons la conception de cette dernière.

Chapitre 3 : Analyse et Conception

3.1 Introduction

Afin d'aboutir à une meilleure organisation et à une bonne maîtrise du travail et donc arriver à développer des applications meilleures, il est nécessaire de suivre une démarche méthodologique rigoureuse. Pour cela le choix d'une méthode de conception est d'une importance capitale pour une description rigoureuse de l'application à mettre en place. Pour le développement de notre application de gestion du personnel, nous avons opté pour une démarche de conception orientée objet, en nous basant sur la modélisation UML, que nous présentons ci-après.

3.2 Unified Modeling Language UML

3.2.1 Présentation

UML (*Unified Modeling Language*), que nous pouvant traduire par « Langage de Modélisation Unifié » est un langage graphique de modélisation des données et des traitements. De plus, il constitue une notation permettant de modéliser un problème de façon standard. Il est l'accomplissement de la fusion de précédents langages de modélisation objet : *BOOCH*, *OMT*, *OOSE*, principalement issu des travaux de *Grady Booch*, *James Rumbaugh* et *Ivar Jacobson*. UML est à présent un standard défini par l'Object Management Group (OMG), et est devenu désormais la référence en termes de modélisation objet.

3.2.2 Diagrammes structurels

Ces diagrammes permettent de visualiser, spécifier, construire et documenter l'aspect statique ou structurel du système informatisé.

- **Diagramme de classes** (*Class diagram*): Le but d'un diagramme de classes est d'exprimer de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. Une classe a des attributs, des opérations et des relations avec d'autres classes.
- **Diagramme d'objets** (*Object diagram*): Le diagramme d'objet permet d'éclairer un diagramme de classe en l'illustrant par des exemples. Il montre des objets et des liens entre ces objets (les objets sont des instances de classes dans un état particulier).
- **Diagramme de composants** (*Component diagram*) : il montre les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...). Il montre la mise en œuvre physique des modèles de la vue logique avec l'environnement de développement.

- **Diagramme de déploiement** (*Deployment diagram*) : Ce type de diagramme UML montre la disposition physique des matériels qui composent le système (ordinateurs, périphériques, réseaux...) et la répartition des composants sur ces matériels. Les ressources matérielles sont représentées sous forme de nœuds, connectés par un support de communication.
- **Diagramme des paquetages** (*Package Diagram*) : un paquetage est un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, il sert à représenter les dépendances entre paquetages.
- **Diagramme de structure composite** (*Composite Structure Diagram*) : Le diagramme de structure composite permet de décrire sous forme de boîte blanche les relations entre les composants d'une seule classe.

3.2.3 Diagrammes comportementaux

Les diagrammes comportementaux modélisent les aspects dynamiques du système. Ces aspects incluent les interactions entre le système et ses différents acteurs, ainsi que la façon dont les différents objets contenus dans le système communiquent entre eux.

- **Diagramme des cas d'utilisation** (*Use Case Diagram*) : Les cas d'utilisation sont une technique de description du système étudié selon le point de vue de l'utilisateur. Ils décrivent sous la forme d'actions et de réactions le comportement d'un système. Donc, le diagramme des cas d'utilisation, permet d'identifier les possibilités d'interaction entre le système et les acteurs. Il permet de clarifier, filtrer et organiser les besoins.
- **Diagramme d'activité** (*Activity Diagram*) : Un diagramme d'activité est une variante des diagrammes d'états-transitions. Il permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. dans un diagramme d'activité les états correspondent à l'exécution d'actions ou d'activités et les transitions sont automatiques.
- **Diagramme états-transitions** (*State Machine Diagram*) : permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants. Il est composé d'un ensemble d'états, reliés par des arcs orientés qui décrivent les transitions.

3.2.4 Diagrammes d'interaction

- **Diagramme de séquence** (*Sequence Diagram*) : Il représente séquentiellement le déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs. Le diagramme de séquence peut servir à illustrer un cas d'utilisation.
- **Diagramme global d'interaction** (*Interaction Overview Diagram*) : permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).
- **Diagramme de communication** (*Communication Diagram*) : C'est une représentation simplifiée d'un diagramme de séquence, en se concentrant sur les échanges de messages entre les objets.
- **Diagramme de temps** (*Timing Diagram*) : Le diagramme de temps permet de décrire les variations d'une donnée au cours du temps.

3.3 Démarche suivie

UML permet de représenter des modèles, mais il ne définit pas de processus, d'élaboration de modèle. Les acteurs d'UML conseillent tout de même une démarche pour favoriser la réussite d'un projet, cette démarche doit être :

- **Une démarche itératif et incrémentale :**
Pour comprendre et représenter un système complexe, pour analyser par étape, pour favoriser le prototype pour réduire et maîtriser l'inconnu.
- **Une démarche guidée pour les besoins des utilisateurs :**
Tout est basé sur le besoin des utilisateurs du système, le but du développement lui-même est de répondre à leur besoin. Chaque étape sera affinée et validée en fonction des besoins des utilisateurs.
- **Une démarche centrée sur l'architecture logicielle :**
C'est la clé de succès d'un développement, les choix stratégiques définiront la qualité du logiciel.

La figure ci-dessous montre la représentation graphique de la démarche de modélisation que nous avons choisie pour concevoir notre application.

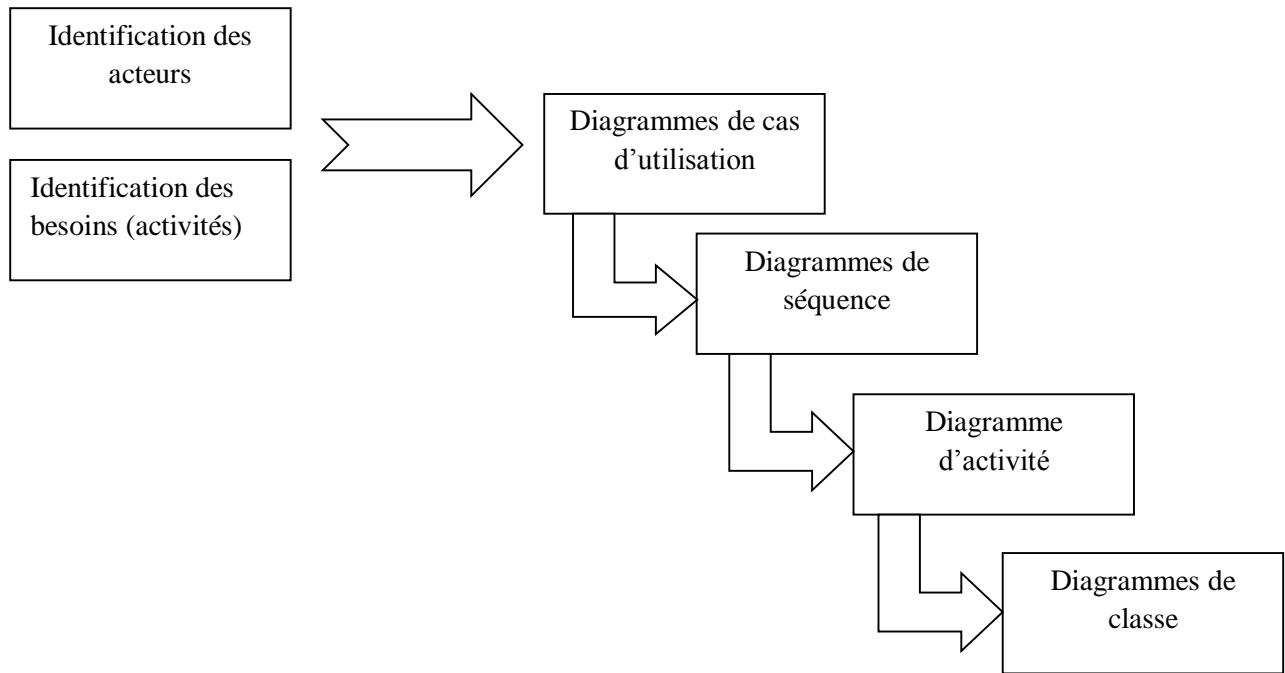


Figure 3.1 Schéma de la démarche suivie

3.3.1 Expression des besoins

Notre projet porte sur la mise en place d'une application client/serveur n-tiers pour l'automatisation des activités de l'AGSA dans le but de maîtriser le suivi des mouvements des documents et dossiers entre les différents services du champ d'étude, l'application doit offrir un environnement interactif et convivial aux différents acteurs des service dispositifs et demandeur, leur permettant de travailler et de communiquer mutuellement, dans l'objectif de la gestion, et ce via un réseau local (intranet).

3.3.2 Identification des acteurs

Un acteur est une entité, humaine ou matérielle, intervenant dans le système d'information. Les acteurs se divisent en deux catégories, internes et externes, selon leur appartenance ou non à l'agence.

Les principaux acteurs de notre système comme nous les avons cité dans le chapitre précédent sont :

- Conseiller d'orientation (Administrateur)
- Candidat.

3.3.3 Choix des diagrammes UML

Nous avons choisi pour la partie statique ou structurelle les diagrammes de contexte et de classe pour représenter les différentes classe intervenantes dans notre système, pour la partie

comportementale nous avons opté pour les diagrammes des cas d'utilisation et d'activé pour identifier les différentes interactions entre le système et les différents acteurs et pour décrire sous forme de flux le comportement de l'AGSA et pour finir nous avons choisi d'élaborer quelques diagrammes de séquences pour la partie dynamique à fin de faire une représentation séquentielle du déroulement des traitements.

3.3.4 Analyse

Dans cette partie, nous allons spécifier d'une manière détaillée et précise les interactions significatives (interactions entre le système et les acteurs, interaction entre les objets) du point de vue de l'application. Pour cela, nous devons d'abord recenser les objectifs et besoins de notre application.

3.3.5 Conception

Le but de la conception est de rendre le modèle réalisable sous forme de logiciel. Après avoir recensé les cas d'utilisation relatifs à chaque acteur de notre système, nous allons établir les diagrammes de séquence et d'activité suivie des diagrammes de classes et le modèle logique de données.

3.4 Analyse

3.4.1 Diagramme de contexte

Pour aborder le premier point, nous allons utiliser un diagramme de contexte pour modéliser le système globalement parlant.

Le diagramme de contexte permet de :

- Mettre en évidence les principaux acteurs ;
- Définir les interactions entre le système modélisé et les différents acteurs.

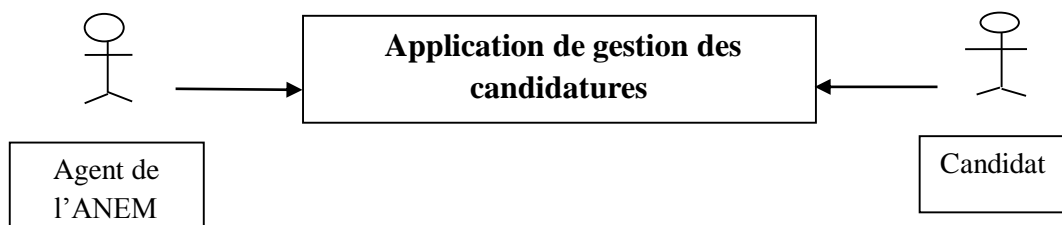


Figure 3.2. Diagramme de contexte.

3.4.2 Tâches et scénarios

Nous vous montrons dans le tableau ci-dessous les différentes tâches et scénarios que peut effectuer l'agent de l'ANEM sur AGSA ainsi que celle qu'il peut effectuer le candidat sur l'application Web

Acteurs	Tâches
Agent de l'ANEM	T0 : S'authentifier. T1 : Accéder à son espace. T2 : Gérer les candidats. T3 : Gérer les contrats. T4 : Gérer les sessions. T5 : Changer son mot de passe. T6 : Se déconnecter.
Candidat	T7 : Faire une préinscription T8 : Prendre un Rendez-vous T9 : S'authentifier. T10 : Accéder à son espace. T11 : Consulter l'état de suivi de son dossier. T12 : Se déconnecter

Tâches	Scénarios
T0 : S'authentifier	S0: Accéder au formulaire. S1 : Saisir le login et le mot de passe. S2 : Valider le login et le mot de passe.
T1 : Accéder à son espace	S3 : Cliquer sur le bouton « Candidat ».
T2 : Gérer les candidats	S4: Ajouter un candidat. S5 : Modifier un candidat. S6 : Supprimer un candidat.
T3 : Gérer les contrats	S7 : Etablir un contrat. S8 : Modifier contrat. S9 : Annuler un contrat.
T4 : Gérer les sessions.	S10 : Ajouter une nouvelle session S11 : Modifier les sessions

	S12 : Supprimer la session
T5 : Changer son mot de passe	S13 : Cliquer sur changer mot de passe. S14 : Remplir le formulaire. S15 : Valider le formulaire.
T6 : Se déconnecter	S16 : Cliquer sur « déconnexion ».
T7 : Faire une préinscription	S16 : Cliquer sur Se préinscrire S17 : Remplir le formulaire S18 : Valider le formulaire
T8 : Prendre un rendez-vous	S19 : Cliquer sur Rendez-vous S20 : Choisir une date
T9 : S'authentifier	S21: Accéder au formulaire. S22 : Saisir le login et le mot de passe. S23 : Valider le login et le mot de passe.
T10 : Accéder à son espace	S24 : Cliquer sur le bouton « Consulter son dossier »
T11 : Consulter l'état de son dossier	S25 : Cliquer sur Consulter son dossier
T12 : Se déconnecter	S26 : Cliquer sur « déconnexion ».

Tableau 3.1 Liste des tâches et scénarios

3.4.2 Diagramme des cas d'utilisation pour les tâches

Dans cette section, nous allons définir le diagramme de cas d'utilisation relatif à chaque acteur. Pour ainsi établir les diagrammes de cas d'utilisation relatifs à chacun de ces derniers.

Acteurs	Cas d'utilisation
Agent de l'ANEM	<ul style="list-style-type: none"> – Authentification – Gestion des candidats – Gestion des contrats – Gestion des sessions – Gestion des organismes d'accueil
Stagiaire	<ul style="list-style-type: none"> – Authentification – Faire une pré-inscription

	<ul style="list-style-type: none">– Prendre un rendez-vous pour finaliser son inscription– Vérifier l'état de son dossier
--	--

Figure.3.2.Tableau représentant les différents cas d'utilisation

Description graphique des cas d'utilisations

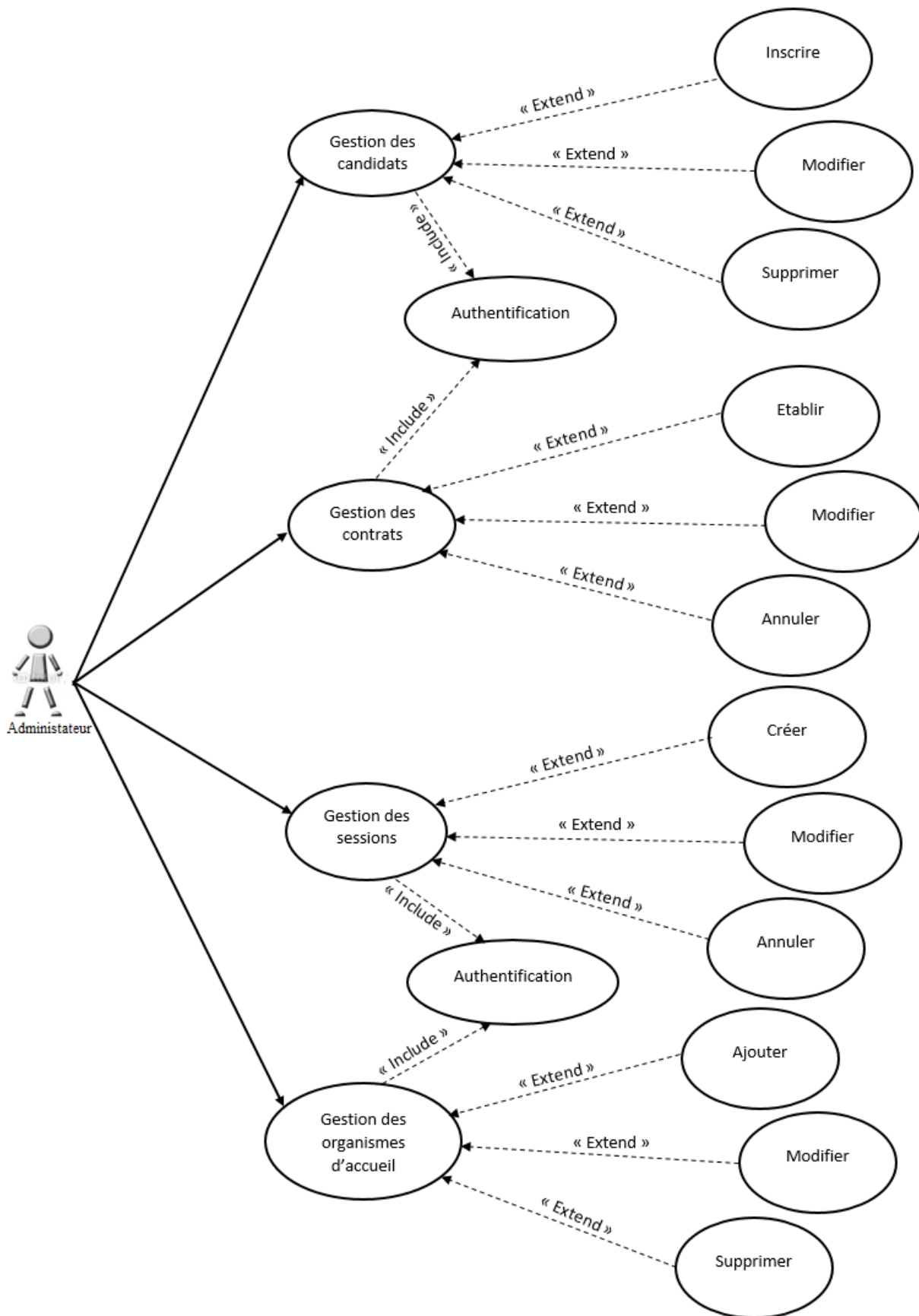


Figure.3.3. Diagramme de cas d'utilisation pour l'acteur Conseiller d'orientation

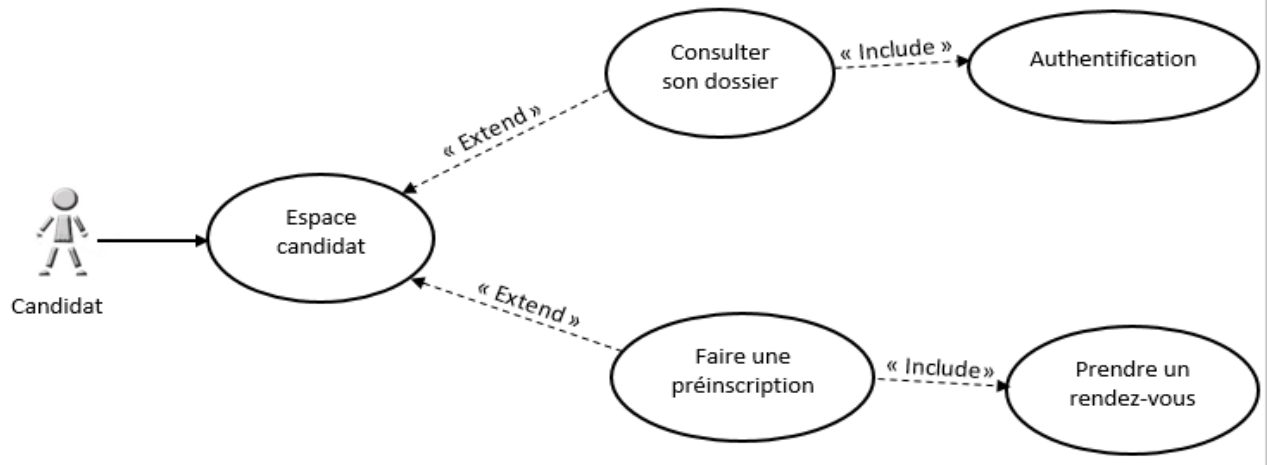


Figure.3.4. Diagramme de cas d'utilisation pour l'acteur Candidat

Description textuelle des cas d'utilisation

Vue le nombre important de cas d'utilisation, nous allons décrire seulement un cas d'utilisation par acteur :

Cas d'utilisation : « **établissement d'un contrat** »

Acteur : Agent de l'ANEM.

Description : Etablir un contrat.

Conséquence : le stagiaire possède un contrat.

Séquence d'événement :

1. L'agent de l'ANEM s'authentifie pour accéder à son espace,
2. Il demande la page d'établissement d'un stagiaire,
3. Le système retourne le formulaire correspondant,
4. L'agent de l'ANEM remplit les champs appropriés,
5. Il clique sur le bouton ajouter,
6. Le système affiche une page de confirmation.

Cas d'utilisation : « **faire une pré-inscription** »

Acteur : Stagiaire

Description : faire une pré-inscription.

Conséquence : le stagiaire s'inscrit via le net.

Séquence d'événement :

1. Il demande la page de pré-inscription,

2. le système retourne le formulaire,
3. le stagiaire remplit les champs appropriés,
4. Il clique sur le bouton confirmer,
5. Le système affiche une page de confirmation.

3.5 Conception

3.5.1 Diagrammes de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation *Unified Modeling Language*.

Nous vous exposant quelques diagrammes d'AGSA.

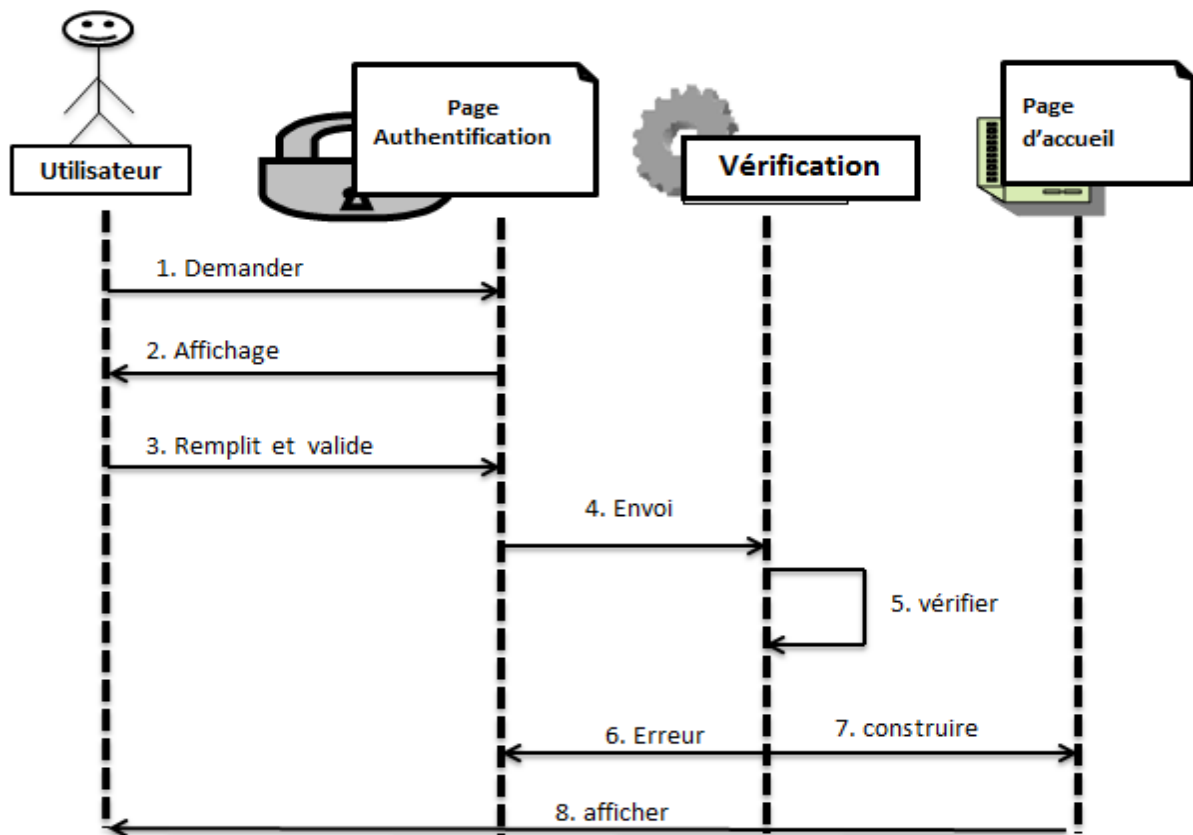
Diagramme de séquence du cas d'utilisation "Authentification"

Figure. 3.5: Diagramme de séquence «l'authentification»

1. l'utilisateur demande le formulaire d'authentification
2. l'application affiche le formulaire d'authentification
3. l'utilisateur saisit le mot de passe
4. l'envoi de mot de passe pour vérification
5. le système vérifie la validité du mot de passe
6. affiche un message d'erreur si le mot de passe est incorrect
7. construire un message de confirmation
8. l'application affiche la page d'accueil

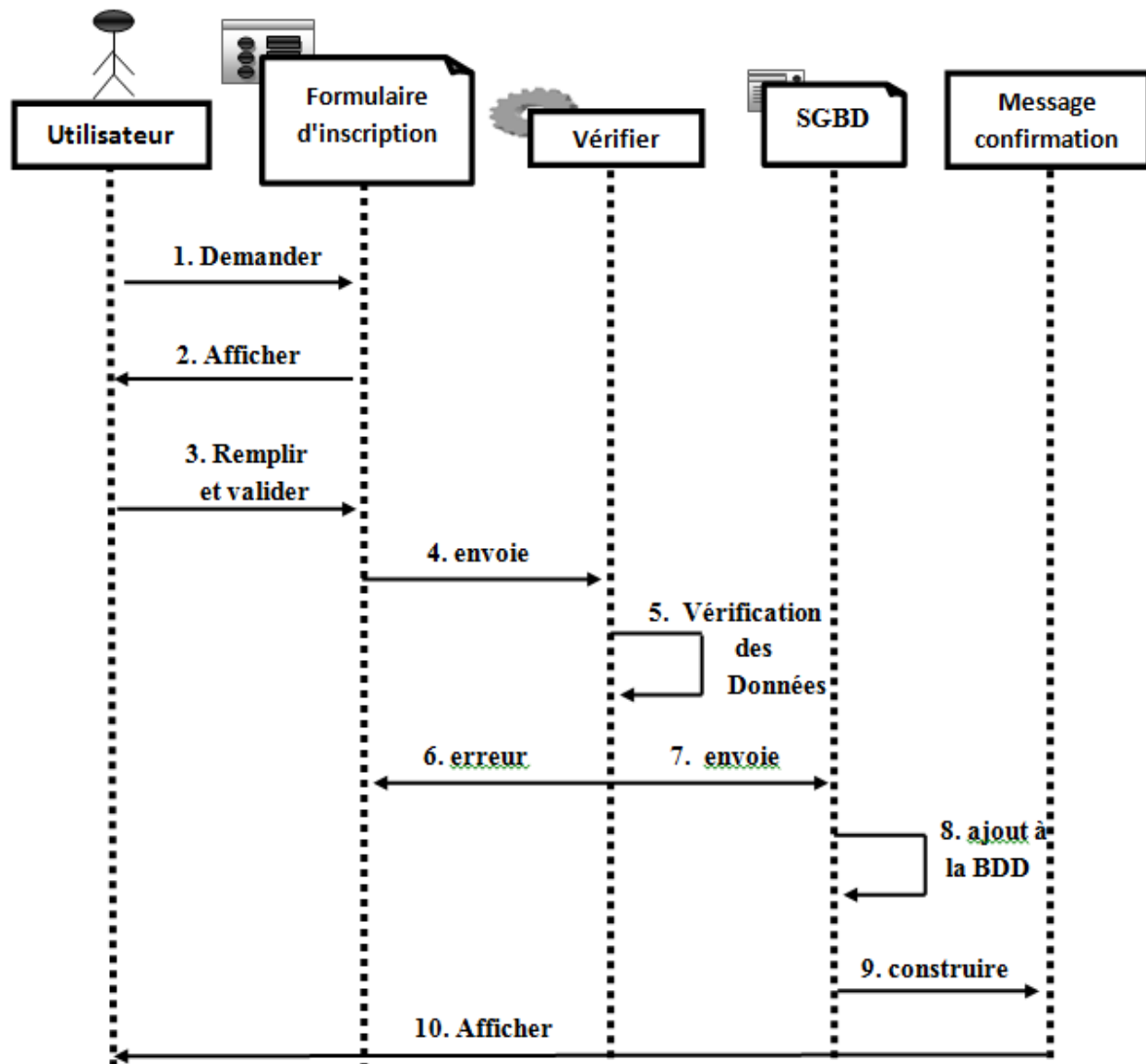
Diagramme de séquence du cas d'utilisation "Inscription"

Figure. 3.6 Diagramme de séquence «inscription d'un stagiaire»

1. l'utilisateur demande le formulaire d'inscription d'un stagiaire
2. affichage du formulaire d'inscription
3. l'utilisateur saisit les données et valide
4. envoi des données pour vérification
5. control des données
6. afficher un message d'erreur
7. envoi de requête pour stocker les données

8. ajout des données dans la BDD
9. construire un message de confirmation
10. afficher la confirmation d'enregistrement

Diagramme de séquence du cas d'utilisation "Etablissement contrat"

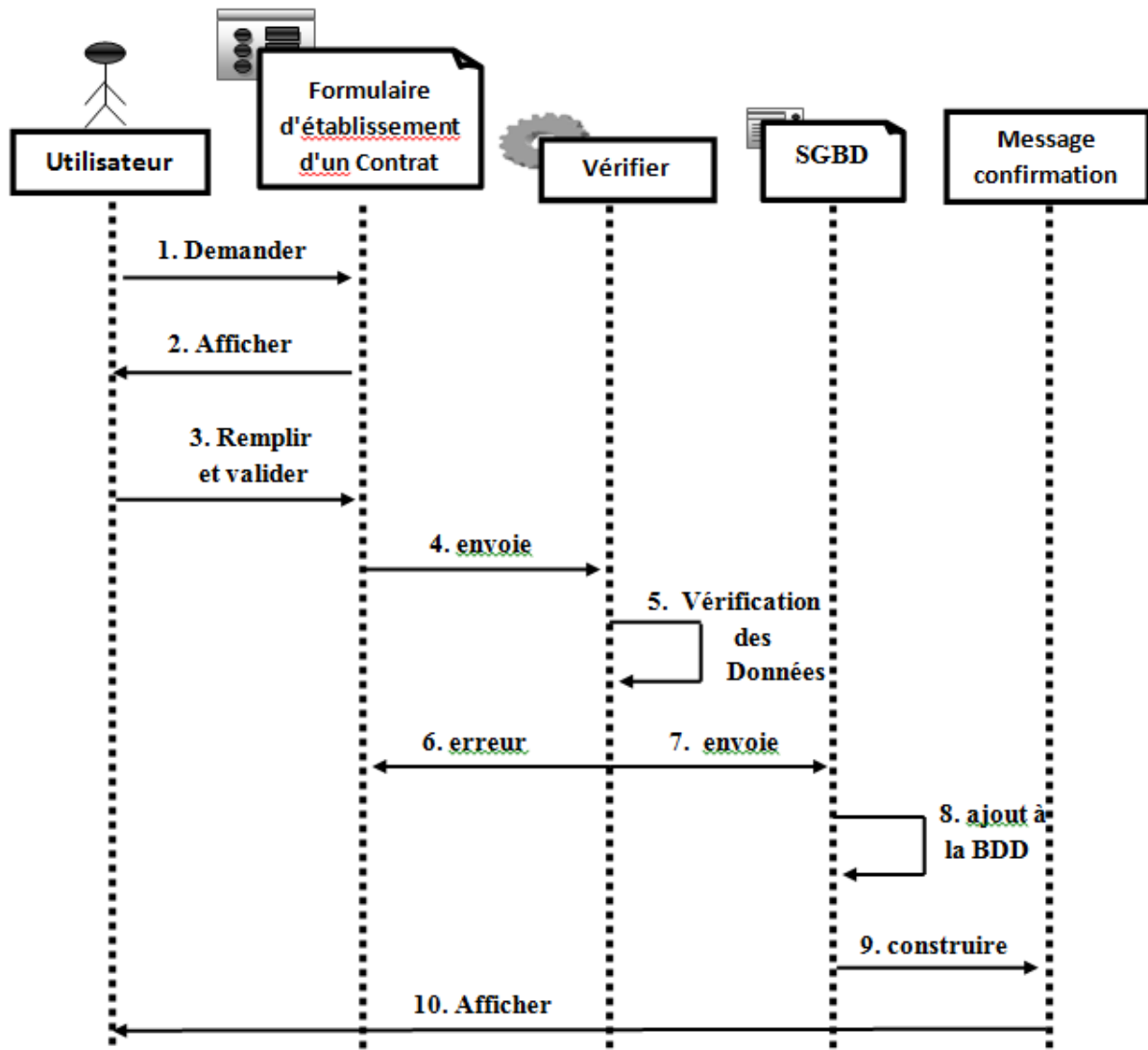


Figure. 3.7 Diagramme de séquence «établissement d'un contrat»

1. l'utilisateur demande le contrat
2. affichage du contrat
3. l'utilisateur saisit les données et valide
4. envoi des données pour vérification
5. control des données

6. afficher un message d'erreur
7. envoi de requête pour stocker les données
8. ajout des données dans la BDD
9. construire un message de confirmation
10. afficher la confirmation d'enregistrement

Diagramme de séquence du cas d'utilisation "Mise à jour"

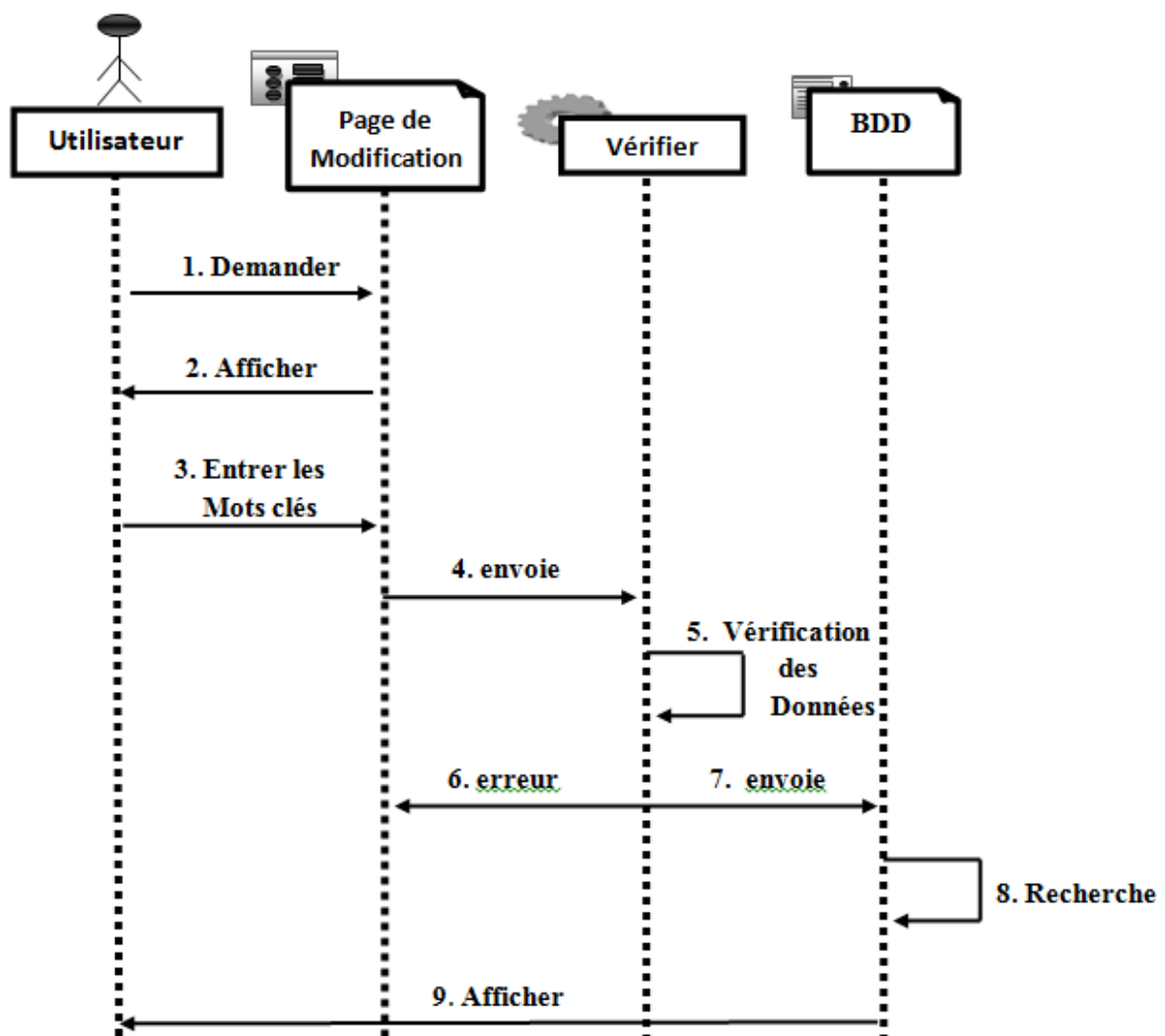


Figure. 3.8 Diagramme de séquence «mise à jour»

1. l'utilisateur demande le formulaire de modification
2. l'application affiche le formulaire
3. l'utilisateur saisit les mots clé

4. l'envoi des données pour vérification
5. control des données
6. envoie de requête pour chercher les données
7. afficher un message d'erreur si données non trouvé
8. recherche des données dans la BDD
9. afficher les données si trouvées sinon afficher un message d'erreur

3.5.2 Diagramme d'activité

Est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables. Il est également utilisé pour décrire un flux de travail (*workflow*). Nous vous présentons quelques diagrammes d'AGSA

Diagramme d'activité du cas d'utilisation "Authentification"

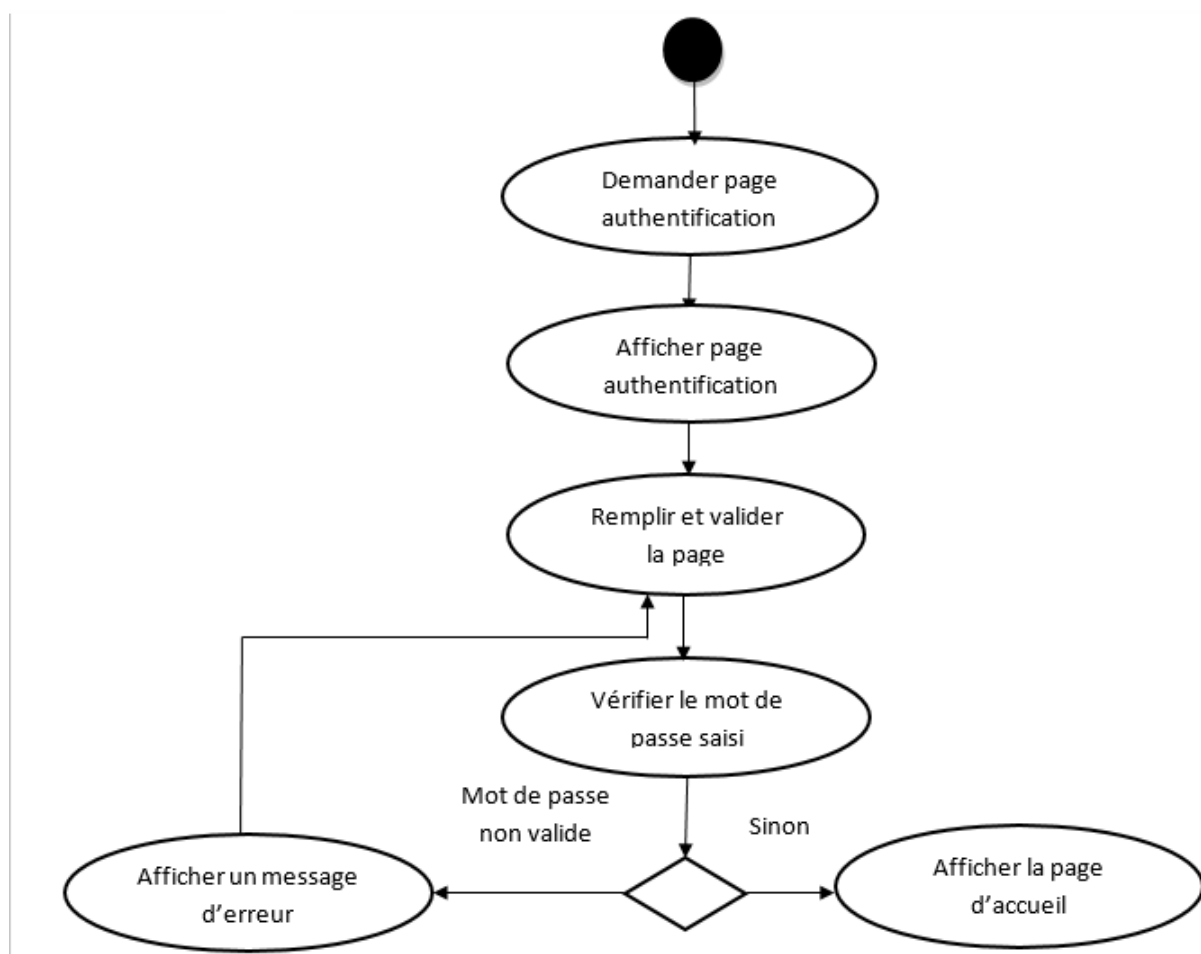


Figure 3.8 Diagramme d'activité du cas d'utilisation « authentification »

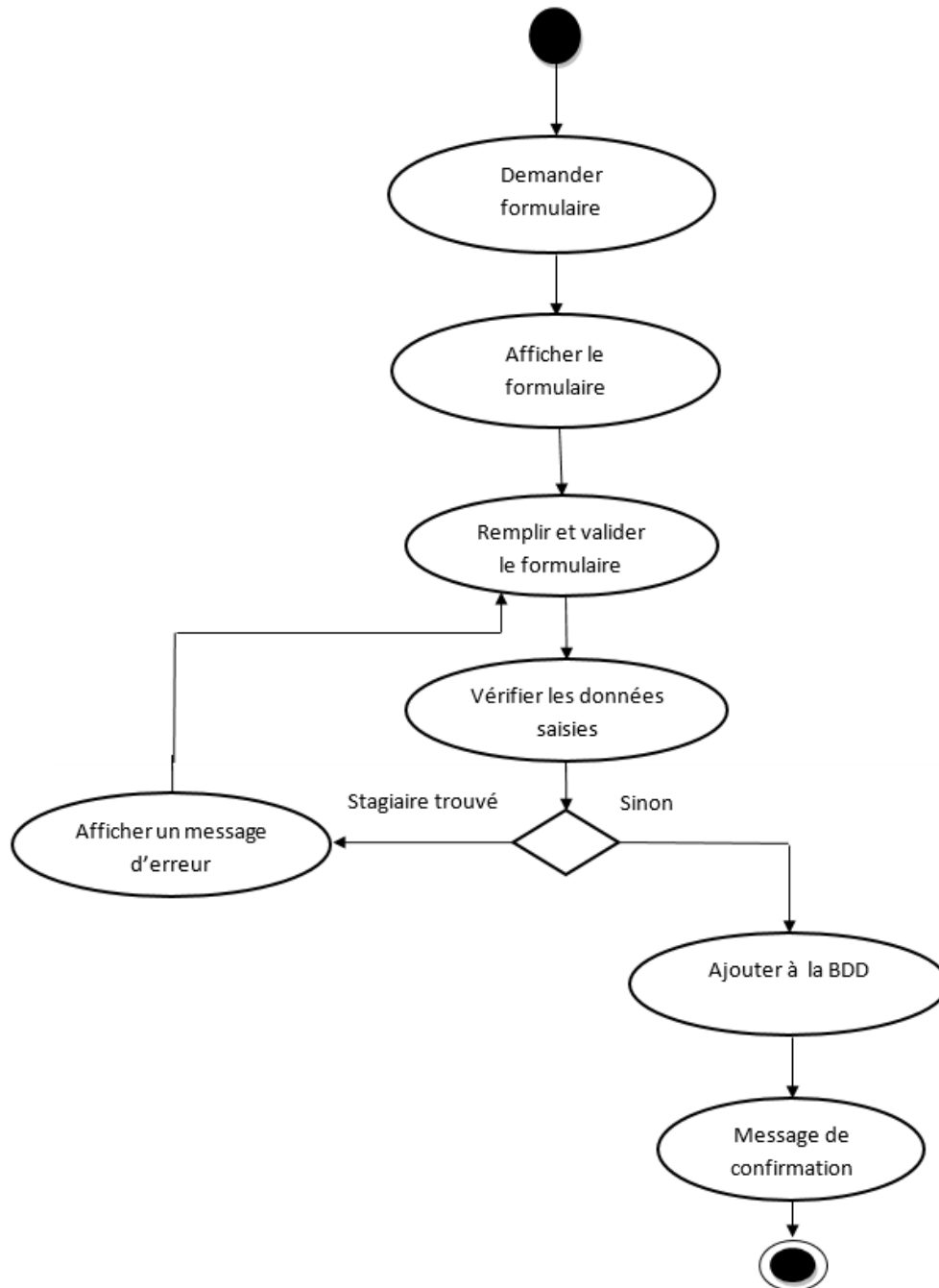
Diagramme d'activité du cas d'utilisation "Inscription"*Figure 3.9 Diagramme d'activité du cas d'utilisation « inscription »*

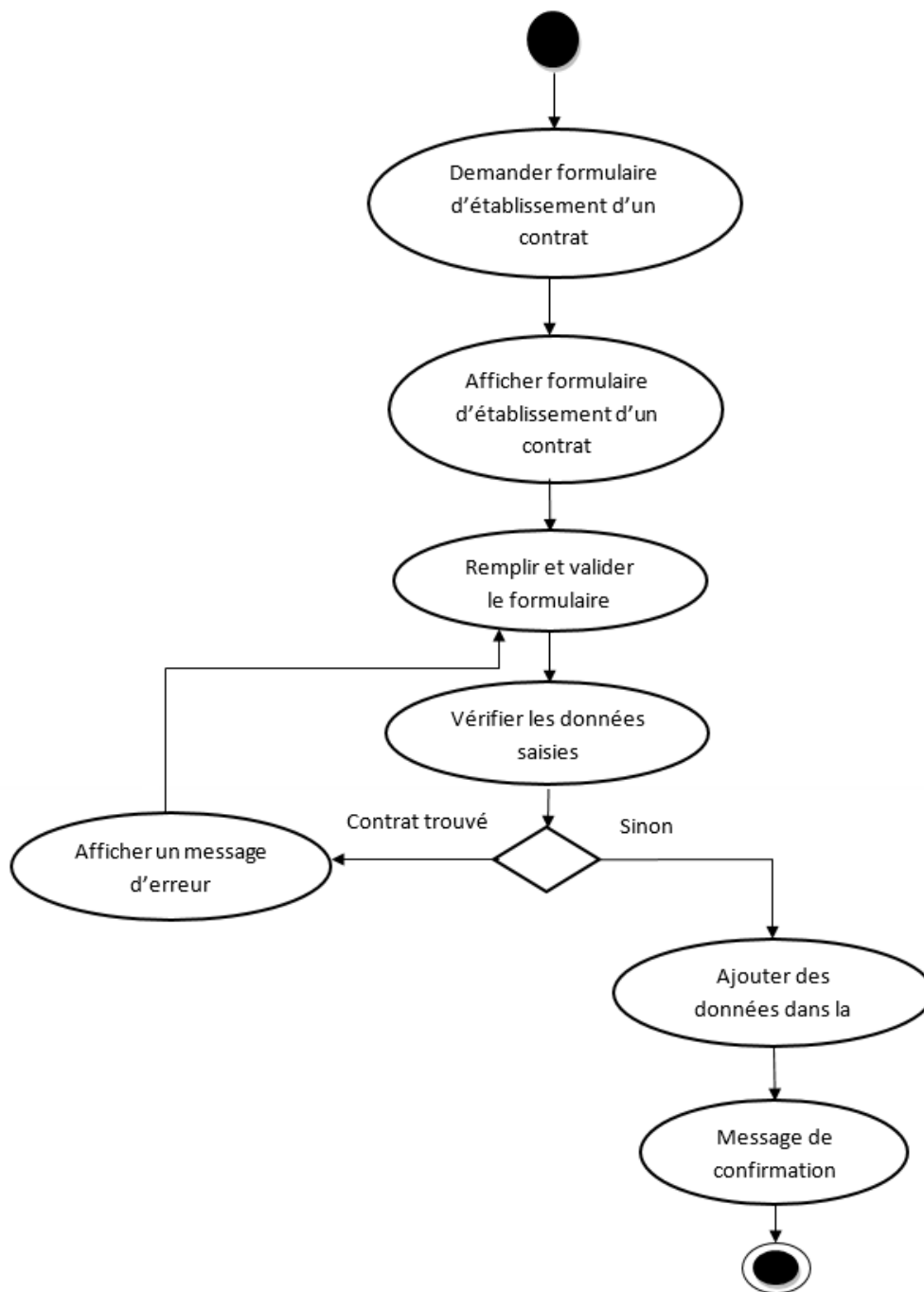
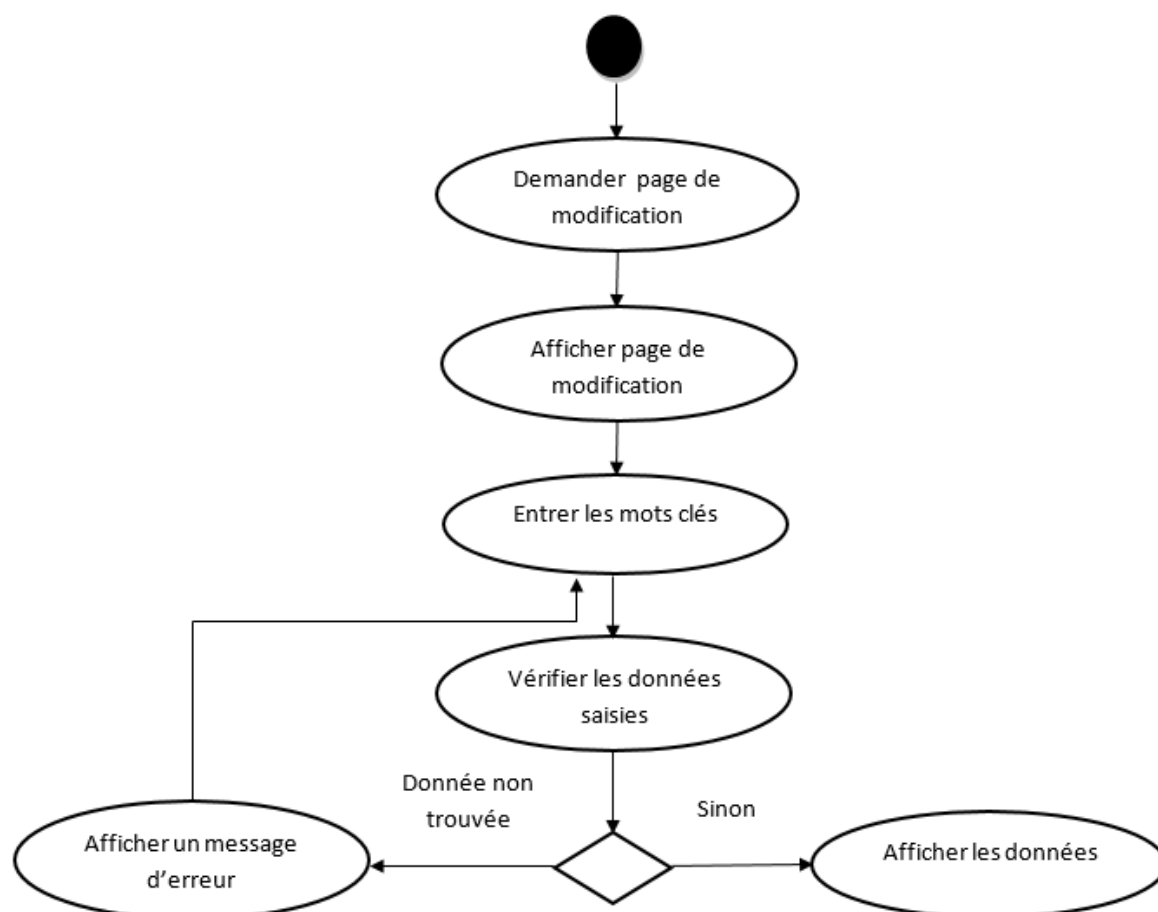
Diagramme d'activité du cas d'utilisation "Etablissement contrat"*Figure 3.10 Diagramme d'activité du cas d'utilisation « Etablissement d'un contrat »*

Diagramme d'activité du cas d'utilisation "Mise à jour"*Figure 3.11 Diagramme d'activité du cas d'utilisation « Mise à jour »*

3.5.3 Diagrammes de classe

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Nous vous exposant quelques diagrammes de l'AGSA

Diagramme de classe du cas d'utilisation "Authentification"

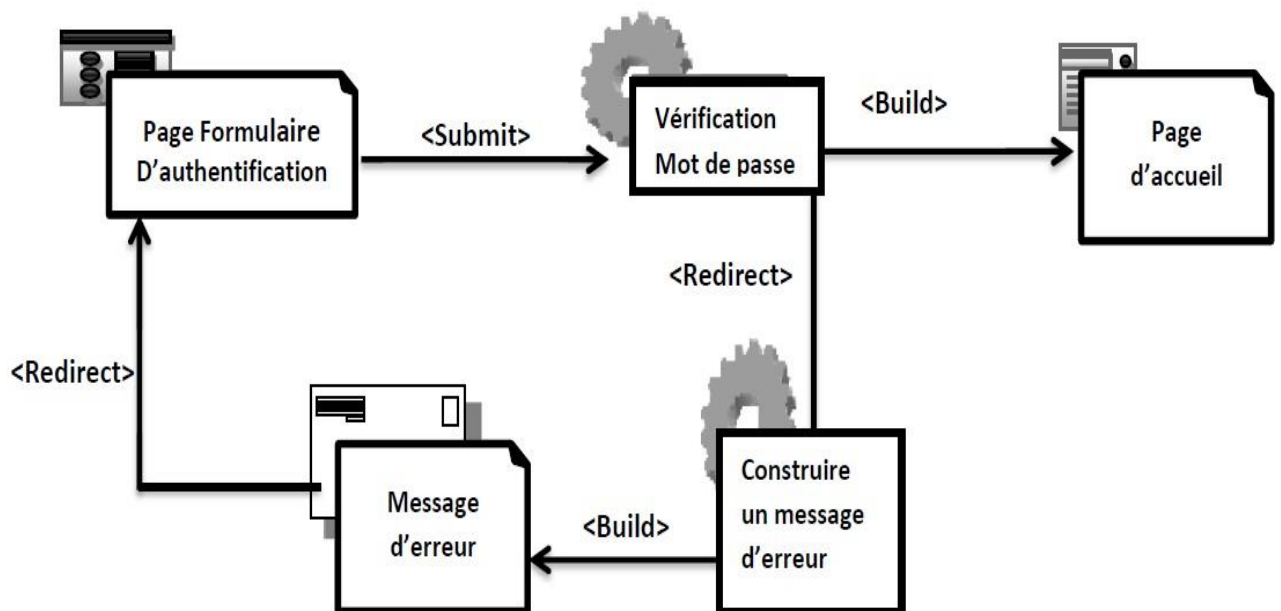


Figure 3.12 Diagramme de classe du cas d'utilisation « authentification »

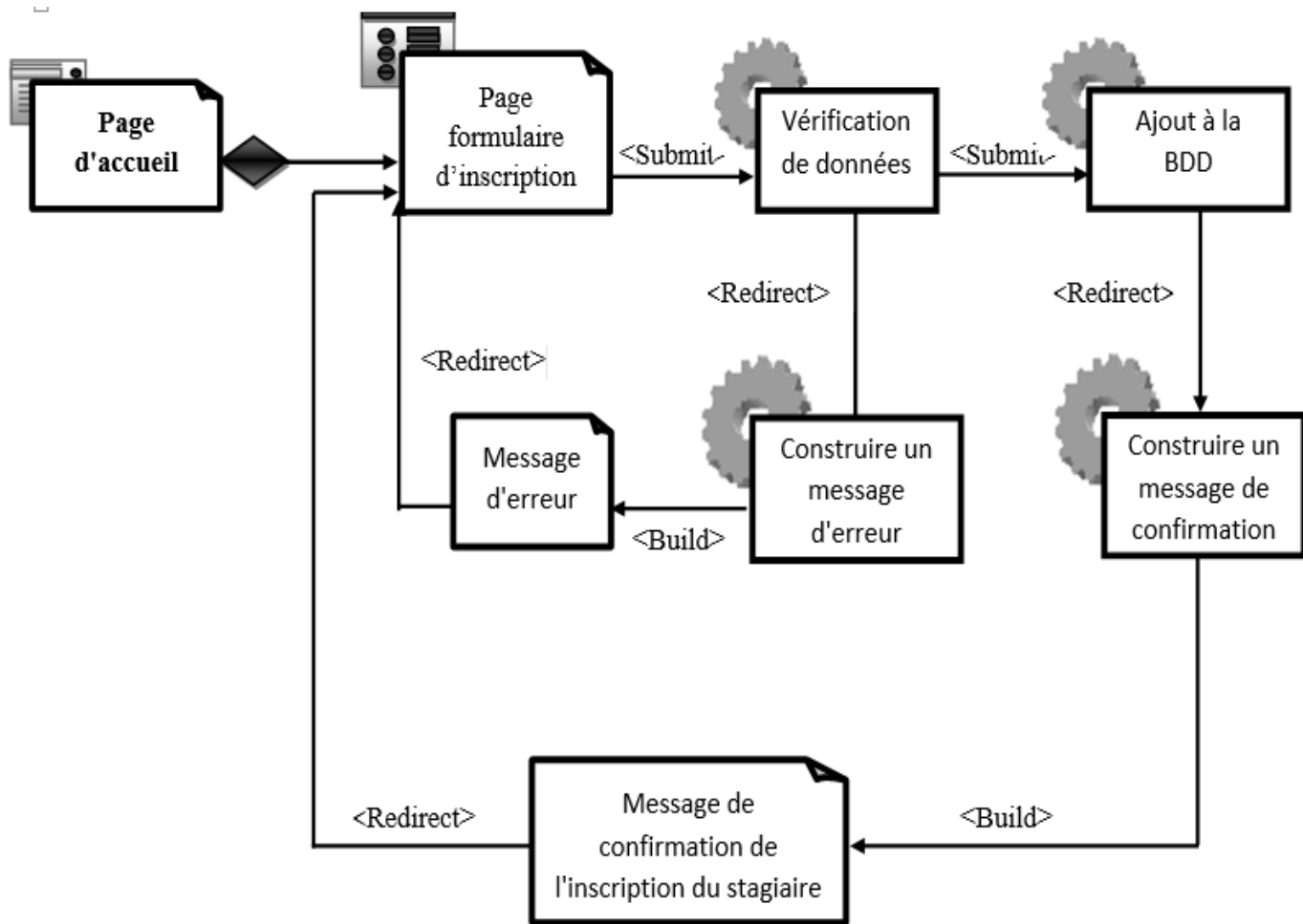
Diagramme de classe du cas d'utilisation "Inscription"*Figure 3.12 Diagramme d'activité du cas d'utilisation « Inscription »*

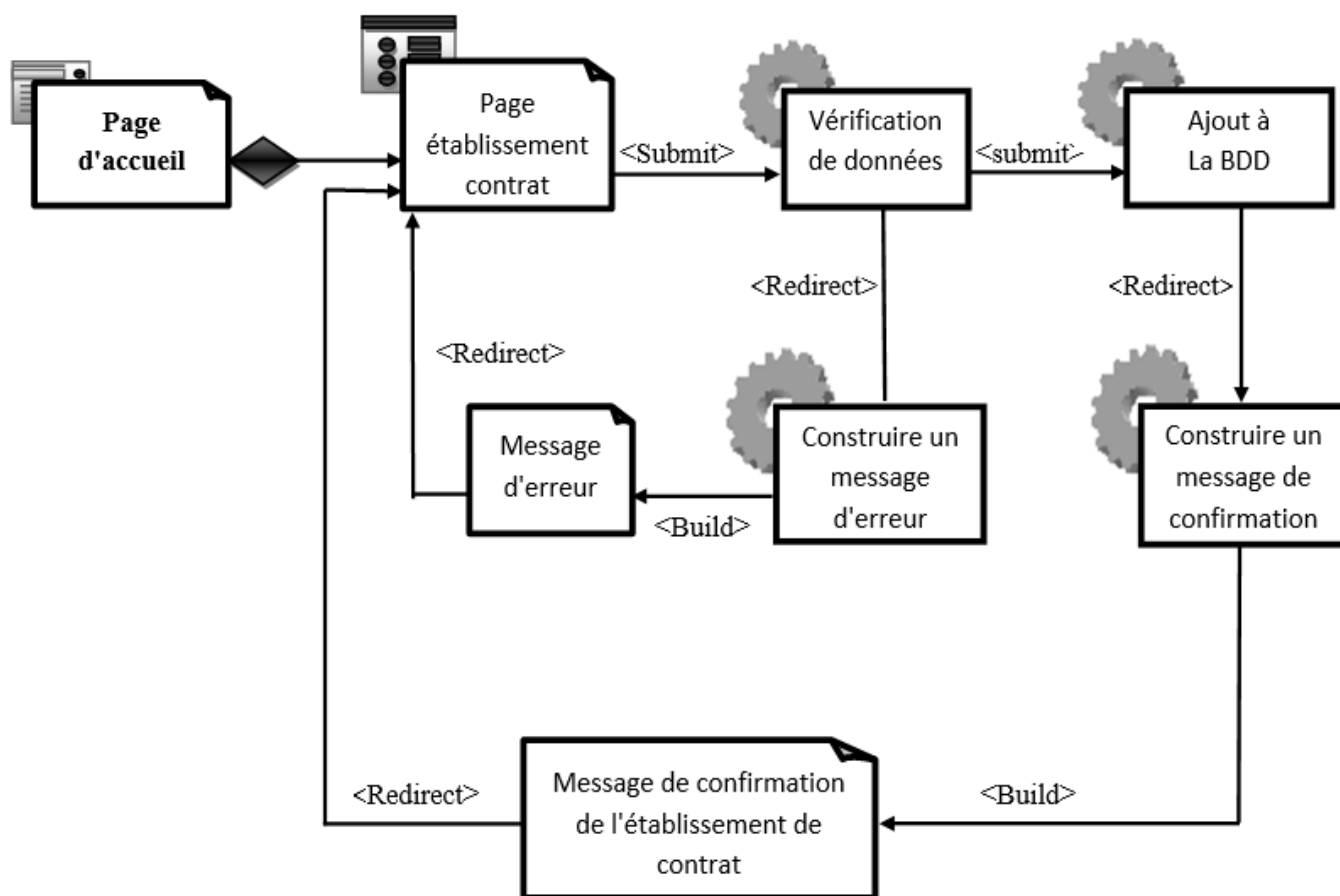
Diagramme de classe du cas d'utilisation "Etablissement contrat"

Figure 3.13 Diagramme d'activité du cas d'utilisation « Etablissement d'un contrat »

Diagramme de classe du cas d'utilisation "Mise à jour"

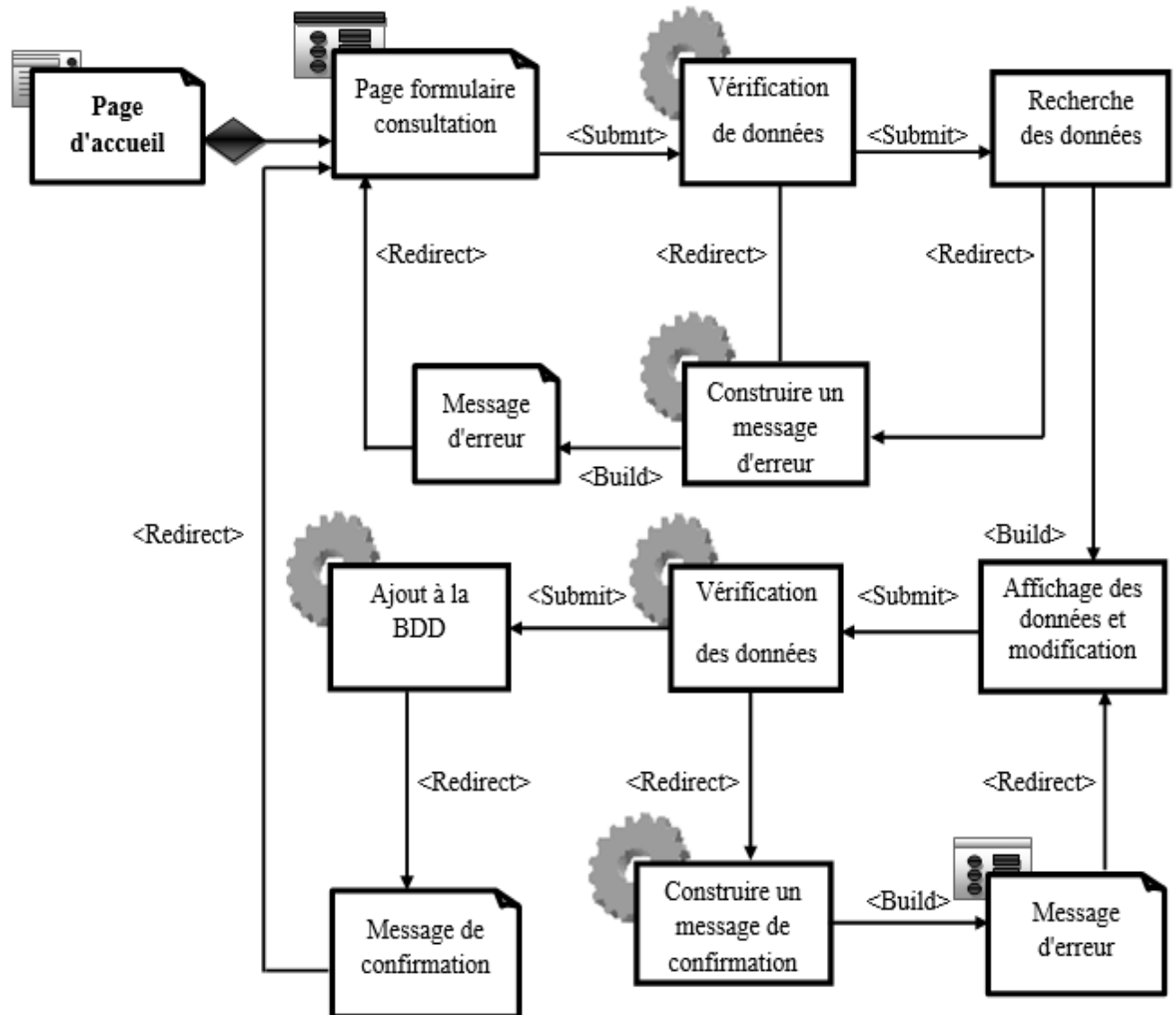


Figure 3.14 Diagramme d'activité du cas d'utilisation « mise à jour »

3.5.4 Diagramme classe global

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Les règles de gestion du diagramme de classes

Ces règles vont nous permettre d'établir le diagramme de classes :

- Plusieurs candidats peuvent maîtriser plusieurs langues.
- Un candidat peut avoir un contrat.
- Un ou plusieurs candidats appartiennent à une seule daïra.
- Une ou plusieurs Agences ANEM sont liés à une seule daïra
- Une ou plusieurs formations peuvent avoir plusieurs sessions.
- Une ou plusieurs formations appartiennent à Un ou plusieurs centres de formations.
- Un ou plusieurs contrats sont liés à un seul centre de formation
- Un ou plusieurs contrats sont liés à un seul organisme d'accueil.
- Plusieurs contrats sont pointés dans plusieurs dates.

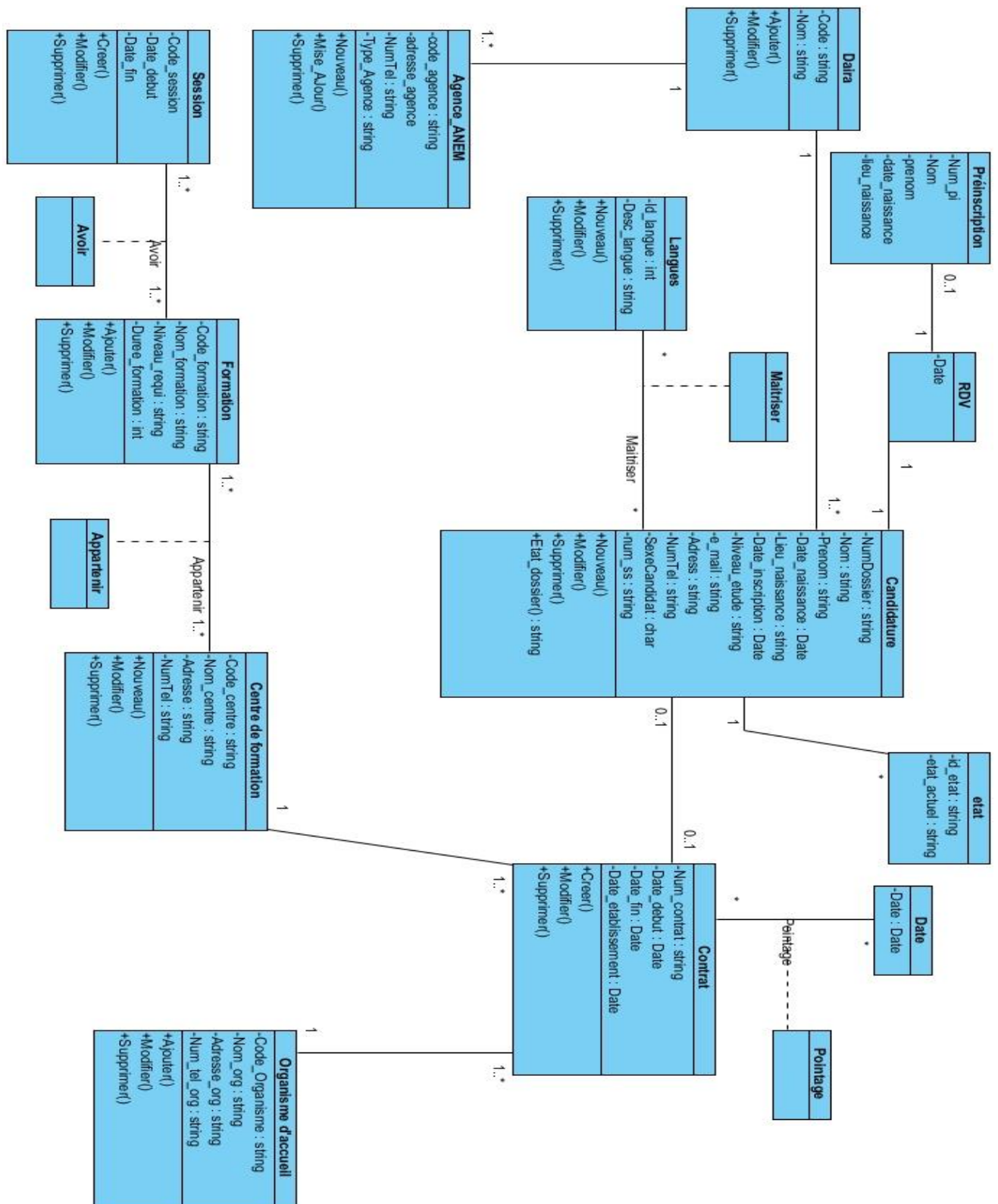


Figure 3.15 Diagramme de classe globale

3.5.5 Modèle logique de données

Nous avons suivi les règles suivantes pour le passage au modèle logique de données :

- La classe devient une table.
- L'attribut d'une classe devient une colonne dans une table.
- Association à ordre 0 ou 1 - * : clé étrangère dans la table ayant * comme terminaison d'association.
- Association à ordre * - * : la classe d'association devient une table contenant deux clés étrangères.

Le modèle logique de données

- Candidat (Num_dossier, Nom, Prénom, Date_naissance, Lieu_naissance, Date_inscription, Niveau_etude, e_mail, commentaire, adresse, Num_tel, Sexe_candidat, Num_ss, Code_daira*)
- Préinscription (num_pi, nom, prenom, date_naissance, lieu_naissance)
- Daïra (Code_daira, Nom)
- Langues (Id_langue, Desc_langue)
- Agence_ANEM (Code_agence, Adresse_agence, Num_tel, Type_agence, Code_daira*)
- Contrat (Num_contrat, Date_debut, Date_fin, Date_etablissement, _Num_dossier*, code_organisme*, code_centreformation*, code_agence*)
- Organisme d'accueil (Code_org, Nom_org, adresse_org, Num_tel_org)
- Formation (Code_formation, Nom_formation, Niveau_requis, Duree_formation)
- Centre de formation (Code_centre, Nom_centre, Adresse, Num_tel, Num_contrat*)
- Session (Code_session, Date_debut, Date_fin)
- Date(Date)
- Appartenir (Code_centre*, Code_formation*)
- Pointage(Num_contrat*, Date*)
- Maitriser (Id_langue*, Num_dossier*)
- Avoir(Code_formation*, Code_session*)
- Etat (Id_etat, etat_actuel, num_dossier*)
- RDV (code_rdv, date_rdv).

Légende

- Attribut : Clé primaire.
- Attribut : Clé étrangère*.

3.6 Conclusion

Dans ce chapitre nous avons suivi une démarche de modélisation pour développer notre application, en se basant sur l'UML. Il a été scindé en trois principales parties : l'étude préliminaire, l'étude conceptuelle statique et pour finir l'étude conceptuelle dynamique.

Le chapitre suivant sera consacré à la partie réalisation de notre application et sa mise en œuvre en utilisant la technologie des web services.

Chapitre 4 : Mise en œuvre

4.1 Introduction

Dans ce chapitre, nous allons présenter notre plateforme de développement et les outils utilisés pour mener à terme la réalisation de notre application ainsi que, quelques interfaces du logiciel.

4.2 Quelques Langages de programmation

4.2.1 Python

Est un largement utilisé à des fins générales , langage de programmation de haut niveau . Sa philosophie de conception souligne Code de lisibilité , et sa syntaxe permet aux programmeurs d'exprimer des concepts en moins de lignes de code que ce ne serait possible dans les langues tels que C .Le langage fournit des constructions destinées à permettre des programmes clairs à la fois sur une petite et à grande échelle.

Python supporte plusieurs paradigmes de programmation, y compris orienté objet , impératif et la programmation fonctionnelle ou de procédure styles. Il dispose d'un type dynamique et système automatique de gestion de mémoire et a une grande et complète la bibliothèque standard .

Comme d'autres langages dynamiques , Python est souvent utilisé comme un langage de script , mais est également utilisé dans un large éventail de contextes non-script. En utilisant des outils tiers, tels que py2exe ou Pyinstaller , du code Python peut être emballé dans des programmes exécutables autonomes. Interprètes Python sont disponibles pour plusieurs systèmes d'exploitation.

CPython , la mise en œuvre de référence de Python, est libre et logiciel open source et dispose d'un modèle de développement basé sur la communauté, comme ne presque toutes ses implémentations alternatives. CPython est géré par le but non lucratif *Python Software Foundation* .

4.2.2 Java

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp).

4.2.3 C++

Le C++ est un langage de programmation permettant la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation orientée objet et la programmation générique. Le langage C++ n'appartient à personne et par conséquent n'importe qui peut l'utiliser sans besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation. C++ est l'un des langages de programmation les plus populaires, avec une grande variété de plateformes matérielles et de systèmes d'exploitation.

4.2.4 Delphi

Delphi est à la fois un langage de programmation orienté objet (POO) et un environnement de développement intégré. Édité par la société Embarcadero (anciennement Borland puis CodeGear), il se pose en alternative au langage Visual Basic en proposant un développement à la fois très rapide et de grande qualité.

Nous avons opté pour le Delphi parce qu'il possède plusieurs avantages qui améliorent la productivité, contrôle strict du compilateur pour éviter les erreurs de mémoire, de débordement, gestion intégrée des chaînes de caractères et des tableaux dynamiques mais aussi pour sa réputation de temps record de compilation.

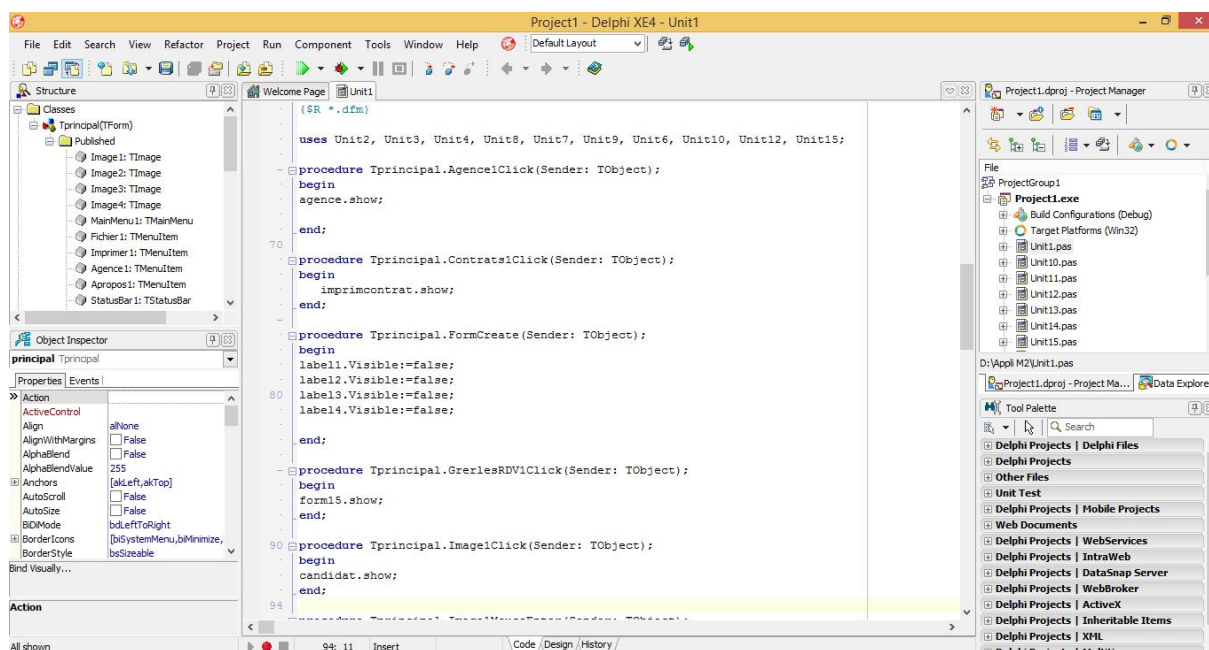


Figure 4.1 Fenêtre Delphi XE4

Exemple d'un fragment de code en Delphi

```
procedure TForm15.SpeedButton1Click(Sender: TObject);  
begin  
  with dm.ajrdv do  
  begin  
    parameters.ParamByName('@daterdv').Value:=datetimepicker1.Date;  
    parameters.ParamByName('@hrdv').Value:=edit1.Text;  
    parameters.ParamByName('@momrdv').Value:=datetostr(datetimepicker1.Date)+' à '+edit1.Text;  
    parameters.ParamByName('@nbplace').Value:=strtoint(edit2.Text);  
    execproc;  
    showmessage('Rendez-vous crée avec succès');  
  end;  
end;
```

4.3 Quelques outils de développement web**4.3.1 Eclipse**

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java. Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi D'AGL recouvrant modélisation, conception, testing, gestion de configuration, reporting... Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio. Bien qu'Eclipse ait d'abord été conçu uniquement pour produire des environnements de développement, les utilisateurs et contributeurs se sont rapidement mis à réutiliser ses briques logicielles pour des applications clientes classiques. Cela a conduit à une extension du périmètre initial d'Eclipse à toute production de logiciel : c'est l'apparition du framework Eclipse RCP en 2004.

4.3.2 Netbeans

Est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL et GPLv2 (*Common Development and Distribution License*). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML.

4.3.3 Dreamweaver

Dreamweaver est un éditeur de site web WYSIWYG pour Microsoft Windows, et Mac OS X créé en 1997, commercialisé par Macromedia puis Adobe Systems sous licence utilisateur final.

Dreamweaver fut l'un des premiers éditeurs HTML de type « tel affichage, tel résultat », mais également l'un des premiers à intégrer un gestionnaire de site. Ces innovations l'imposèrent rapidement comme l'un des principaux éditeurs de site web, aussi bien utilisable par le néophyte que par le professionnel.

4.3.5 Choix d'outils de développement web

Nous avons opté pour l'IDE Netbeans ces caractéristiques modernes (éditeurs en couleurs, projets multi-langage, refactoring et éditeur graphique d'interfaces et pages web)

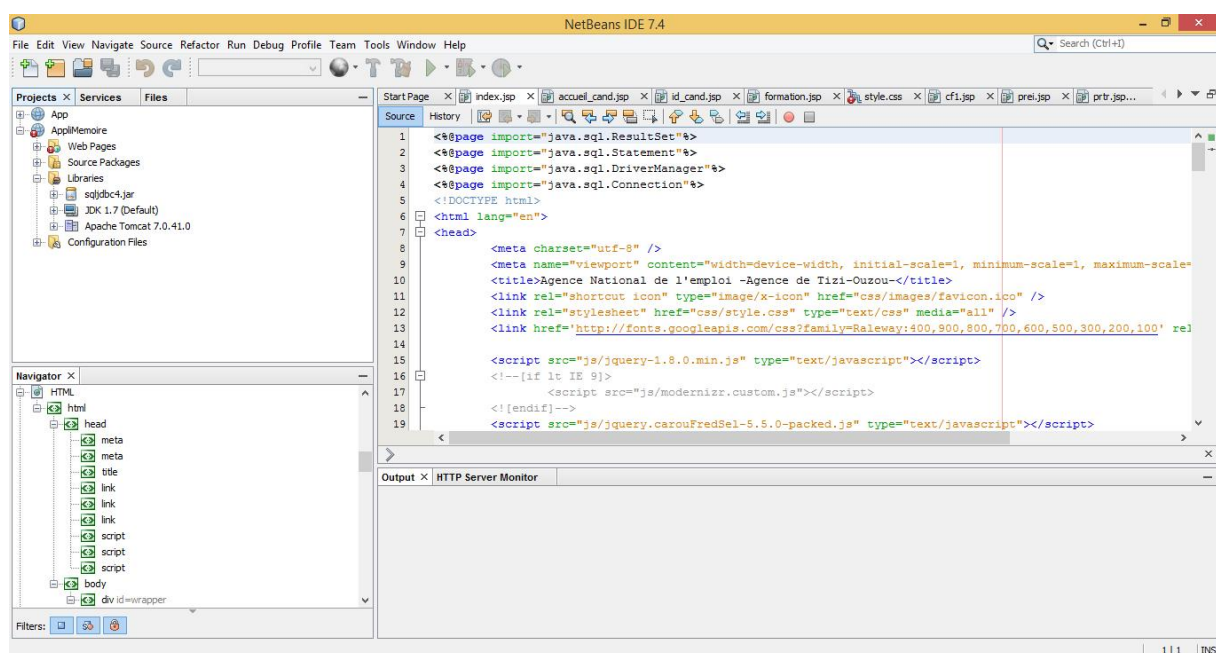


Figure 4.2 Fenêtre Netbeans IDE 7.4

Langages web utilisés

- **Le langage HTML** : pour la création des pages statiques de l'application qui constitueront l'interface utilisateur.
- **JavaScript** : **JavaScript** (souvent abrégé JS) est un langage de programmation de scripts principalement utilisé dans les pages web interactives.
- **JSP (JAVA SERVER PAGE)** : pour la partie dynamique du site. Il constitue le langage pilier de l'application, c'est-à-dire, que c'est sur lui que reposent le dynamisme de l'application, c'est la raison pour laquelle nous allons mieux le détailler.

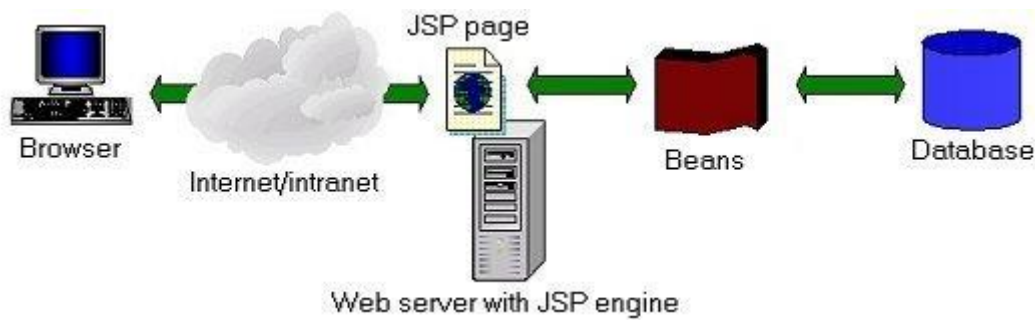
4.4 JSP

JSP présente de nombreux avantages par rapport aux autres langages de génération de contenu. Comme il s'agit d'une technologie basée sur Java, JSP profite de tous ce qu'offre ce langage pour le développement et le déploiement d'applications. En effet Java est un langage orienté objet fortement typé, permettant l'encapsulation, le traitement des exceptions et la gestion automatique de la mémoire.

D'autre part, grâce aux API normalisées pour les JSP et à la portabilité du bytecode compilé Java, on n'est pas limité à un seul type de plate-forme, de système d'exploitation. On peut changer les composants d'une page JSP comme on le souhaite. De plus, JSP utilise l'ensemble de la plate-forme JAVA sous-jacente ; par conséquent, les JSP peuvent directement tirer avantage de toutes les API Java.

JSP n'est pas rattaché à un éditeur particulier, les développeurs peuvent donc choisir les solutions de leurs choix.

Lorsque la page est demandée par un utilisateur en HTTP, alors le serveur web HTTP va transmettre sa requête à un moteur de JSP qui va interpréter le page, compiler le code et générer la réponse. L'utilisateur ne verra aucune ligne de code Java dans les sources de la page qu'il aura reçu.



Fragment de code en JSP

```
<%  
    java.util.Date dateToday = new java.util.Date();  
%>  
  
<html>  
<body>  
  
<p>La date est <%= dateToday %> </p>  
  
</body>  
</html>
```

Servlet

Un **servlet** est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP. Ces données sont le plus généralement présentées au format HTML, mais elles peuvent également l'être au format XML ou tout autre format destiné aux navigateurs web. Les servlets utilisent l'API **Java Servlet** (package `javax.servlet`).

Example :

```
import javax.servlet.* ;  
import java.io.* ;  
  
public class HelloServlet extends GenericServlet  
{  
    public void service (HttpServletRequest request, HttpServletResponse  
response)  
    {  
        try  
        {  
            PrintWriter out = response.getWriter() ;  
            out.println ("<!DOCTYPE html PUBLIC \"-//W3C//DTD HTML 4.01//EN\">")  
;  
            out.println ("Bonjour tout le monde&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~!</title>") ;<br/>            out.println ("<p>Hello world!</p>") ;</code></pre>
```

```
    }  
    catch (IOException e)  
    {  
        e.printStackTrace() ;  
    }  
}  
}
```

4.4.1 Serveur Web Apache tomcat

L'ancêtre d'apache est le serveur libre développé par le (National Center for Supercomputing Applications) de l'université de l'Illinois. L'évolution de ce serveur s'est arrêtée lorsque le responsable a quitté le NCSA en 1994, les utilisateurs ont continué à corriger les bugs et à créer des extensions qu'ils distribuaient sous forme de " patches " (bouts de programmes ajoutés par les utilisateurs des NCSA pour étendre les fonctionnalités d'Apache) d'où le nom "a patchee Server" la version 1.0 d'Apache a été disponible le 1 décembre 1995.

Nous choisissons et présentons ce serveur par les avantages qu'il offre et que nous allons détailler ci-après :

- C'est un serveur gratuit (peut être téléchargé à partir du site du groupe Apache à l'adresse '<http://www.apache.org>').
- Un niveau élevé de performances pour des besoins matériels modeste.
- Extensible, modulaire et configurable.
- Robuste.
- Très portable contrairement à IIS (Internet Information Services) de Microsoft qui tourne seulement sous Windows, le serveur Web apache dispose d'une version pour chaque plateforme (Linux, Windows, ...).

4.5 Quelques système de gestion de bases de données

4.5.1 Oracle DATABASE

Est un système de gestion de base de données relationnel (SGBDR) qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO). Fourni par Oracle Corporation, il a été développé par Larry Ellison, accompagné d'autres personnes telles que Bob Miner et Ed Oates.

4.5.2 Firebird

Est un serveur de bases de données relationnelles SQL fonctionnant sous Linux, Mac OS X, Unix et Windows. Il est transactionnel multi-niveaux (*two-phase commit* sur une transaction partagée sur plusieurs connexions à des bases de données différentes).

4.5.3 MySQL

Est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

Son nom vient du prénom de la fille du Co créateur Micheal Widenius, MySQL fait allusion au *Structured Query Language*, le langage de requête utilisé.

MySQL AB a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle Database et MySQL. Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010.

Depuis mai 2009, son créateur Michael Widenius a créé Maria DB pour continuer son développement en tant que projet Open Source.

4.5.4 PostgreSQL

Autre grand nom de la base de données libre, PostgreSQL, dont la version actuelle est la 8.1, est une base de données relationnelle objet (Possibilité de stocker des objets dans la base). Le SGBD supporte une grande partie de la norme SQL et offre les fonctionnalités des bases de données modernes comme les requêtes complexes, les clés étrangères, les déclencheurs, les vues. Il supporte aussi l'objet relationnel dont les extensions sont spécifiées dans la norme SQL-99. Néanmoins, PostgreSQL étant la seule base libre à supporter l'objet relationnel, la comparaison avec d'autres bases sur ce sujet n'a pas de vraiment de sens.

4.5.5 Microsoft SQL Server

Est un système de gestion de base de données développé et commercialisé par la société Microsoft.

Bien qu'il ait été initialement Co développé par Sybase et Microsoft, Ashton-Tate a également été associé à sa première version, sortie en 1989. Cette version est sortie sur les plates-

formes Unix et OS/2. Depuis, Microsoft a porté ce système de base de données sous Windows et il est désormais uniquement pris en charge par ce système.

Nous avons opté pour Microsoft SQL Server par rapport à ses concurrents que sont Oracle, MySQL ou PostgreSQL, SQL Server se distingue par le fait que c'est un SGBDR originellement multi base (contrairement à Oracle) et multi schémas (contrairement à MySQL). Il est possible de faire des requêtes nativement interbases. Lié deux tables de deux bases de données différentes.

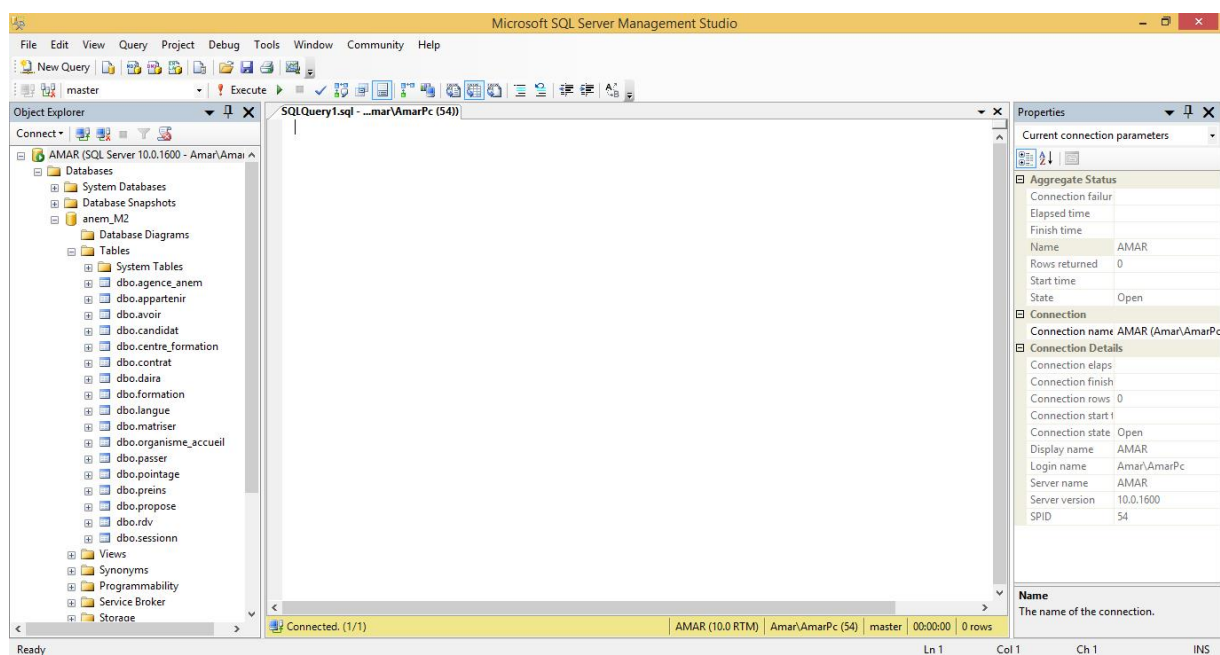


Figure 4.3 Microsoft SQL Server 2008 Management Studio

3.5.6 Langages d'interrogation de bases de données utilisés

SQL

Sigle de Structured Query Language, en français langage de requête structurée est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

TSQL

Le Transact-SQL (T-SQL) est une extension propriétaire de Sybase et Microsoft au langage SQL. Transact-SQL a été développé à l'origine par la société Sybase, dès les premières versions de son moteur de base de données du même nom. De manière similaire au PL/SQL d'Oracle, Transact-SQL fournissait le moyen d'étendre les fonctionnalités de base du SGBD, via des programmes appelés "procédures stockées". Le code source comme compilé, est en effet stocké dans la base de données, par opposition aux programmes écrits en langage de programmation classique, dont le code source d'une part, le code compilé d'autre part, sont stockés dans des fichiers du système de fichiers.

Procédures stockées

En informatique, dans la technologie des bases de données, une procédure stockée (ou *stored procedure* en anglais) est un ensemble d'instructions SQL précompilées, stockées dans une base de données et exécutées sur demande par le SGBD qui manipule la base de données.

Exemple de procédure stockée

```
GO

SET QUOTED_IDENTIFIER ON
GO

create procedure [dbo].[Ajout_contrat]
@num_contrat varchar(15),
@date_etablissement date=NOW,
@date_debut date,
@date_fin date,
@num_dossier varchar(20),
@code_agence varchar(15),
@code_formation varchar(20),
@code_cf varchar(15),
@cod_org varchar(15),
@existe int output
as
begin
set nocount on
```

```

if exists(select num_contrat from contrat where num_contrat=@num_contrat)
begin
set @existe=1;
end
else
set @existe=0;
begin
insert into
contrat(num_contrat,date_etablissement,date_debut,date_fin,num_dossier,code
_agence,code_formation,code_cf,cod_org)
values(@num_contrat,@date_etablissement,@date_debut,@date_fin,@num_dossier,
@code_agence,@code_formation,@code_cf,@cod_org)
end
END
GO

```

4.6 Modèle physique de données

4.6.1 Candidat

Champ	Signification	Type	Clé
Num_dossier	Numéro du dossier	Varchar(20)	Primaire
Nom	Nom du candidat	Varchar(30)	
Prénom	Prénom du candidat	Varchar(30)	
Date_naissance	Date de naissance du candidat	Date	
Lieu_naissance	Lieu de naissance du candidat	Varchar(30)	
Date_inscription	Date d'inscription	Date	
Sexe_candidat	Sexe du candidat	Varchar(10)	
Num_tel	Numéro de téléphone du candidat	Varchar(20)	
Num_ss	Sécurité social du candidat	Varchar(15)	
Num_ccp	Compte CCP du candidat	Varchar(30)	
E_mail	Compte mail du candidat	Varchar(20)	

Code_secret	Code pour authentication	Varchar(20)	
-------------	--------------------------	-------------	--

Tableau 4.1.1 Table candidat

4.6.2 Agence

Champ	Signification	Type	Clé
Code_agence	Code de l'agence	Varchar(15)	Primaire
Adresse_ag	Adresse de l'agence	Varchar(50)	
Num_tel_ag	Numéro de téléphone de l'agence	Varchar(20)	

Tableau 4.1.2 Table agence

4.6.3 Centre de formation

Champ	Signification	Type	Clé
Code_cf	Code du centre de formation	Varchar(15)	Primaire
Nom_cf	Nom du centre	Varchar(50)	
Num_tel_cf	Numéro de téléphone du centre	Varchar(20)	
Adresse_cf	Adresse du centre de formation	Varchar(50)	
Code_agence	Code de l'agence ANEM	Varchar(15)	Etrangère

Tableau 4.1.3 Table centre de formation

4.6.4 Formation

Champ	Signification	Type	Clé
Code_formation	Code de la formation	Varchar(15)	Primaire
Nom_formation	Nom de la formation	Varchar(50)	
Duree_form	Durée de la formation	Varchar(10)	

Niveau	Niveau minimum de la formation	Varchar(20)	
--------	--------------------------------	-------------	--

Tableau 4.1.4 Table Formation

4.6.5 Session

Champ	Signification	Type	Clé
Code_session	Code de la session	Varchar(15)	Primaire
Date_debut	Debut de la session	Date	
Date_fin	Fin de la session	Date	

Tableau 4.1.5 Table session

4.6.6 Avoir

Champ	Signification	Type	Clé
Code_formation	Code de formation	Varchar(15)	Primaire
Code_session	Code de la session	Varchar(15)	Primaire

Tableau 4.1.6 Table avoir

4.6.7 Appartenir

Champ	Signification	Type	Clé
Code_formation	Code de formation	Varchar(15)	Primaire
Code_cf	Code du centre de formation	Varchar(15)	Primaire

Tableau 4.1.7 Table appartenir

4.6.8 Langue

Champ	Signification	Type	Clé
Id_langue	Identifiant de la langue	Varchar(15)	Primaire
Desc_langue	Description de la langue	Varchar(15)	

Tableau 4.1.8 Table langue

4.6.9 Maitriser

Champ	Signification	Type	Clé
Num_dossier	Code du candidat	Varchar(20)	Primaire
Id_langue	Identifiant de la langue	Varchar(15)	Primaire

*Tableau 4.1.9 Table maitriser***4.6.10 Proposer**

Champ	Signification	Type	Clé
Code_cf	Code du centre de formation	Varchar(15)	Primaire
Code_formation	Code de la formation	Varchar(15)	Primaire

*Tableau 4.1.10 Table proposer***4.6.11 Organisme accueil**

Champ	Signification	Type	Clé
Code_org	Code de l'organisme d'accueil	Varchar(15)	Primaire
Adresse_org	Adresse de l'organisme d'accueil	Varchar(30)	
Num_tel_org	Téléphone de l'organisme d'accueil	Varchar(15)	

*Tableau 4.1.11 Table accueil***4.6.12 Contrat**

Champ	Signification	Type	Clé
Num_contrat	Numéro du contrat	Varchar(20)	Primaire
Date_etablissement	Date d'établissement	Date	
Date_debut	Date début du contrat	Date	

Date_fin	Date de la fin du contrat	Date	
Num_dossier	Numéro du dossier du candidat	Varchar(20)	Etrangère
Code_agence	Code de l'agence ANEM	Varchar(15)	Etrangère
Code_formation	Code de la formation	Varchar(15)	Etrangère
Code_cf	Code du centre de formation	Varchar(15)	Etrangère
Code_org	Code de l'organisme d'accueil	Varchar(15)	Etrangère

Tableau 4.1.12 Table contrat

4.6.13 Etat

Champ	Signification	Type	Clé
Id_etat	Identifiant de l'état du dossier	Varchar(15)	Primaire
Etat_actuel	Etat du dossier actuellement	Varchar(15)	
Num_dossier	Numéro du dossier du candidat	Varchar(20)	Etrangère

Tableau 4.1.13 Table candidat

4.6.14 Pointage

Champ	Signification	Type	Clé
Num_contrat	Numéro de contrat	Varchar(15)	Primaire
Date	Date de pointage	Date	Primaire

Tableau 4.1.14 Table pointage

4.6.15 Daïra

Champ	Signification	Type	Clé
Code_daira	Code de la Daïra	Varchar(15)	Primaire
Nom_daira	Nom de la Daïra	Varchar(40)	

*Tableau 4.1.15 Table daïra***4.6.16 Préinscription**

Champ	Signification	Type	Clé
Num_pi	Numéro de pièce d'identité	Varchar(20)	Primaire
Nom	Nom	Varchar(30)	
Prenom	Prénom	Varchar(30)	
Date_naissance	Date de naissance	Date	
Lieu_naissance	Lieu de naissance	Varchar(20)	Etrangère

*Tableau 4.1.16 Table préinscription***4.6.17 RDV**

Champ	Signification	Type	Clé
Code_rdv	Code du rendez-vous	Int	Primaire
Date_RDV	Date du rendez vous	Date	

Tableau 4.1.17 Table RDV

4.7 Quelques interfaces de l'AGSA

4.7.1 Interface D'accueil

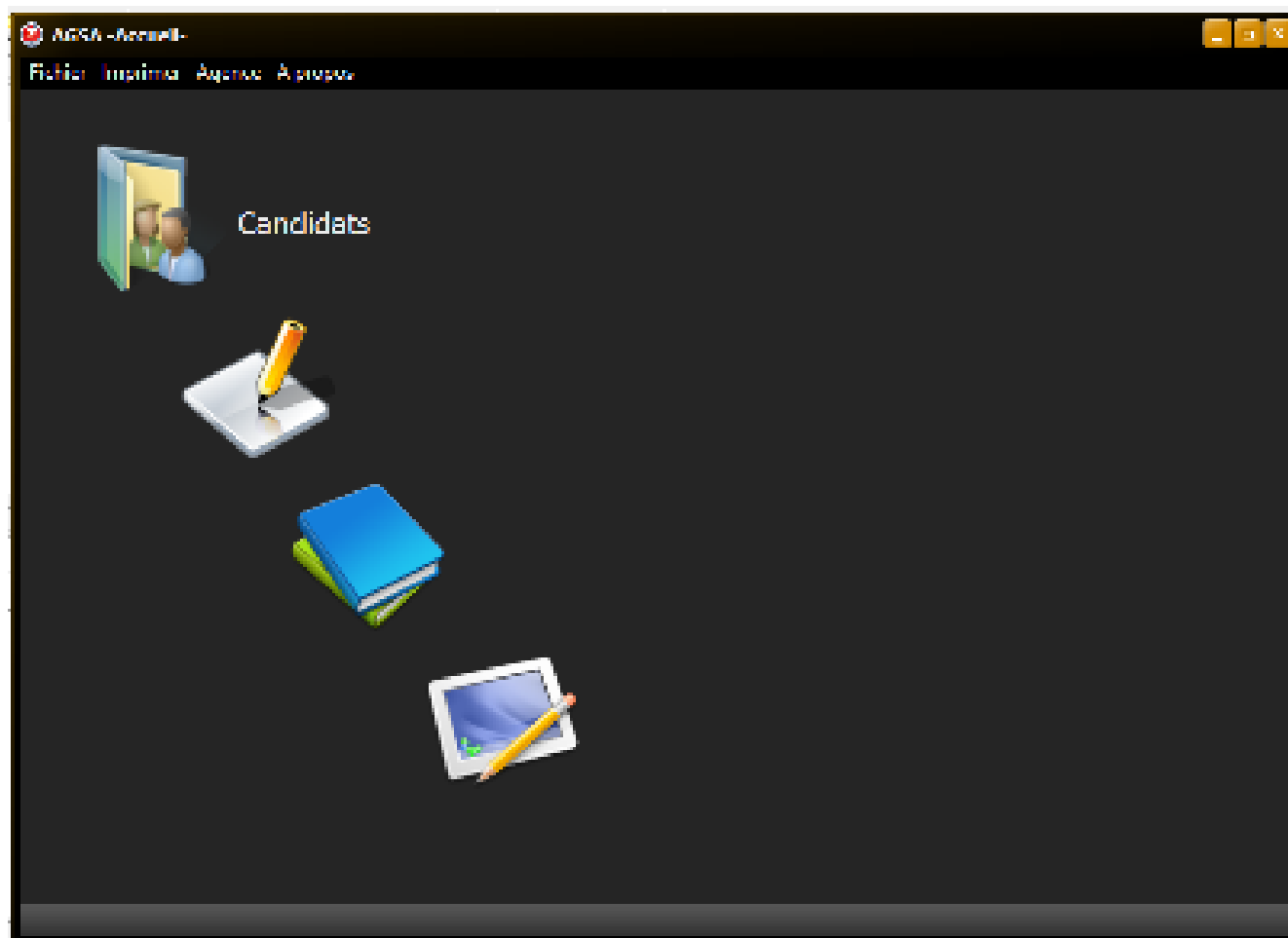


Figure 4.4 Interface d'accueil d'AGSA

4.7.2 Interface Gestion des candidats

AGSA - Accueil -

Fichier Imprimer Aide À propos

Gestion de candidature

Creation d'un candidat:

Matricule

Nom Prénom Date de naissance

Lieu de naissance Adresse

Sexe

Numéro de téléphone Numéro de sécurité social

Niveau d'études

Adresse électronique

Valider Annuler

Matricule	Nom	Prénom	Date de nais
1508/2014/0001	AICHOIR	AMIROUCHE	1996-05-31
1508/2014/0002	LAOUFI	MOHAMED AMINE	1995-07-13

Supprimer

Imprimer fiche

Suivi des dossier

Figure 4.5 interface Gestion des candidats

4.7.3 Interface gestion des formations

Gestion des formations

Ajout de formation

Code de formation: Nom de formation:
Durée de formation: Niveau minimum:

Code	Nom	Durée	Niveau
▶ 15-0001	Maçon	12	7

Buttons: Valider, Supprimer la ligne, Annuler

Session

Code Session: Date Début: Date Fin:
Buttons: Valider, Annuler

Code	Début	Fin
▶		

Button: Supprimer

Formation/Session

Code formation: Code Session:
Buttons: Valider, Annuler

code_formation	code_session
▶	

Button: Supprimer

Figure 4.6 Interface Gestion des formations

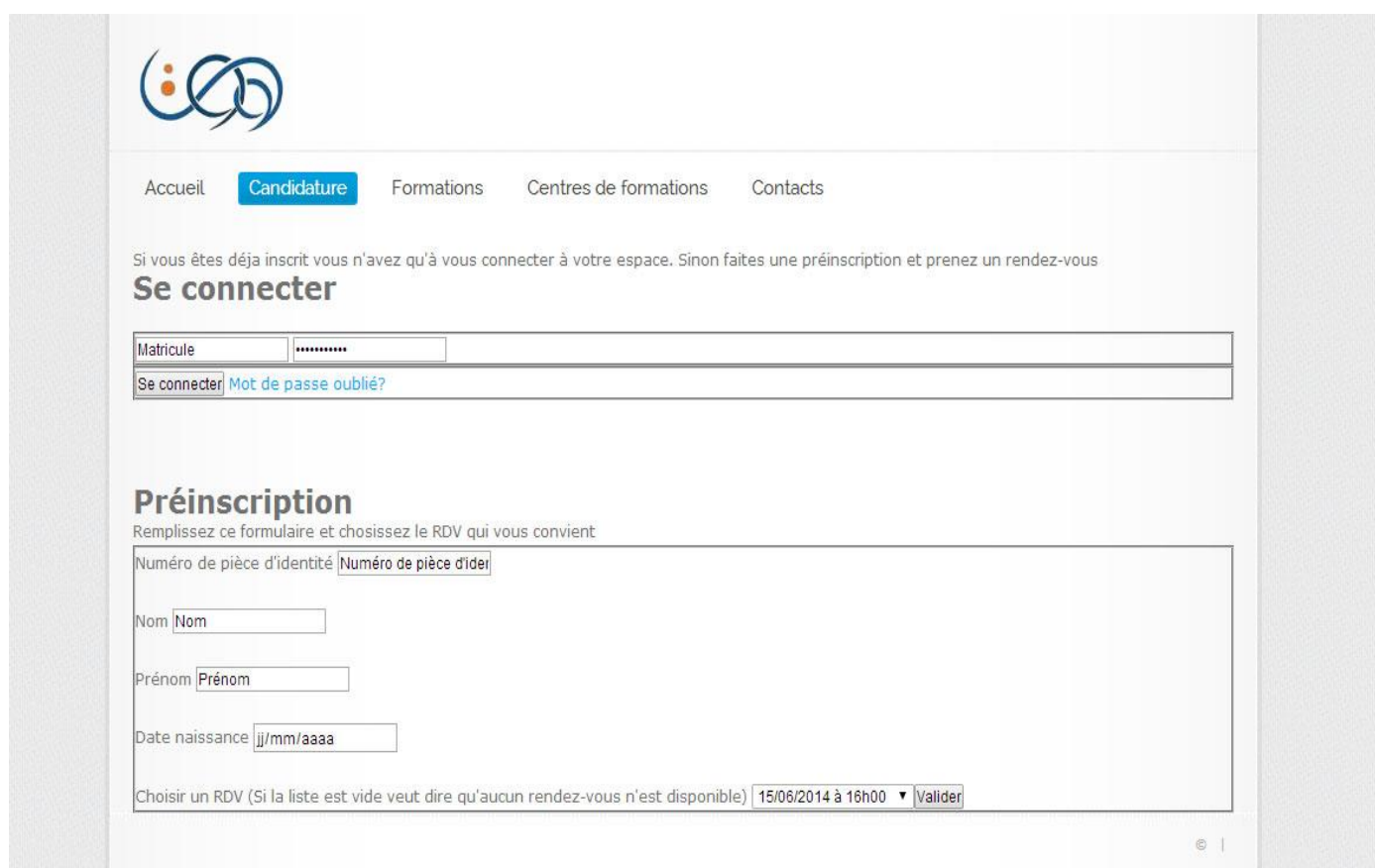
4.8 Quelques interfaces de l'application Web

4.8.4 Interface Accueil



Figure 4.7 Interface d'accueil de l'application web

4.8.5 Interface l'inscription en ligne



The screenshot displays a web interface for online registration. At the top, there is a logo consisting of two interlocking circles with a dot in the center. Below the logo is a navigation bar with links: Accueil, **Candidature** (highlighted in blue), Formations, Centres de formations, and Contacts.

Below the navigation bar, a message states: "Si vous êtes déjà inscrit vous n'avez qu'à vous connecter à votre espace. Sinon faites une préinscription et prenez un rendez-vous".

The main section is titled "Se connecter". It contains a form with two input fields: "Matricule" and a password field (indicated by dots). Below these fields is a "Se connecter" button and a link "Mot de passe oublié?".

The next section is titled "Préinscription". It includes the instruction "Remplissez ce formulaire et choisissez le RDV qui vous convient". The form contains several input fields: "Numéro de pièce d'identité" (with a placeholder "Numéro de pièce d'identité"), "Nom" (with a placeholder "Nom"), "Prénom" (with a placeholder "Prénom"), and "Date naissance" (with a placeholder "jj/mm/aaaa").

At the bottom of the form, there is a field for "Choisir un RDV (Si la liste est vide veut dire qu'aucun rendez-vous n'est disponible)" with a dropdown menu showing "15/06/2014 à 16h00" and a "Valider" button.

Figure 4.8 interface d'inscription en ligne

4.8.6 Interface espace candidat

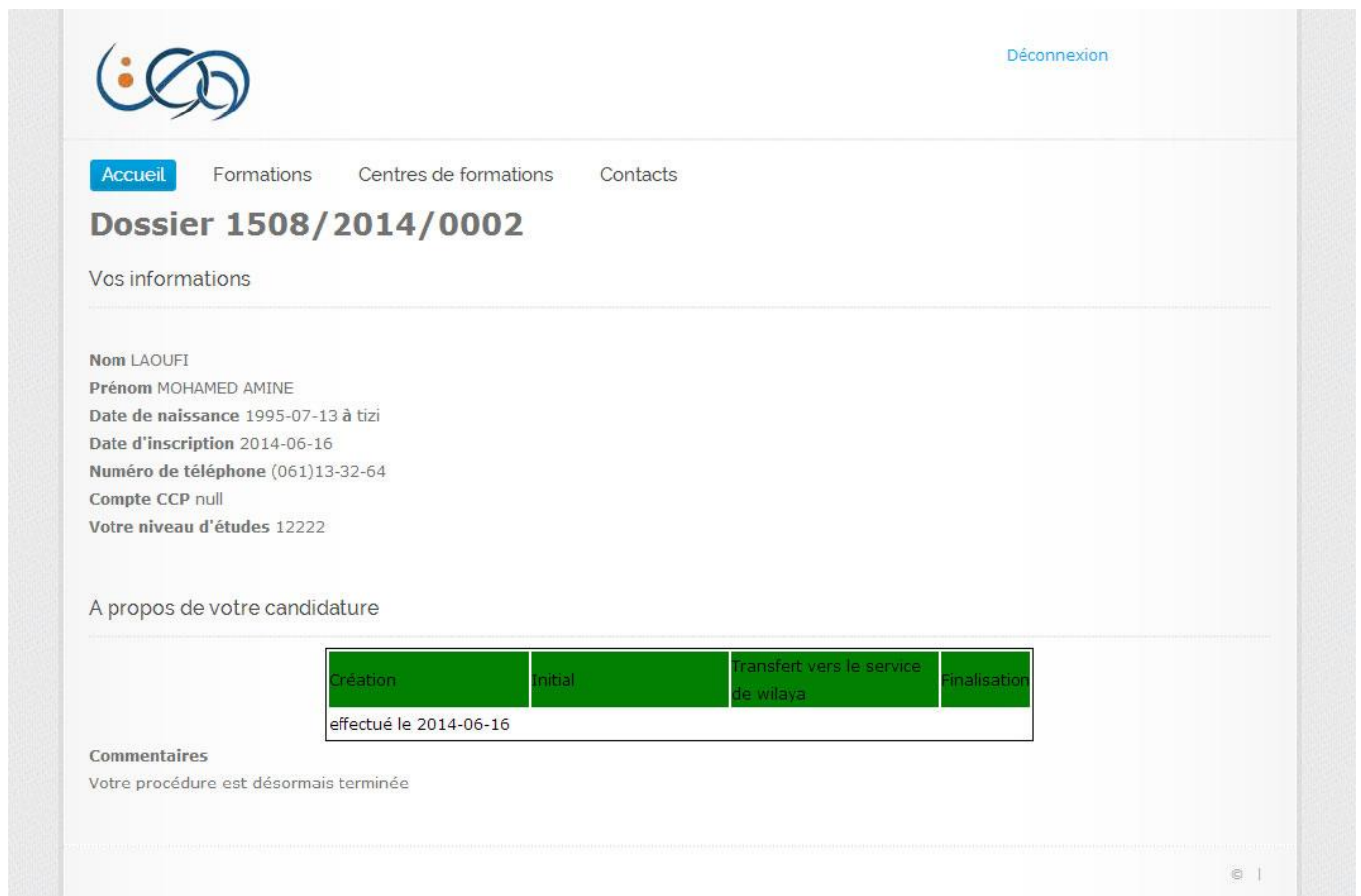


Figure 4.9 Interface de l'espace candidat

4.9 Conclusion

Dans ce chapitre nous avons présenté les outils et l'environnement développement de l'application. Nous avons en outre explicité les composants de la base de données, puis pour terminer nous avons présenté quelques interfaces de l'application.

Conclusion générale

La valeur d'une œuvre se reconnaît à ses fruits, c'est-à-dire aux résultats qu'elle produit. Nous pensons également qu'étant arrivé à la fin de notre travail, sa valeur ne saurait être jugée que par les résultats qui pourront se laisser percevoir au fur et à mesure de la réalisation de ce projet par l'Etablissement bénéficiaire.

En effet, le projet que voici et qui arrive à son terme, a consisté en la conception d'un système de Gestion des demandeurs d'emploi âgés de 16 à 20 ans pour l'agence nationale de l'emploi.

La réalisation de ce projet fera en sorte que la gestion des candidats, formations et des centres de formations qu'ils constitueront plus un problème mais plutôt un résultat d'un traitement automatique par l'ordinateur. Ce qui a l'avantage d'éviter maintes erreurs dues à la manipulation manuelle d'une kyrielle des données.

Bien sécurisée, sur le plan Hard et Soft, cette BDD assure en même temps un grand service d'archivage d'informations très utiles à la prise des décisions dans le futur. Il suffirait tout simplement de se référer aux cas similaires et adapter la décision à prendre selon le temps et les réalités actuelles.

Au-delà de toute prétention de perfection, nous reconnaissons avec humilité que tout travail d'homme est sujet à quelques imperfections. Ce projet étant un travail d'homme, nous estimons qu'il n'échappe pas à la règle. Sur ce nous sollicitons l'indulgence de nos lecteurs futurs quant à une éventuelle coquille ou incohérence pouvant être relevée dans ce travail. Sachant que c'est la falsifiabilité qui fait avancer la science, nous restons ouverts à toute critique susceptible de nous aider à améliorer notre vision du problème.

Ce projet a fait l'objet d'une expérience intéressante qui nous a permis de renforcer nos connaissances concernant le développement des applications web avec J2EE, applications de bureau avec DELPHI XE4, modélisation avec UML, analyse avec Merise et enfin la conception et l'administration des bases de données avec Microsoft SQL Server 2008.

Références

- [AD 11] Réseaux Andrew Tanenbaum et David Wetherall, Editeur Pearson 26/08/2011
- [DC 09] TCP/IP Architecture, Protocoles et application Douglas Comer, Editeur Pearson 01/01/2009
- [JL 08] Jessica Livingston & Judith Strauser, Les plus grandes réussites du web. Edition ESF 2008.
- [DD 08] Dominique Desmont, Conception Web. Edition : ENI 2008.
- [FL 04] F.LALAND Outils du web (langages, serveur). Février 2004.
- [HB 06] UML2 Compact de Heide Balzert, Editeur Eyrolles 16/02/2006
- [BAY 10] UML2 Pratique de la modélisation de Benoît Charroux, Aomar Osmani, Yann Thierry-Mieg Editeur Pearson 26/06/2010
- [GJ 03] Le guide de l'utilisateur UML Grady Booch, James Rumbaugh, Editeur Eyrolles 24/04/2003
- [PR 08] Pascal Roques les cahiers des programmeurs UML2 : Modéliser une application web 3ème Édition. Edition Eyrolles 02 Octobre 2008
- [JR 03] Jérôme Molière les cahiers de programmeur J2EE. Edition Eyrolles 23 Octobre 2003.
- [JG 09] Jérôme GABILLAUD Administration d'une base de données avec SQL Server Management Studio. Edition ENI Fevrier 2008

<http://www.commentcamarche.net>

<http://fr.wikipedia.org>

<http://www.developpez.com>

[http:// www.uml.free.fr.](http://www.uml.free.fr)

<http://www.memoireonline.com>

<http://www.futura-sciences.com/fr>

<http://java.developpez.com/>

<http://delphi.developpez.com/cours/>