

Remerciements

Avant tout, nous remercions Dieu qui nous a donné la foi, la santé, le courage et la volonté pour terminer ce modeste travail.

Nous tenons aussi à exprimer notre reconnaissance et profonde gratitude à notre promotrice, madame HOCINI née OUKFIF Karima pour sa présence et sa disponibilité durant cette année, pour son exigence scientifique, pour son encouragement et sa patience.

Un grand Merci aux membres du jury pour l'honneur qu'ils nous ont fait d'évaluer et de juger notre travail.

Nos sincères sentiments s'adressent à nos familles et amis (es) pour leurs soutiens et leurs encouragements.

Enfin, nos remerciements vont à tous ceux qui de près ou de loin nous ont aidés à l'aboutissement de cette quête.

Dédicaces :

Je dédie ce modeste travail :

*À la première femme que mon regard a croisée,
à ma mère.*

*À mon guide qui me montra le chemin dans les
nuits obscures,*

à mon père.

À mes chers frères.

À ma très chère sœur et son époux Ali.

À toi mon cher neveu Axil.

À tous les membres de ma famille.

À tous mes amis que j'aime beaucoup.

Brahim

Je dédie ce modeste travail :

A vous mes remarquables parent, pour vos sacrifices et votre total dévouement, vous qui m'avez toujours soutenu et été là pour moi.

A mes très chers frères Essaid et Madjid, et à leurs femmes.

A mes adorables sœurs Nora et Hakima.

A vous mes chers neveux Alexandre, Wassim et Chayma.

A tous mes amis avec qui j'ai passé d'agréables moments, et une pensée très particulière pour Malika B.

Farid

Sommaire

CHAPITRE I

I. Introduction	2
II. Historique des web services	2
III. Définition des Services Web.....	3
IV. Cycle de vie d'utilisation des web services.....	4
V. Architecture des web services.....	4
VI.XML et les trois standards SOAP, WSDL, UDDI	7
VI.1- XML (eXtensible Markup Language).....	7
VI.1.2- Définition.....	7
VI.1.3- Quelques différences entre le HTML et le XML.....	7
VI.2- Protocol SOAP.....	8
VI.2.1- Définition.....	8
VI.2.2- Invocation de service d'objets distant via SOAP RPC	9
VI.2.3- structure d'un message SOAP	11
VI.3- WSDL (Web Services Description Language	11
VI.3.1- La structure d'un document WSDL.....	12
VI.4- L'Annuaire UDDI.....	13
VI.4.1- Définition.....	13
VI.4.2- Pages de l'annuaire.....	13
VI.4.3- Mécanismes d'accès aux services fournis par UDDI.....	14
VI.4.4- Le modèle de données dans UDDI.....	15
VI.4.5- L'interface d'UDDI.....	16
VI.4.6- L'usage de l'annuaire UDDI.....	17
VI.4.7- Opérateurs d'annuaires UDDI actuels	18
VII. Architecture en couches des standards des services Web.....	19
VIII. Avantage des web services.....	20
IX. Apport des Web services aux entreprises.....	20
X. Limites des web services.....	21
XI. Composition des web Services.....	21
XII-1. Définition.....	22
XII-2. Types de composition.....	22
XII-2.1. Composition manuelle.....	22
XII-2.2. Composition semi-automatique.....	22
XII-2.3. Composition automatique.....	22
XIII. Langages de définition de la composition des web services.....	22
XIII-1. Business Process Modeling Language (BPML).....	23
XIII-2. Web Service Choreography Interface (WSCI).....	23
XIII-3. Web Service Conversation Language (WSCL).....	23
XIV. Sécurité des web services	23

XIV.1- vulnérabilités des web services.....	23
XIV.2- Quelques protocoles de Sécurisation des web services.....	25
XV. Conclusion.....	26

CHAPITRE II

I. Introduction.....	27
II. Architecture Logiciel.....	27
II.1 Définition1.....	27
II.2 Définition2	27
III. Critères de qualité d'un logiciel.....	28
IV. Présentation de J2EE.....	29
V. Topologie(s) d'une application J2EE	30
V.1 Application multi tiers :	30
V.2 Application web.....	30
VI. Architecture J2EE.....	30
VII. Composants de l'architecture J2EE.....	31
VII.1. Conteneur de composants J2EE.....	31
VII.1.1 Les conteneurs.....	31
VII.1.2 Le conteneur web	32
VII.1.3 Le conteneur d'EJB.....	32
VII. 2.Clients J2EE.....	33
VII .3.Composants web	33
VII .3.1. Pages JSPsC.....	33
VII .3.2. Tags	33
VII .4. Composants Enterprise JavaBeans [EJB].....	34
VII.4.1. Beans de session (session bean).....	34
VII .4.2. Les EJB entité	34
VII.4.3. Bean de message (Message-Driven Beans)	34
VII.5. Serveur EJB	35
VIII. Les services proposés par la plate-forme J2EE.....	36
VIII.1 Les services d'infrastructures.....	37
VIII.2 :Les services de communication	37
IX. Les différentes étapes pour créer un objet distant et l'appeler avec RMI.....	39

X. L'environnement d'exécution des applications J2EE.....	39
XI. L'assemblage et le déploiement d'applications J2EE.....	40
XII. Conclusion.....	40

CHAPITRE III

I. Introduction.....	41
II. Organisation de NAFTAL	41
III. Organigramme général de NAFTAL	42
IV. Organigramme du champ d'étude	43
V. Problématique de notre champ d'études.....	43
VI. Analyse.....	47
VI.1 Spécification des besoins.....	44
VI.2 Identification des acteurs	44
VI.3 Identification des cas d'utilisations	45
VI.4 Architecture générale du système.....	46
VI.5 : Spécification des scénarios	48
VII Conception	53
VII.1 : Diagramme de cas d'utilisation	53
VII.2 : Diagramme de séquence	58
VII.3 : Diagramme d'activité	62
VII.4 : Diagramme de classe	68
VII.5 : Conception de la base de données.....	69
VII.6 : Le modèle relationnel.....	70
VII.7 : Dictionnaire de données	71
VIII Conclusion	72

CHAPITRE IV

I.Introduction.....	73
II.	73
II. Partie I :Outils de développement	73
II.1.1- PostgreSQL	73
II.1.1.1- Présentation de PostgreSQL.....	73
II.1.1.2- Vue sur notre base de données sur POSTGRES.....	74
II.1.2- Le langage Java	75
II.1.2.1- La programmation graphique.....	75

II.1.2.2- Environnement de développement NetBeans	76
II.1.3- Le serveur d'application (Glassfish)	77
III. Partie II.....	78
III.1. Implémentation des différentes parties de l'application.....	78
III.1.1. Partie stockage.....	78
III.1.2. Partie Traitement.....	79
III.1.2.1. le projet [projet_memoire].....	79
III.1.2.2. le module projet_memoire-ejb.....	80
III.1.2.3. les modules web services.....	83
III.1.2.4. les modules client.....	84
IV- Présentation de l'application.....	85
IV.1. Interfaces authentification	86
IV.2. Interfaces de gestion des paramètre de compte.....	87
IV.3. Interfaces Espace DT du Service la Maintenance.....	88
IV.4. Interfaces Formulaire Compte-rendu.....	89
IV.5. Interfaces Formulaire ajout contrôle débit.....	90
IV.6. Interfaces ajout DT.....	91
V- Conclusion.....	91

Figure I-1 : Cycle de vie d'utilisation des web services.....	4
Figure I-2: Architecture des web services	5
Figure I-5 : Structure d'un document WSDL.....	7
Figure I-3: Etapes d'invocation d'objets distant avec SOAP	9
Figure I-4 : Structure de message SOAP.....	11
Figure I-6 : Page de l'annuaire UDDI.....	14
Figure I-7 : Mécanismes d'accès aux services de UDDI	14
Figure I-8 : Modèle de données de l'annuaire UDDI.....	15
Figure I-9 : Architecture en couches des protocoles et standards des services Web.....	19
Figure I-10: Web Service Orchestration.....	21
Figure I-11 : Web Services chorégraphie.....	22
Figure II-1 : Architecture et les composants de J2EE.....	31
Figure II-2 : l'architecture générale d'un EJB.....	35
Figure III-1 : Organigramme De NAFTAL	42
Figure III-2 : Organigramme du champ d'étude.....	43
Figure III-3 : Diagramme de cas d'utilisation général.....	53
Figure III-4 :Diagramme de cas d'utilisation Personnel.....	55
Figure III-5 :Diagramme de cas d'utilisation Service Maintenance.....	56
Figure III-6 :Diagramme de cas d'utilisation Personnel.....	57
Figure III-7 :Diagramme de séquence «cas Authentification ».....	59
Figure III-8 :Diagramme de séquence «cas Ajout ».....	60
Figure III-9 :Diagramme de séquence «cas modification du mot de passe ».....	61
Figure III-10 :Diagramme d'activité de l'authentification.....	63
Figure III-11 :Diagramme d'activité d'ajout.....	64
Figure III-12 :Diagramme d'activité de suppression.....	65
Figure III-13 : Diagramme d'activité de recherche	66
Figure III-14 : Diagramme d'activité de consultation.....	67
Figure III-15 : Diagramme de classe général.....	68
Figure III-16 : Dictionnaire de données.....	71
Figure IV.1 : Interface de notre BDD sous POSTGRES.....	74
Figure IV-2 : les classes de base de la classe JComponent.....	76
Figure IV-3 : l'interface graphique de NetBeans.....	77
Figure IV.4 Interface administrateur du serveur d'applications GlassFish.....	78
Figure IV.5. Script de création de la table panne.....	78
Figure IV.6. pilote de connexion JDBC	79
Figure IV.7. projet_memoire.....	80
Figure IV.8. le module projet_memoire-ejb	80
Figure IV.9 Les web services.....	81
Figure IV.10. Le déploiement des web services.....	82
Figure IV.11. page test du web service.....	83

Figure IV.13. Module Client.....	84
Figure IV.14. Le déploiement des client.....	84
Figure IV.15. page lancement des client.....	85
Figure IV.16.page d'enregistrement ou de lancement du client.....	85
Figure IV.17: L'interface d'authentification du service maintenance et personnel.....	86
Figure IV.18: L'interface d'authentification des stations.....	86
Figure IV.19: Interfaces de gestion des paramètres du compte.....	87
Figure IV.20: Interfaces Espace DT de la maintenance.....	88
Figure IV.21: Interfaces Formulaire Compte-rendu.....	89
Figure IV.22: Interfaces Formulaire ajout contrôle débit.....	90
Figure IV.23 : Interfaces ajout DT.....	91

Glossaire

API : Application Program Interface.

B2B : Business to Business.

BPML4WS : Business Process Execution Language for Web Services.

BPML : Business Process Markup Language.

CORBA : Common Object Request Broker Architecture.

DCOM : Distributed Component Object Model.

DTD : Document Type Definition.

ebXML : Electronic Business XML.

EDI : Electronic Data Interchange.

FTP : File Transfert Protocol.

JSP : java Server Page.

HTML : HyperText Markup Language.

LMS : Learning Management System.

LCMS : Learning Content Managemnt System.

OASIS : Organization for the Advancement of Structured Information Standards.

OSI : Open Systems Interconnexion.

RMI : Remote Method Invocation.

RPC : Remonte Procedure Call.

SMTP : Simple Mail Transfer Protocol.

SOAP : Simple Object Acces Protocol.

TCP/IP : Transmission Control Protocol /Internet Protocol.

TTTP : HyperText Transfert Protocol.

URL : Uniform Ressource Locator.

URI : Uniform Resource Identifier.

UML : Unified Modeling Language.

UDDI : Universal Description, Discovery and Integration.

WSDL : Web Service Description Language.

W3C : World Wide Web Consurtium.

WS-I : Web Services Interoperability.

WSCL : Web Services Conversation Language.

WSFL : Web Services Flow Language.

XML : eXtensible Markup Language.

Introduction Générale

De nos jours, il est indispensable pour une entreprise de suivre les progrès techniques et les nouvelles innovations technologiques dans le but de s'agrandir et de rester concurrentielle. Mais, cette extension est elle-même un défi majeur pour ces entreprises, dû aux faits des difficultés de coordination de tous leurs sites distants.

C'est dans ce contexte que le service maintenance des travaux de NAFTAL a décidé d'endiguer ce problème de coordination, en œuvrant à la mise en place d'une application qui puisse être déployé sur un serveur et accessible depuis tous les sites distants.

Notre projet consiste en la mise en œuvre d'une application répartie qui procurera un environnement convivial aux différents acteurs afin de répondre aux contraintes de fiabilité, efficacité et surtout de disponibilité.

Pour la réalisation de notre projet, nous avons opté pour une application basée sur les web services implémentée sur la plateforme J2EE.

Notre travail est structuré en deux parties: l'état de l'art et conception & réalisation.

L'état de l'art : cette partie est composée de deux chapitres :

Le 1er chapitre est la présentation de la technologie des services Web.

Le 2ème chapitre est la présentation du J2EE.

Conception & Réalisation : cette partie inclut deux chapitres :

Analyse et conception : Ce chapitre s'intéresse à l'étude conceptuelle du système à développer. Dans ce cadre, nous exposons la démarche pour la conception de l'architecture de notre application, ensuite nous détaillons les acteurs impliqués et les fonctionnalités à développer dans cette plateforme.

La réalisation : Chapitre relatif à l'application que nous avons développée; dans lequel on retrouvera la description détaillée de notre solution, les outils utilisés pour la développer et quelques interfaces réalisées.

Des annexes ont été ajoutées afin d'apporter de plus amples informations sur certains points décrits dans le rapport.

Chapitre I

Etat de l'art sur les web services

I- Introduction

Le web se présente actuellement comme un espace immense d'informations, dans lequel l'individu aura certainement besoin des machines pour l'aider à s'y retrouver; or la structure actuelle du web ne facilite pas cette tâche.

Il existe sur le Web bon nombre d'applications, qui peuvent aider ses utilisateurs à y voir un peu plus clair dans cet amalgame d'informations, ce qui a amené à rechercher un moyen sophistiqué pour permettre l'interopérabilité entre ces différentes applications quelque soit leurs plates-formes de développement ou leurs systèmes utilisés.

Les web services apportent une solution simple au problème d'interopérabilité des systèmes informatiques, leur mise en place facilite le dialogue entre les différentes applications. Ils peuvent aussi être implémentés sur diverses plates-formes, systèmes d'exploitation et avec des langages variés.

Les web services ont pour objectif de repousser toutes les limitations des technologies de mise en œuvre de l'informatique répartie.

II- Historique des web services [01]

L'histoire des web services commence en 1975, lorsque l'informatique souffrait encore de peu de standardisation et que constructeurs et éditeurs se rendent compte de la nécessité d'uniformiser les échanges de données. Ils firent alors les vœux pieux de "l'interopérabilité" afin de standardiser la communication entre applications au travers d'un réseau. C'est la naissance de l'EDI, l'ancêtre des Web Services.

De nombreuses technologies ont alors émergé depuis avec toujours cet objectif de connecter des logiques métiers au travers d'un réseau: c'est la naissance de CORBA, DCOM, Unix RPC, Java RMI, mais aucune de ces technologies n'a réellement réussi à s'imposer comme standard universel car souvent rattaché à un système d'exploitation, à un éditeur ou à un langage particulier.

L'arrivée du web a tout bouleversé. Jusque-là, toutes les tentatives d'accorder les acteurs du marché autour d'un protocole réseau d'interconnexion ont échoués. Le web a rendu cela possible en s'appuyant sur des modèles académiques (modèle OSI) et en reprenant les clés du succès des protocoles de bas niveaux déjà adopté.

L'émergence du Web va passer par la création du protocole HTTP et sa standardisation en 1997 comme protocole de transfert de données hyper texte (du texte avec

des liens) et qui s'appuiera sur TCP/IP. A ce moment, IP est un protocole d'interconnexion mature et dont la diffusion était importante dans les universités et les entreprises. Il ne manquait alors qu'un format d'encapsulation et d'échange de messages pour finaliser la chaîne de l'interopérabilité. Ce qui fut réalisé avec l'arrivée d'XML et la naissance de XML-RPC.

En 1998, année de sa standardisation en version 1.0 par le W3C, XML viendra compléter la chaîne de l'interopérabilité en s'imposant comme format d'échange et de description des données, indépendant de toute plateforme.

III- Définitions des Services Web

Les web services fournissent une nouvelle manière de développer des applications conformes aux besoins des réseaux, et ils donnent la solution la plus adaptée pour assurer l'interopérabilité, qui permet de transmettre les données entre les différentes applications d'une organisation (entreprise, société).

Il existe plusieurs définitions pour les services web mais la plus simple pourrait être "fonctionnalité utilisable à travers un réseau en mettant en œuvre un format standard, généralement utilisant XML"[Mel04].

Un web service est un ensemble de composants applicatif utilisés pour échanger les données entre les applications. C'est un composant logiciel représentant une fonction applicative (ou un service applicatif). Il peut être accessible depuis une autre application (un client, un serveur ou un autre web service) à travers un réseau en utilisant les protocoles de transports disponibles. Ce service applicatif peut être implémenté comme une application autonome ou comme un ensemble d'applications (liées ensemble par une infrastructure d'intégration). [Ric04]

IV- Cycle de vie d'utilisation des web services

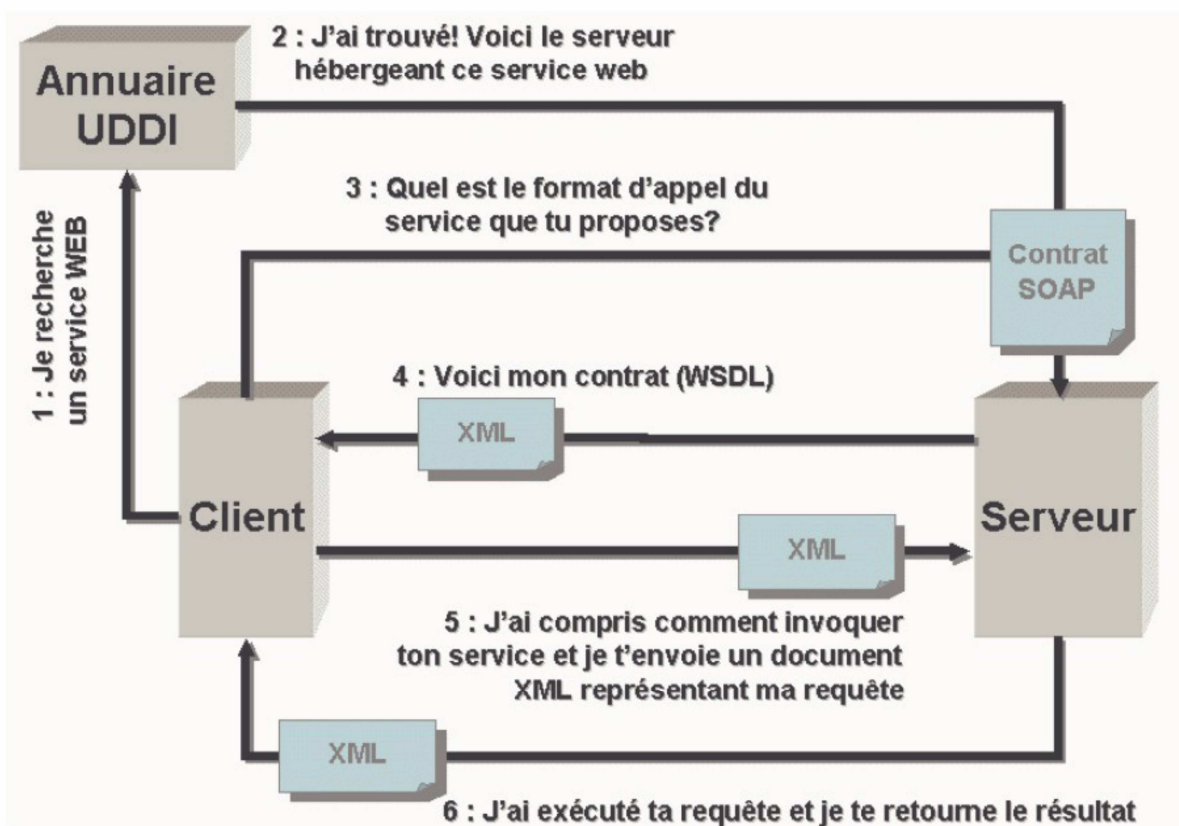


Figure I-1 : Cycle de vie d'utilisation des web services

V- Architecture des web services

L'architecture des web services est une architecture orientée-composants ou encore une architecture orientée-services SOA (Service Oriented Architecture), elle vise trois objectifs importants :

- l'identification des composants fonctionnels ;
- définition des relations entre ces composants ;
- l'établissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture. [WSS04] [03]

Cette architecture est constituée de trois éléments :

- **Le fournisseur de service :**

C'est le propriétaire du service. D'un point de vue technique, il est constitué de la plateforme d'hébergement du service. Le plus grand intérêt de ce modèle est de fournir un

accès à des applications particulières en utilisant un protocole standard comme le protocole http.

Il existe quatre types de fournisseurs :

- ✓ **spécialiste** ou **fonctionnel** : fournit une application seule, comme le processus de paiement par carte de crédit ;
- ✓ **vertical** : fournit une solution sous forme de progiciel pour un type de client spécifique comme les techniciens d'un service donné;
- ✓ **entreprise** : fournit un spectre complet de solutions ;
- ✓ **local** : fournit aux PME des services dans une zone géographique limitée ;

- **Le client :**

C'est le demandeur de service. Techniquement, il est constitué de l'application qui va rechercher et invoquer un service. Une application cliente peut être elle-même un web service.

- **L'annuaire des services :**

C'est un registre de descriptions de services offrant des facilités de publication de services pour les fournisseurs de services ainsi que des facilités de recherche de services pour les clients.

Ces trois éléments de l'architecture interagissent entre eux selon trois types d'opérations : les opérations de publication, de recherche et de liens.

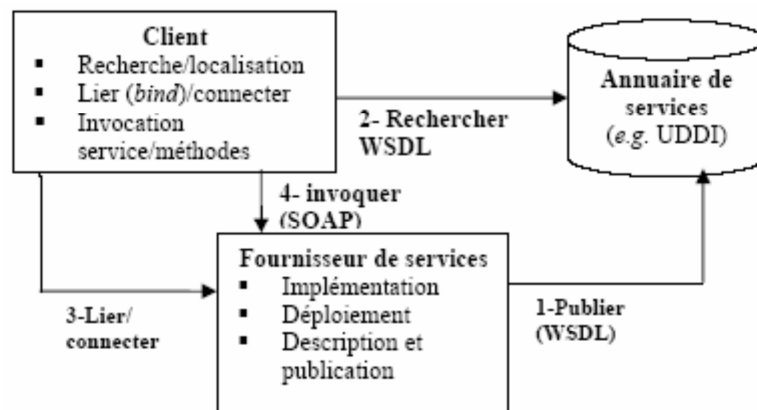


Figure I-2 : Architecture des web services[WSS04]

1- Le fournisseur de services définit la description de son service et la publie dans un annuaire de service UDDI.

La publication : Les services Web peuvent être publiés pour des clients potentiels de diverses manières :

- ✓ à l'aide d'un fichier de découverte XML, doté d'une extension .disco ;
- ✓ à l'aide d'une URL qui spécifie une extension .vsdisco ;

Le déploiement : Le déploiement d'un service Web implique la copie dans un répertoire virtuel sur un serveur Web du fichier .asmx. Supposons par exemple que vous ayez un service Web appelé StockServices. Pour le déployer, vous créez un répertoire virtuel sur votre serveur Web et placez le fichier .asmx du service dans ce répertoire:[14]

```
\Inetpub
  \Wwwroot
    \StockServices
      StockServices.asmx
    \Bin
```

La description : Il s'agit d'une tentative de normalisation regroupant la description des éléments permettant de mettre en place l'accès à un service réseau (Service Web). Il fait notamment référence au langage XML et a été proposé en 2001 au W3C pour standardisation.[13]

2- Le client utilise les facilités de recherche disponibles au niveau de l'annuaire pour retrouver et sélectionner un service donné.une fois sa description est reçue. En exploitant les informations de cette description, le client peut générer des requêtes SOAP pour invoquer le service.

3- Le client récupère ensuite les informations nécessaires sous format WSDL, à partir de la description du service sélectionné, lui permettant de se connecter au fournisseur du service et d'interagir avec l'implémentation du service considéré.

4- La communication entre le demandeur de service et le fournisseur est assurée par le SOAP et les autres protocoles de communication. Le demandeur de service envoie une requête SOAP vers le fournisseur de service, cette requête est véhiculée par le HTTP jusqu'au fournisseur. Ensuite le web service du fournisseur de service renvoie sa réponse au demandeur sous la forme d'un document XML via SOAP etHTTP.

VI- XML et les trois standards SOAP, WSDL, UDDI

VI.1 - XML

VI.1.1 - Définition

Le XML créé en 1999, est resté durant près de deux ans, un concept plutôt abstrait et théorique faute de moyens fiables pour en afficher le résultat. Avec le développement des nouvelles techniques comme le XSL, il est devenu possible de percevoir concrètement les énormes potentialités de ce nouveau langage.

Le XML est un langage de balises comme le Html mais il est évolutif. En XML, les balises ne sont pas prédéfinies. C'est l'utilisateur qui définit ses propres balises.

XML a été conçu pour des documents complexes, basé sur 5 grands principes :

- Lisibilité par les machines et les utilisateurs
- Définition sans ambiguïté du contenu d'un document
- Définition sans ambiguïté de la structure d'un document
- Séparation entre documents et relations entre documents
- Séparation entre structure du document et présentation du document.

Le format XML a été conçu pour permettre le déchiffrement direct par l'utilisateur comme par les programmes.

Le contenu d'un document est décrit par une succession d'éléments (blocs de texte encadrés par deux paires de balises ouvrantes et fermantes) qui sont les unités du contenu. Ces éléments sont liés entre eux par une hiérarchie, certains éléments apparaissent imbriqués dans d'autres.

VI.1.2 - Quelques différences entre le HTML et le XML

Le HTML et le XML sont différents en de très nombreux points comme :

Le XML	Le HTML
° Le XML décrit, structure, stocke, transporte et échange des données.	° Le Html affiche des données par l'intermédiaire d'un navigateur.
° Le XML est un générateur de langages [métalanguage].	° Le Html est un langage statique (normalisé) de publication sur le Web.
° Outre les PCs, le XML se veut adapté aux outils comme les mobiles, les pockets.	° Le Html est surtout conçu pour les ordinateurs.
° Le XML est un langage strict dont l'écriture doit être rigoureuse.	° Le Html, à cause des navigateurs récents, est devenu très permissif.

Exemple XML

< ?xml version="1,0" standalone= "yes" ? >	Document autonome
< ?xml :stylesheet type="text/xsl"href="style.xsl" ?>feuille de style associée	
<Listepersonne>	Un employé est défini par
<personne sexe="M" âge =" 34">	- son(s) nom(s)
<nom>Khelifi</nom>	- son(s) prénom(s)
<prenom>Axel</prenom>	- son âge et son sexe
</personne>	
<personnesexe="M" âge =" 24">	
<nom>Kebaili</nom>	
<prenom>Alex</prenom>	
</personne></Listepersonne>	

La première ligne de cet exemple est la déclaration XML, elle contient les informations suivantes :

Version='1.0' : version du XML utilisé dans le document.

Encoding='ISO-8859-1' : le jeu de codage de caractère utilisé.

L'élément racine (en anglais : document element) de notre exemple est personne. Il s'ouvre juste après la déclaration XML et se ferme à la fin du document. Il est unique et englobe quatre éléments enfants : nom, prénom, sexe et âge, qui contiennent du texte.

VI.2 - Protocol SOAP [04]

VI.2.1 - Définition

SOAP est un protocole de RPC orienté objet bâti sur le XML. Il permet la transmission de messages entre objets distants, autrement dit, il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Cela se fait à l'aide du protocole HTTP surtout, mais également avec d'autres protocoles, comme SMTP.

Il comporte trois composantes principales qui sont : un framework de messagerie, un standard d'encodage et le mécanisme RPC. [Tid 01]

✓ Le framework de messagerie SOAP :

Le framework de messagerie SOAP requiert que le message SOAP soit composé d'une enveloppe qui contient un Header et un Body. Le Header comporte les métas – données du message et le Body comporte le corps du message lui-même.

✓ Encodage / sérialisation standard pour les objets : Ce standard permet l'encodage des objets dans des messages SOAP, ainsi que leurs décodage au niveau du destinataire

Le message SOAP est composé de deux parties :

- L'entête (balise Header) qui est facultative et qui peut contenir les informations d'authentification, de sécurité (chiffrement, certificats, billets de sessions). Elle peut contenir d'autres directives qui règlent les appels et les messages.
- Le corps (balise Body) qui est obligatoire et contient l'information sur l'appel de la procédure distante ou sur le message XML envoyé.

```
<soap:Envelope xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'  
soap:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'>  
<!-- Entête et corps-->  
</soap:Envelope>
```

VI.2.2 - Invocation de services d'objets distants via SOAP RPC

SOAP permet à l'aide de XML, la réalisation d'un appel RPC. XML n'est pas utilisé pour transporter des documents mais pour véhiculer des appels de procédure et leur résultat. Les différentes étapes d'invocation d'un objet distant sont illustrées dans le schéma suivant :

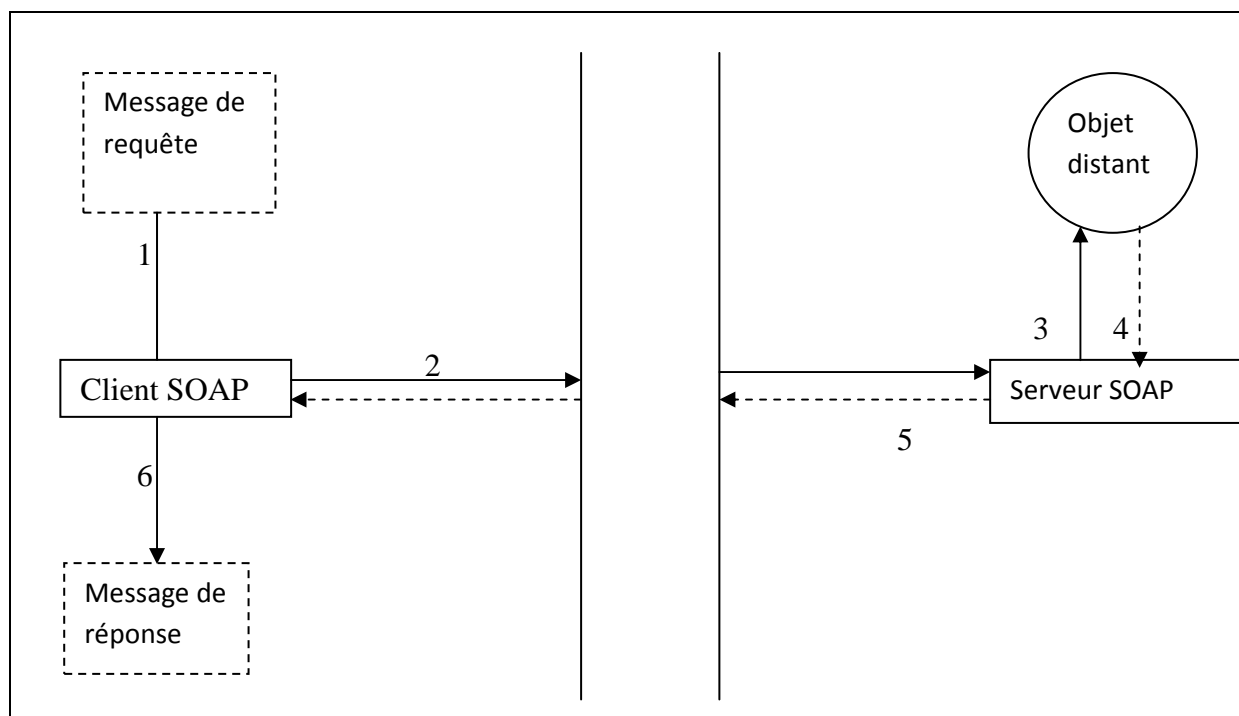


Figure I-3: Etapes d'invocation d'objets distant avec SOAP.

1. Le client SOAP crée un document XML qui contient les informations nécessaires pour invoquer les services d'un objet distant. Ensuite ce document est inséré dans une enveloppe SOAP avant d'être transmis sous forme d'une requête HTTP.
2. Le message est transmis via une connexion HTTP.
3. Le message est reçu et analysé par le serveur SOAP, ensuite envoyé à l'objet distant.
4. L'objet fait le traitement de la requête et envoie la réponse au serveur SOAP.
5. La réponse est envoyée, sous forme d'un document SOAP, au client via le HTTP.
6. Le client reçoit la réponse, ouvre l'enveloppe et envoie le résultat au demandeur initial.

Requête SOAP

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Header/>
<S:Body>
<ns2:afficherdt xmlns:ns2="http://ws3/" />
</S:Body>
</S:Envelope>
```

Réponse SOAP

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:afficherdtResponse xmlns:ns2="http://ws3/">
<return>
<datedt>23 sept. 2012</datedt>
<etatdt>0</etatdt>
<iddt>3</iddt>
<idequip>RÃ©paration</idequip>
<idpanne>a</idpanne>
<idpersonne>1</idpersonne>
<idstation>station3</idstation>
<indexdt/>
<remarque>hdgcjhedg</remarque>
</return>
<return>
<datedt>12 sept. 2012</datedt>
<etatdt>1</etatdt>
<iddt>1</iddt>
<idequip>RÃ©paration</idequip>
<idpanne>a</idpanne>
<idpersonne>1</idpersonne>
<idstation>station3</idstation>
<indexdt/>
<remarque>sdds</remarque>
</return>
</ns2:afficherdtResponse>
</S:Body>
</S:Envelope>
```

VI.2.3 - Structure d'un message SOAP

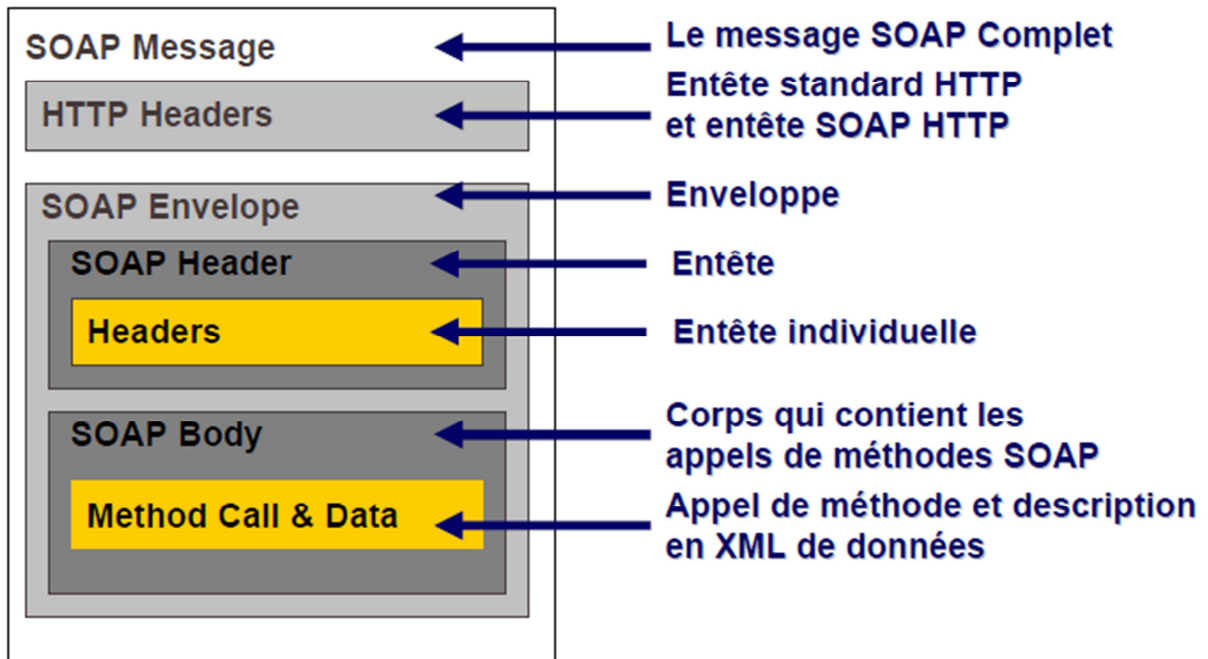


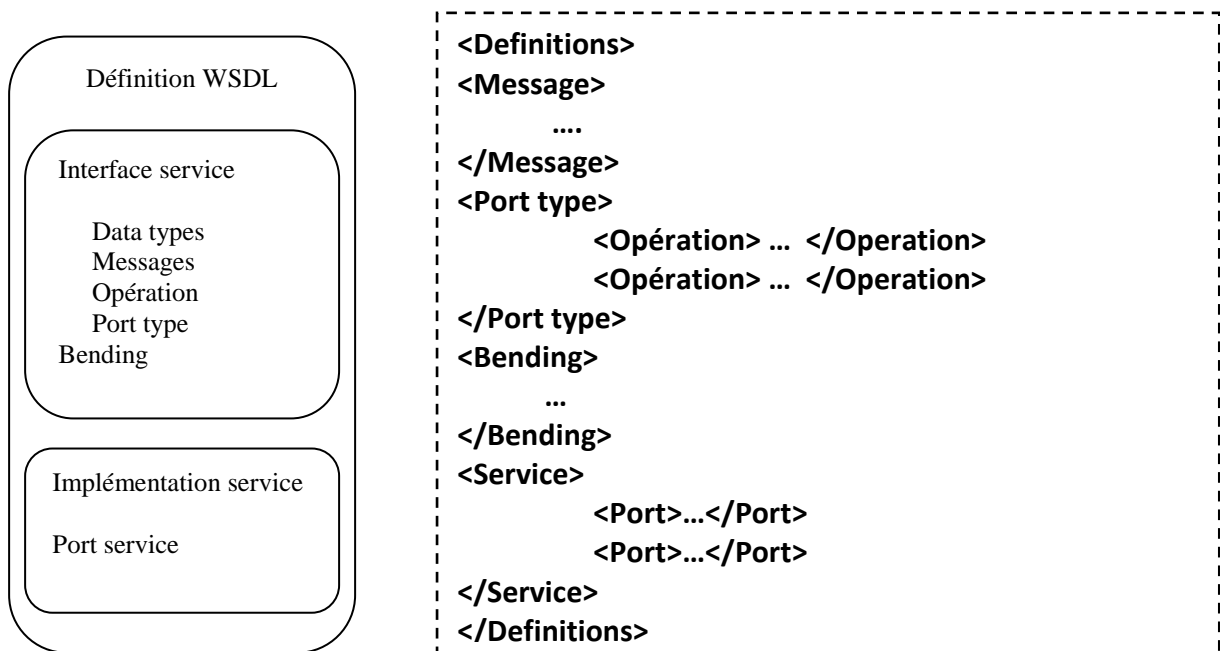
Figure I-4:structure de message SOAP. [HUK03][VM03]

- **L'enveloppe** : C'est un élément obligatoire dans un message SOAP, elle permet de spécifier la version de SOAP utilisée, ainsi que de spécifier les règles d'encodage (sérialisation et désérialisation) mises en œuvre dans le message.
- **Entête** : Optionnel, pour spécifier certaines directives pour le traitement du message.
- **Corps** : Contient les données.

VI.3 - WSDL

WSDL décrit qu'est-ce que le service Web, où le trouver (serveur) et comment l'appeler. Il permet aussi de connaître les protocoles, les ports ainsi que le format des messages.

VI.3.1- La structure d'un document WSDL



- **balise Définitions** : elle contient le nom du service décrit et les namespaces faisant référence aux types utilisés dans le document.
- **balise Message** : une abstraction du plus haut niveau de paramètres (messages) pour les appels possibles au service Web
- **balise Port type** : contient les classes accessibles. Si le service propose différentes classes, leurs noms doivent être différents.
- **balise Binding** : définit les liens entre les opérations du service Web et le protocole de communication utilisé (généralement SOAP). C'est ici qu'on définit comment on utilise SOAP, les paramètres d'entrées et de sorties.
- **les balises service et port** : permettent de spécifier le nom, où se trouve la documentation concernant le service mais aussi où envoyer les messages d'invocation. Le port spécifie l'adresse (URI) à laquelle un service peut être invoqué. L'élément binding permet d'associer le nom du port au binding du fichier WSDL.

VI.4 - L'Annuaire UDDI

VI.4.1- Définition

L'annuaire UDDI permet aux utilisateurs de faire la publication des applications (des web services). En effet, UDDI peut contenir des informations sur les fournisseurs et les services qu'ils publient.

L'inscription d'un fournisseur de services à l'annuaire UDDI lui permet de se présenter et présenter ses services, l'adoption de cet annuaire par les fournisseurs permet l'accélération des échanges surtout les échanges commerciaux de type B2B. L'enregistrement des web services dans un annuaire UDDI, s'effectue auprès d'un opérateur en accédant au site web de ce dernier à partir d'un navigateur ou d'un outil intégré dans environnement de développement. Des recherches précises peuvent s'effectuer dans l'annuaire par catégorie de fournisseurs en utilisant des standards de taxinomie et d'identification de fournisseurs. [WEE.05].

VI.4.2- Pages de l'annuaire

Les données retrouvées dans l'UDDI sont divisées en trois catégories [HUB03] :

- Pages blanches : le référentiel comporte des informations sur les fournisseurs de services telles que le nom et les coordonnées du fournisseur.
- Pages jaunes : le référentiel comporte des critères de catégorisation de services, les critères de catégorisation s'appuient sur des standards de classification de fournisseurs, les services sont décrits par des documents au format WSDL. Un fournisseur peut disposer de plusieurs entrées dans l'annuaire pour l'ensemble des différents services et produits qu'il publie.
- Pages vertes : le référentiel comporte des informations techniques (WSDL) détaillée sur les services fournis telles que les informations sur les processus métier, les descriptions de services et les informations de liaison sur les services.

Les annuaires offrent des mécanismes standards de classifications. Ils peuvent être publics ou privés

- Les annuaires publics sont hébergés par des sociétés comme IBM ou Microsoft.
- Les annuaires privés peuvent être hébergés par une société quelconque sur un réseau privé ou sur internet.

NB : Les annuaires publics sont moins développés que les annuaires privés en terme de sécurité.

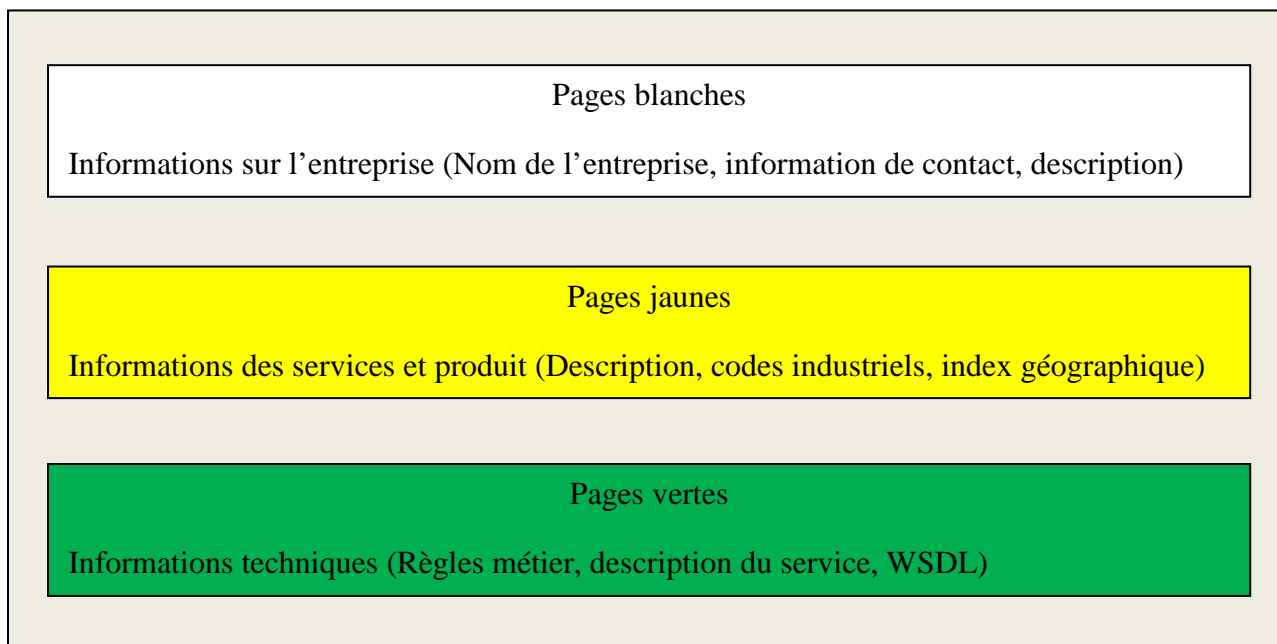


Figure I-6 : Pages de l'annuaire UDDI.

VI.4.3 - Mécanismes d'accès aux services fournis par l'UDDI

La communication (requêtes / réponses) avec un annuaire UDDI repose sur le protocole de transport SOAP [WEE.05]

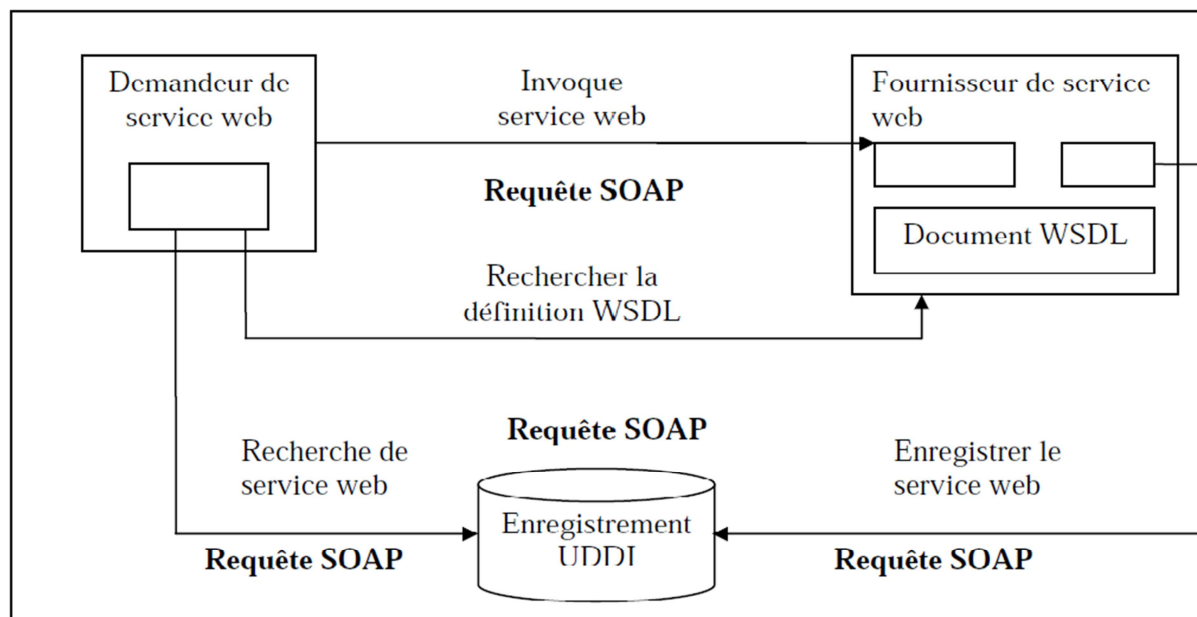


Figure I-7: Mécanismes d'accès aux services de UDDI.

Un enregistrement UDDI, a deux types de clients :

- Les fournisseurs de services.
- les utilisateurs de ces services.

UDDI comporte des pages qui fournissent des informations sur les fournisseurs (nom, coordonnées,...etc.) et des pages qui comportent la description, au format WSDL, des web services. Un fournisseur peut disposer de plusieurs entrées dans l'annuaire pour l'ensemble de services qu'il propose, et des pages qui disposent d'informations techniques détaillées sur les produits proposés.

VI.4.4- Le modèle de données dans UDDI

Le modèle de données UDDI est défini sous forme de schéma W3C XML Schéma, ce schéma XML comporte cinq structures de données principales ; [WEE05] [NEW04]

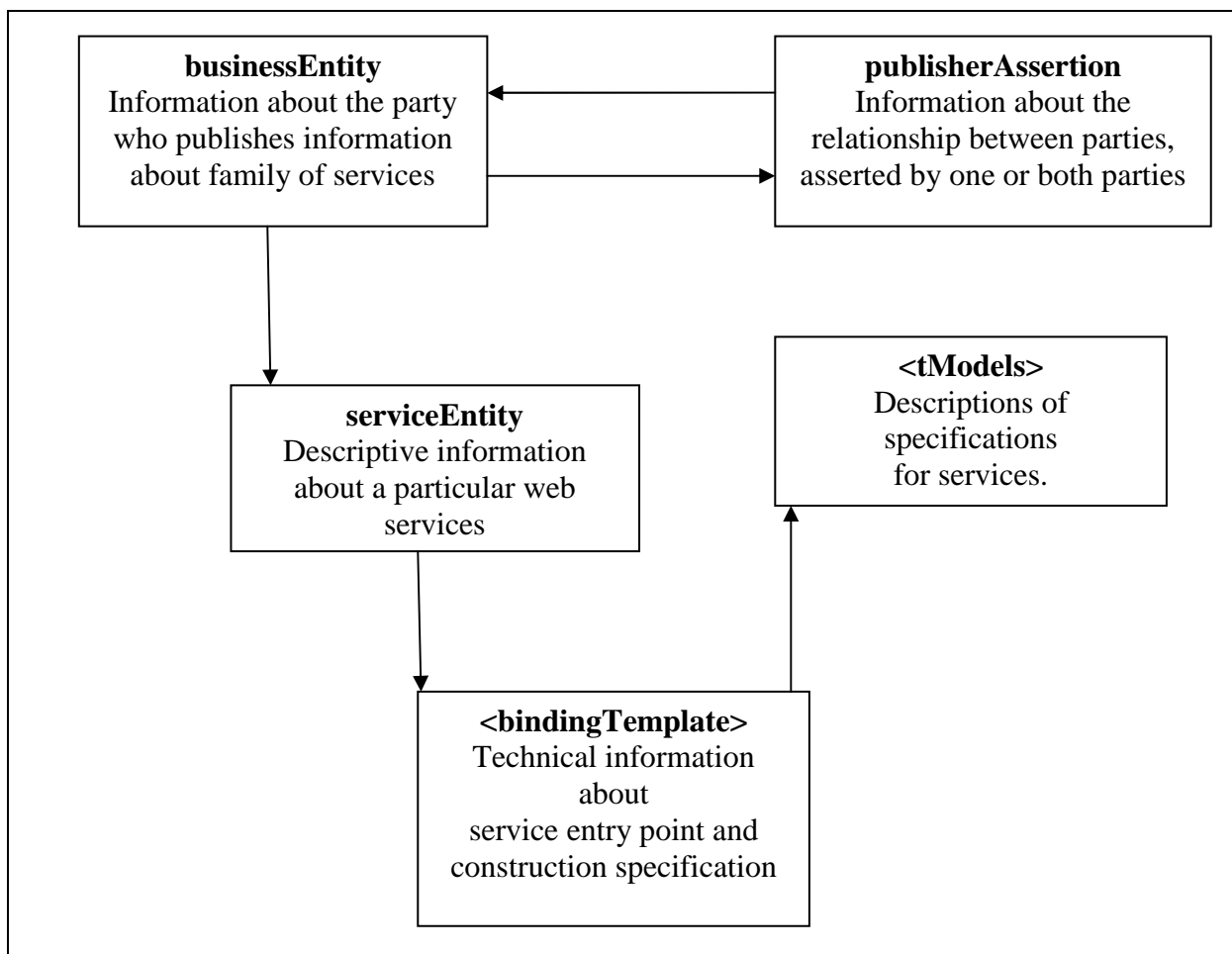


Figure I-8: Modèle de données de l'annuaire UDDI.

- **BusinessEntity** : chaque businessEntity est identifiée par une «businessKey». Les «businessEntities» sont en quelque sorte les pages blanches d'un annuaire UDDI, elles comportent des informations sur les fournisseurs de services ayant publié des services dans l'annuaire. On y trouvera notamment le nom de l'organisation, ses adresses(physiques et web), des éléments de classification, une liste de contacts, ...etc.
- **ServiceEntity** : ce sont en quelque sorte les pages jaunes d'un annuaire UDDI, qui décrivent de manière non technique les services proposés par les différents fournisseurs. On y trouvera essentiellement le nom et la description textuelle des services ainsi qu'une référence à l'organisation proposant le service et un ou plusieurs«bindingTemplates».
- **BindingTemplate (coordonnées des services)** : ce sont des informations qui concernent le lieu d'hébergement du service, elles donnent les coordonnées des services. UDDI permet de décrire des web services HTTP, mais également des services invoqués par d'autres moyens (SMTP, FTP, fax, téléphone,...etc.). Elles contiennent notamment une description, la définition du « point d'accès » (suivant les cas, une URL, un numéro de téléphone, ...) et les éventuels « tModels » associés.
- **tModel (descriptions techniques des services)** : ce sont des informations qui concernent le mode d'accès au services, ce sont les descriptions techniques des services. UDDI n'impose aucun format pour ces descriptions qui peuvent être publiées sous n'importe quelle forme et notamment sous forme de documents textuels(XHTML par exemple). C'est à ce niveau que WSDL intervient comme le vocabulaire de choix pour publier des descriptions techniques de services.
- **publisherAssertion** : assertions contractuelles entre partenaires dans le cadre des échanges d'exécution d'un service.

VI.4.5 - L'interface UDDI [NEW04]

L'interface UDDI est définie sous forme de documents UDDI et implémentée sous la forme de web service SOAP. Elle est composée des modules suivants :

- **Interrogation** : Cette interface permet de rechercher des informations dans un répertoire UDDI et de lire les différents enregistrements enregistrés suivant le modèle de données UDDI.
- **Publication** : Cette interface permet de publier des informations dans un répertoireUDDI conformément à son modèle de données.

- **Sécurité** : Cette interface est utilisée pour obtenir et révoquer les jetons d'authentification nécessaires pour accéder aux enregistrements protégés dans un annuaire UDDI.

- **Contrôle d'accès et propriété** : Cette interface permet de transférer la propriété d'informations (qui est à l'origine attribuée à l'utilisateur ayant publié ces informations) et de gérer les droits d'accès associés.

- **Abonnement** : Cette interface permet à un client de s'abonner à un ensemble d'informations et d'être avertis lors des modifications de ces informations. Tous les répertoires UDDI doivent gérer un avertissement par polling (le client interroge le serveur pour savoir si des modifications ont eu lieu sur les données auxquelles il est abonné). Une fonctionnalité optionnelle est également prévue permettant au client de communiquer au serveur la définition d'un web service sur lequel il souhaite être prévenu en cas de modification.

- **Réplication interne (Nœuds d'un même annuaire)** : A côté des interfaces utilisateurs que nous venons de voir, UDDI définit également l'interface permettant de synchroniser les nœuds d'un même annuaire UDDI.

- **Réplication externe (interrogation, publication, abonnement)** : La réplication externe par duplication d'informations entre différents annuaires UDDI n'a pas donné lieu à la définition d'une interface spécifique mais se fait en utilisant les interfaces d'interrogation (pour la lecture dans un annuaire), publication (pour la publication dans un autre annuaire) et éventuellement abonnement (pour pouvoir propager les modifications ultérieures).

VI.4.6 - L'usage de l'annuaire UDDI [NEW04]

L'annuaire UDDI permet, la publication et la découverte des web services:

1. **La publication de services web**: Un web service peut être publié en publiant sa description, après sa production bien sur. Cette description peut être générée manuellement ou automatiquement à l'aide des outils qui peuvent générer des parties WSDL et de créer des entrées dans UDDI à partir de méta données par exemple. Un UDDI peut être de plusieurs types et leurs utilisations dépendent du domaine des services à publier, à savoir :

- Nœud UDDI pour une application interne : les nœuds UDDI se trouvent derrière le firewall, avec ce type de nœud on a plus de contrôle sur l'enregistrement, l'accessibilité, la disponibilité et les spécifications de publication lors de la publication de service.

- Nœud portail UDDI : il se trouve à l'extérieur du firewall du fournisseur de service ou entre les firewalls. Sur ce type de nœud on publie des web services pour les partenaires externes tout en implémentant des mécanismes d'accès sélectifs selon le profil des utilisateurs.
- Nœud catalogue du partenaire avec UDDI : les web services peuvent être publiés sur un nœud du catalogue du partenaire avec UDDI (derrière le firewall). Le partenaire est choisi avec une autorisation d'accès spécifique.
- Nœud de place de marché UDDI : il s'agit de relations interentreprises et de contrôle du partage de l'information entre systèmes d'information.

2. La découverte de services web: C'est la recherche et la localisation d'un web service particulier dans un annuaire de services décrivant le nom du fournisseur, l'objectif de chaque services. Donc il s'agit de l'acquisition des descriptions de web services et leur utilisation.

VI.4.7- Opérateurs d'annuaires UDDI actuels

Opérateurs	Adresse du site
HP	http://uddi.hp.com/uddi/index.jsp
IBM	http://www.3.ibm.com/services/uddi/v2beta/protect/registry.html
Microsoft	http://uddi.microsoft.com/register.aspx
NTT	www.ntt.com
SAP	www.sap.com

VII. Architecture en couches des standards des services Web

Le fonctionnement des services web repose sur un modèle en couches, dont les trois couches fondamentales sont les UDDI, WSDL, SOAP, initié de la couche service et processus composable et la couche service de composition.



Figure I-9: Architecture en couches des protocoles et standards des services Web [KRC 01]

I- Caractéristique des web services [HUKO3][VM][02]

- Une infrastructure d'échange de messages entre applications et entre clients et applications.
- Permet d'offrir une application comme toute autre ressource adressable par une URL notamment sur Internet.
- Les clients et les applications n'ont pas à savoir comment un service est implémenté.
- Assurent à travers le réseau internet, l'interaction entre les applications, les ordinateurs et les processus métier via les protocoles internet comme HTTP, SMTP.
- La réutilisation d'applications
- La distribution d'applications
- Assurent l'interconnexion de services logiciels en se basant sur des standards facilitant le transport, l'invocation, la description et la recherche des web services comme : SOAP, HTTP, WSDL, UDDI.

- Permettent d'intégrer, gérer et automatiser rapidement les processus métier intra et inter entreprises en échangeant des informations au format XML.

VIII. Avantage des web services

L'idée essentielle derrière les services Web est de partager les applications et les programmes en un ensemble d'éléments réutilisables appelés « service », de sorte que chacun de ces éléments effectuent une tâche principale et efficace, afin de faciliter l'interopérabilité entre tous ces services. D'autre part les services Web :

- ✓ Permettent l'interopérabilité dans des environnements applicatifs, cela veut dire que les logiciels écrits dans différents langages de programmation, et évoluant sur différents systèmes d'exploitation peuvent communiquer et/ou échanger des données entre eux facilement.
- ✓ Permettent de profiter de différents environnements et langages de développement par une publication, localisation, description et une invocation, donc les services Web sont très flexibles et indépendants des langages de programmation et des systèmes d'exploitation.
- ✓ Permettent d'accéder aux applications à travers les pare-feux (les firewalls) à l'aide d'utilisation des protocoles Internet standards comme HTTP sur le port 80, qui est généralement ouvert, leur permet d'assurer une transmission des données transactionnelles et sécurisées.
- ✓ Utilisables à distance via n'importe quel type de plateforme, et sont accessibles depuis n'importe quel type de clients.

IX. Apport des Web services aux entreprises [Bry04]

- ✓ Donner aux clients un accès direct à l'information, aux données et aux fonctionnalités dont ils ont besoin pour interagir avec une entreprise.
- ✓ Donner aux partenaires d'une entreprise un accès direct à la fonctionnalité dont ils ont besoin pour mieux servir les clients qu'ils ont en participation avec cette entreprise.
- ✓ Donner aux fournisseurs d'une entreprise un accès direct à l'information et la fonctionnalité dont ils ont besoin pour leur permettre d'ajuster les inventaires.
- ✓ Intégrer efficacement des applications de bout en bout, de manière abordable, meilleur et facile à implanter autant hors des frontières de l'entreprise qu'à l'intérieur de l'entreprise.

- ✓ Les services Web offrent donc aux entreprises la flexibilité de réponse et d'anticipation des besoins changeant des clients, la normalisation des infrastructures logicielles, la flexibilité d'interaction et de configuration des accords externes avec les partenaires et fournisseurs.

X. Limites des web services

- Ils ne permettent pas de spécifier le sens à donner au document.
- Les web service ne sont qu'un mécanisme de transfert de données/ d'informations d'un système à l'autre. N'apportent, en aucun cas, plus de valeurs à l'information déjà possédée.
- La consommation automatique de services pour des fonctions critiques peut être néfaste pour l'entreprise
- Les web services ne remplacent pas les démarches commerciales effectuées auprès des partenaires de la société.

XI. Composition des web Services

XII-1. Définition

Un service web est dit *composé* ou *composite* lorsque son exécution implique des interactions avec d'autres services web afin de faire appel à leurs fonctionnalités. La composition de services web spécifie quels services ont besoin d'être invoqués, dans quel ordre et comment gérer les conditions d'exception. La composition des Services Web peut se faire de deux manières: orchestration et chorégraphie [NEW 04].

➤ **Orchestration**

L'orchestration décrit l'interaction des services au niveau de messages, incluant la logique métier et l'ordre d'exécution des interactions. Les services web n'ont pas de connaissance (et n'ont pas besoin de l'avoir) d'être mêlés dans une composition et d'être partie d'un processus métier. Seulement le coordinateur de l'orchestration a besoin de cette connaissance.

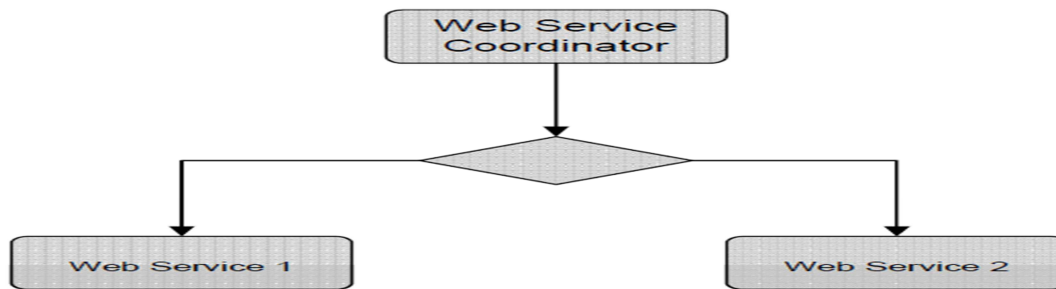


Figure I-10: Web Service Orchestration [WEE05]

➤ **Chorégraphie**

Contrairement à l'orchestration, la chorégraphie n'a pas un coordinateur central. Chaque service web mêlée dans la chorégraphie connaît exactement quand ses opérations doivent être exécutées et avec qui l'interaction doit avoir lieu.

La chorégraphie est un effort de collaboration dans lequel chaque participant du processus décrit l'itération qui l'appartient. Elle trace la séquence des messages qui peut impliquer plusieurs Services Web.

La collaboration dans la chorégraphie des services web peut être représenté de la manière suivant:

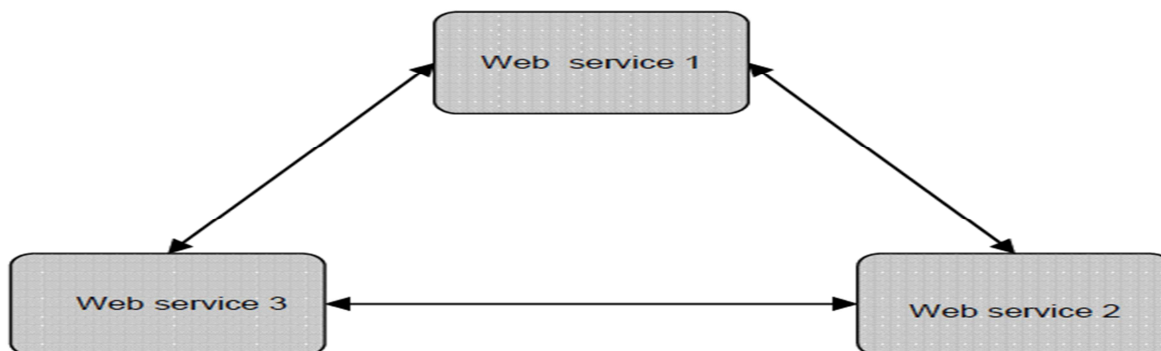


Figure I-11: Web Services chorégraphie

XII-2. Types de composition

La composition des services web peut être classifiée en trois catégories [Fost03]

XII-2.1. Composition manuelle

La composition manuelle des services Web suppose que l'utilisateur génère la composition à la main via un éditeur de texte et sans l'aide d'outils dédiés.

XII-2.2. Composition semi-automatique

Les techniques de composition semi-automatiques sont un pas en avant en comparaison avec la composition manuelle, dans le sens qu'ils font des suggestions sémantiques pour aider à la sélection des services Web dans le processus de composition.

XII-2.3. Composition automatique

La composition totalement automatisée prend en charge tout le processus de composition et le réalise automatiquement, sans qu'aucune intervention de l'utilisateur ne soit requise.

XIII. Langages de définition de la composition des web services

Ils existent plusieurs langages de définition pour la composition de services web. Dans cette section nous allons présenter brièvement quelques uns d'entre eux.[05] [06]

XIII-1. Business Process Modeling Language (BPML)

Le BPML est un langage de modélisation des processus métiers. Il permet de définir un modèle abstrait d'interaction entre collaborateurs participant à une activité de l'entreprise, voire entre une organisation et ses partenaires.

Les processus métiers sont représentés par un flux de données, un flux d'événements sur lesquels on peut influencer en définissant des règles métier, des règles de sécurité, des règles de transactions. On peut ensuite lancer l'exécution du modèle et vérifier le fonctionnement théorique des différents processus.

XIII-2. Web Service Choreography Interface (WSCCI)[07] [08]

WSCCI est un langage reposant sur XML. Il propose de se focaliser sur la représentation des services Web en tant qu'interfaces décrivant le flux de messages échangés (la chorégraphie de messages).

Il propose ainsi de décrire le comportement externe observable du service. Pour cela, WSCCI propose d'exprimer les dépendances logiques et temporelles entre les messages échangés à l'aide de contrôles de séquences, corrélation, gestion de fautes et transactions.

XIII-3. Web Service Conversation Language (WSCL) [07] [08]

WSCL propose de décrire à l'aide de documents XML les services Web en mettant l'accent sur les conversations de ceux-ci. En outre, les messages à échanger sont pris en compte. WSCL a été pensé pour s'employer conjointement avec WSDL.

Les définitions WSDL peuvent être manipulées par WSCL pour décrire les opérations possibles ainsi que leur chorégraphie. En retour, WSDL fournit les concrétisations vers des définitions de messages et des détails techniques pour les éléments manipulés par WSDL.

XIV- Sécurité des web services

XIV.1- vulnérabilités des web services

Les principales vulnérabilités rencontrées :

1 Pénétration du réseau privé par envoi de code malveillant

- Logiciels mis en frontal sur internet sans avoir été prévu.

2 Dénis de services :

➤ Risques :

- perte d'activité, perte de connectivité

➤ Causes (fréquentes) :

- un réseau de machines infestées se connecte à un serveur et sature ses ressources (réseau, processeur, mémoire),
- attaque par paquets IP
- attaque d'une application, forçant une boucle infinie.

➤ Remèdes :

- algorithmique : il faut pouvoir filtrer efficacement des listes d'IP contenant des millions d'adresses, et de manière "intelligente"

➤ théorique : revoir la conception des protocoles Internet.

3 Buffer Overflow

➤ Risques :

- prise de contrôle du serveur
- DoS

➤ Cause : pas de vérification ni statique ni dynamique du traitement des chaînes

➤ Remèdes :

- complexe : vérifier statiquement la correction du traitement des chaînes [Dor01][SM02]
- simple : utiliser un langage qui alloue dynamiquement la mémoire et vérifie dynamiquement les indices des tableaux (OCaml).

4 SQL Injection

➤ Risques

- vol de données : récupération de l'ensemble de la base
 - accès non autorisé (p.ex. login sans mot de passe)
 - Cause :
 - erreur de parsing des requêtes SQL écrites dans les programmes sous forme de chaînes de caractères et interprétées dynamiquement
- Exemple :
- ```
SELECT ... WHERE login='...' AND passwd='...'login : login' --
```
- Remèdes :
    - vérifier dynamiquement les requêtes SQL
    - garantir l'échappement des entrées utilisateurs
    - compiler les requêtes et les procédures stockées
    - ne pas utiliser SQL

## 5 CSS(Cross Site Scripting)

- Risque :
  - vol d'information des clients exemple : 14 avril 2008, tous les services Google vulnérables !  
<http://xs-sniper.com/blog/2008/04/14/google-xss/>
- Cause : une entrée utilisateur est insuffisamment vérifiée par le serveur Web ; elle permet d'afficher un code arbitraire
- Remèdes :
  - des fonctions pour vérifier systématiquement les entrées utilisateurs
  - vérifier que ces fonctions sont bien appelées (relecture du code, analyse statique )

## XIV.2- Quelques protocoles de Sécurisation des web services :

- ✓ Authentification des utilisateurs ou des autres applications en utilisant différentes méthodes
  - le login et mot de passe (le moins sécurisé)
  - Kerberos : Est un protocole d'authentification réseau qui repose sur un mécanisme de clés secrètes et l'utilisation de tickets, et non de mots de passe en clair, évitant ainsi le risque d'interception frauduleuse des mots de passe des utilisateurs. Créé au Massachusetts Institute of Technology (MIT), il porte le nom grec de Cerbère, gardien des Enfers (Κέρβερος). Kerberos a d'abord été mis en œuvre sur des systèmes Unix.[09]

- Certificat X.509 :X.509 a été créé en 1988 dans le cadre de la norme X.500. Est une norme de cryptographie de l'Union internationale des télécommunications pour les infrastructures à clés publiques (PKI). [09]
  - SAML(Security Assertion MarkupLanguage) : Basé sur le formalisme XML, permet de définir les informations d'identification et d'autorisation. Sa spécification prévoit que les informations d'authentification puissent se propager de service Web en service Web, en utilisant n'importe quel protocole de transport. [09]
  - Chaque document SAML comporte un élément XML "Request". Les éléments-fils obligatoires sont "AuthenticationQuery" et "Subject".
  - Lorsque rien de tout cela est disponible, au minimum il faut utiliser un filtrage sur les adresses IP source des serveurs clients.
- ✓ Contrôle d'accès aux fonctionnalités des web service par exemple la gestion des permissions d'accès des utilisateurs en fonction de leur profil.

### **XV- Conclusion**

Dans ce chapitre nous nous sommes intéressés aux technologies sur lesquelles sont basés les web services. Nous avons analysé le moyen de transmission et de transaction qu'est SOAP. Ce dernier permet l'échange d'informations structurées grâce au format XML. D'autre part, nous avons vu un langage de description de services, le WSDL basé aussi sur XML, qui permet de procurer aux applications clientes de nombreux renseignements sur le contenu d'un service et sur les façons de l'invoquer. Puis, nous avons abordé le registre annuaire pour les web services, nommé UDDI et considéré comme un standard. Enfin, nous avons illustré quelques moyens fondamentaux qui participent au développement et à la sécurisation des web services.

# Chapitre II

## Etat de l'art sur le J2EE

## **I. Introduction :**

De nos jours, le développement des applications ne s'arrête pas qu'à leur réalisation, mais aussi à leur qualité de dépendances vis-à-vis des nouvelles technologies. L'une des ces technologies révolutionnaires est l'interconnexion des ordinateurs sur internet ou même en réseau local. Avec cette interconnexion, il devient possible de faire fonctionner des applications sur des machines distantes pour répondre aux problématiques suivantes :

- Les données peuvent être présentes uniquement sur le serveur distant.
- Le serveur distant peut disposer d'une puissance de calcul ou de capacités de stockage dont l'utilisateur local ne dispose pas.
- L'application distante peut être utilisée simultanément par un grand nombre d'utilisateurs et sa mise à jour n'intervient qu'à un seul endroit.

La problématique revient donc à trouver une architecture logicielle qui nous permettra de réaliser nos applications distribuées.

Dans ce chapitre nous allons présenter l'architectures J2EE ainsi que les notions de base nécessaires à sa compréhension.

## **II. Architecture Logiciel :**

### **II.1 Définition1 :**

Une architecture est une infrastructure composée de modules, d'un mécanisme d'interaction entre ces modules et d'un ensemble de règles qui gouvernent cette interaction

### **II.2 Définition2 : [16]**

L'architecture logicielle décrit d'une manière symbolique et schématique les différents éléments d'un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions. Contrairement aux spécifications produites par l'[analyse fonctionnelle](#), le modèle d'architecture, produit lors de la phase de conception, ne décrit pas ce que doit réaliser un système informatique mais plutôt comment il doit être conçu de manière à répondre aux spécifications. L'analyse décrit le « quoi faire » alors que l'architecture décrit le « comment le faire ».

### III. Critères de qualité d'un logiciel :[17]

**1.L'interopérabilité extrinsèque** : exprime la capacité du logiciel à communiquer et à utiliser les ressources d'autres logiciels comme, par exemple, les documents créés par une certaine application.

**2.L'interopérabilité intrinsèque** : exprime le degré de cohérence entre le fonctionnement des commandes et des modules à l'intérieur d'un système ou d'un logiciel.

**3.La portabilité** : exprime la possibilité de compiler le code source et/ou d'exécuter le logiciel sur des plates-formes (machines, systèmes d'exploitation, environnements) différents.

**4.La compatibilité** : exprime la possibilité, pour un logiciel, de fonctionner correctement dans un environnement ancien (compatibilité descendante) ou plus récent (compatibilité ascendante).

**5.La validité** : exprime la conformité des fonctionnalités du logiciel avec celles décrites dans le cahier des charges.

**6.La vérifiabilité** : exprime la simplicité de vérification de la validité.

**7.L'intégrité** : exprime la faculté du logiciel à protéger ses fonctions et ses données d'accès non autorisés.

**8.La fiabilité** : exprime la faculté du logiciel à gérer correctement ses propres erreurs de fonctionnement en cours d'exécution.

**9.La maintenabilité** : exprime la simplicité de correction et de modification du logiciel, et même, parfois, la possibilité de modification du logiciel en cours d'exécution.

**10.La réutilisabilité** : exprime la capacité de concevoir le logiciel avec des [composants](#) déjà conçus tout en permettant la réutilisation simple de ses propres composants pour le développement d'autres logiciels.

**11.L'extensibilité** : exprime la possibilité d'étendre simplement les fonctionnalités d'un logiciel sans compromettre son intégrité et sa fiabilité.

**12.L'efficacité** : exprime la capacité du logiciel à exploiter au mieux les ressources offertes par la ou les machines où le logiciel sera implanté.

**13.L'autonomie** : exprime la capacité de contrôle de son exécution, de ses données et de ses communications.

**14.La transparence** : exprime la capacité pour un logiciel de masquer à l'utilisateur (humain ou machine) les détails inutiles à l'utilisation de ses fonctionnalités.

**15.La composabilité** : exprime la capacité pour un logiciel de combiner des informations provenant de sources différentes.

**16.La convivialité** : décrit la facilité d'apprentissage et d'utilisation du logiciel par les usagers.

#### **IV. Présentation de J2EE :**

J2EE est une plate-forme fortement orientée serveur proposé par SUN pour le développement et l'exécution d'applications distribuées multi-niveaux. Elle s'appuie entièrement sur le langage JAVA. Composée de deux parties essentielles :

**Partie1** : c'est un ensemble de spécifications pour l'infrastructure dans laquelle s'exécutent les composants écrits en Java : un tel environnement se nomme serveur d'application.

**Partie2** : est un ensemble d'API qui peut être obtenu et utilisé séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers.

L'utilisation de J2EE pour développer et exécuter une application propose plusieurs avantages :

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

J2EE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Ce choix dépend des besoins auxquels doit répondre l'application.

L'architecture d'une application se découpe idéalement en au moins trois tiers :

- ✓ la partie cliente : c'est la partie qui permet le dialogue avec l'utilisateur. Elle peut être composée d'une application standalone, d'une application web ou d'applets

**NB** : standalone est un qualificatif qui indique qu'un produit peut être utilisé seul

- ✓ la partie métier : c'est la partie qui encapsule les traitements (dans des EJB ou des JavaBeans)
- ✓ la partie données : c'est la partie qui stocke les données

## V. Topologie(s) d'une application J2EE :

### V.1 Application multi tiers :

Son Architecture logicielle sépare les fonctions en plusieurs étages (strates) de traitements. Il existe trois modèles de tier :

- ✓ modèle 2-tier : client/serveur : 2 niveaux = client et serveur.
- ✓ modèle 3-tier : client (utilisateur), serveur d'applications, serveur de données
- ✓ modèles n-tier avec  $n > 3$  : client (utilisateur), divers niveaux de liaison et de traitement de règles (serveur Web, serveurs d'applications, brokers d'objets distants, moniteur transactionnel, web service), serveur de données

### V.2 Application web :

En informatique, une application Web (aussi appelée *WebApplicatoin*) est un logiciel applicatif, elle est placée sur un serveur et se manipule en actionnant des widgets à l'aide d'une interface web, *via* un réseau informatique (Internet, intranet, réseau local).

Exemple d'application web :

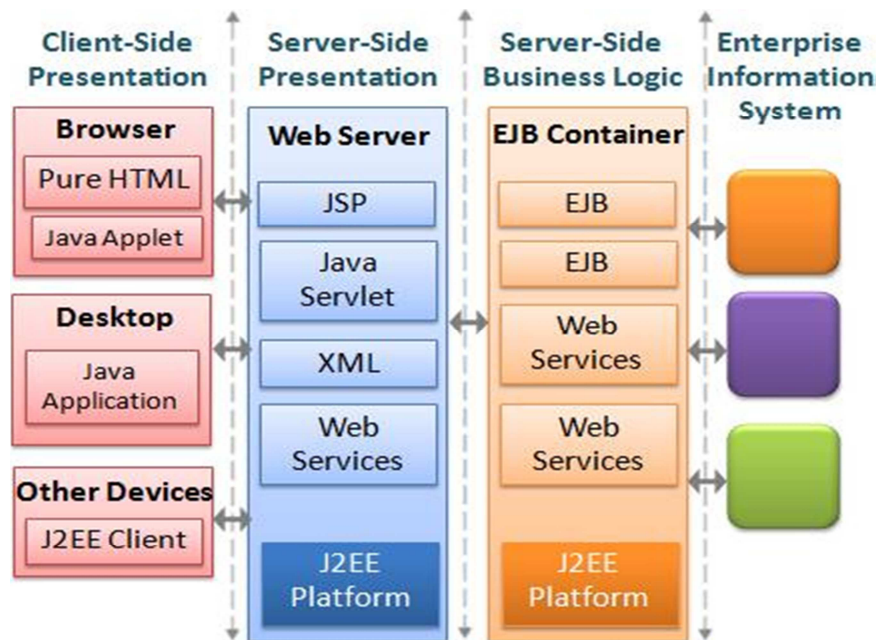
- ✓ Les messageries web, les wikis, les blogs sont des applications Web.
- ✓ Les moteurs de recherches, les logiciels de commerce électronique, les jeux en ligne, les logiciels de forum peuvent être sous forme d'application Web.
- ✓ Des appareils réseau tels par exemple les routeurs sont parfois équipés d'une application Web dans leur

## VI. Architecture J2EE :[17]

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

- **Technologie Web Application:** technologies liées à la production des interfaces web dynamiques, par exemple JSP (Java Servlet Pages) et servlet.
- **Technologie Enterprise Application:** technologies plus directement liées à la logique de business : EJB (Enterprise Java Bean), JavaMail, JMS (Java Message Service), JTA (Java Transaction), etc.
- **Technologie Web Services:** technologies utiles au développement des applications adhérentes au paradigme SOA (Service Oriented Architecture) : web services, JAX-WS (java API for XML-based web services), JAX-RPC (Java API for XML-Based RPC)

- **Technologie Management and Security:** technologies liées à la gestion de la technologie entreprise afin de réaliser l'accès et l'échange d'information entre machines et services distribués : JAAS (Java Authentication and Authorization Service), JCA (Java Connector Architecture)



**Figure N°II.1 : Architecture et les composants de J2EE.**

## VII. Composants de l'architecture J2EE

- ✓ Conteneur de composants
- ✓ Clients J2EE
- ✓ Composants web
- ✓ Composants Enterprise JavaBeans [EJB]:
- ✓ Serveur EJB

### VII.1. Conteneur de composants J2EE

#### VII.1.1 Les conteneurs

Les conteneurs assurent la gestion du cycle de vie des composants qui s'exécutent en eux. Ils fournissent des services qui peuvent être utilisés par les applications lors de leur exécution.

Pour cela, il existe plusieurs conteneurs définis par J2EE:

- ✓ conteneur web : pour exécuter les servlets et les JSP
- ✓ conteneur d'EJB : pour exécuter les EJB

- ✓ conteneur client : pour exécuter des applications standalone sur les postes qui utilisent des composants J2EE

Les serveurs d'applications peuvent fournir un conteneur web uniquement (exemple : Tomcat) ou un conteneur d'EJB uniquement (exemple : JBoss, Jonas, ...) ou les deux (exemple : Websphere, Weblogic, ...).

### **VII.1.2 Le conteneur web : [MAS 2011]**

Le conteneur web est une implémentation des spécifications servlets et par extension des spécifications des JSP. Ce type de conteneur est composé de deux éléments majeurs : un moteur de servlets (servlets engine) et un moteur de JSP (JSP engine).

Les conteneurs web peuvent généralement utiliser leur propre serveur web et être utilisés en tant que plug in d'un serveur web dédié (Apache, IIS, ...).

L'implémentation de référence pour ce type de conteneur est le projet open source Tomcat du groupe Apache.

### **VII.1.3 Le conteneur d'EJB : [MAS 2011]**

La spécification EJB définit un conteneur comme étant l'environnement dans lequel un ou plusieurs composants EJB s'exécutent. Le conteneur fournit l'infrastructure requise pour exécuter les composants distribués, permettant ainsi aux développeurs de clients et de composants de se concentrer sur la programmation de la logique de gestion et non sur le code niveau système. Sous EJB Server, le conteneur encapsule :

- ✓ la version d'exécution cliente et les classes de stubs générées ce qui permet aux clients d'exécuter des composants sur un serveur distant comme s'il s'agissait d'objets locaux.
- ✓ le service de nommage ce qui permet aux clients d'instancier les composants par leur nom et aux composants d'obtenir des ressources telles que les connexions aux bases de données par nom.
- ✓ le distributeur de composants EJB Server qui exécute la classe d'implémentation des composants et fournit des services tels que la gestion des transactions, le pool de connexions aux bases de données et la gestion du cycle de vie des instances.

## VII. 2. Clients J2EE

Il s'agit de composants spécifiques ou composants lourds chargés des traitements propres à un secteur d'activité (on parle de *logique métier* ou de *logique applicative*).

### VII .3. Composants web :

#### VII .3.1. Pages JSPs :[18]

Les JSP (Java Server Pages) sont une technologie Java qui permettent la génération de pages web dynamiques. Permettent d'introduire du code Java dans des tags prédéfinis à l'intérieur d'une page HTML. La technologie JSP mélange la puissance de Java côté serveur et la facilité de mise en page d'HTML côté client.

**VII .3.2. Tags :** Il existe trois types de tags :

- **tags de directives :** ils permettent de contrôler la structure de la servlet générée et de préciser des informations globales sur la page JSP. Les spécifications des JSP définissent trois directives :

- ✓ **page :** permet de définir des options de configuration
- ✓ **include :** permet d'inclure des fichiers statiques dans la JSP avant la génération de la servlet
- ✓ **taglib :** permet de définir des tags personnalisés Leur syntaxe est la suivante :

`<% @ directive attribut="valeur" ... %>`

- **tags de scripting:**

Ces tags permettent d'insérer du code Java qui sera inclus dans la servlet générée à partir de la JSP. Il existe trois tags pour insérer du code Java :

- ✓ **le tag de déclaration :** le code Java est inclus dans le corps de la servlet générée. Ce code peut être la déclaration de variables d'instances ou de classes ou la déclaration de méthodes. Sa syntaxe est `<%! declarations %>`
- ✓ **le tag d'expression :** évalue une expression et insère le résultat sous forme de chaîne de caractères dans la page web générée. Sa syntaxe est `<%= expression %>`
- ✓ **le tag de scriptlets :** par défaut, le code Java est inclus dans la méthode `service()` de la servlet. Sa syntaxe est la suivante : `<% code Java %>`
- **tags d'actions:** permettent de réaliser des traitements couramment utilisés.

## VII .4. Composants Enterprise JavaBeans [EJB]: [19]

L'architecture Enterprise JavaBeans est une technologie côté serveur pour développer et déployer des composants contenant la logique métier d'une application d'entreprise. Il y a 3 types d'enterprise beans : les beans de sessions, d'entité et de messages.

### VII.4.1. Beans de session (session bean)

Un EJB session est un EJB de service dont la durée de vie correspond à un échange avec un client. Ils contiennent les règles métiers de l'application. Il existe deux types d'EJB session

- ✓ sans état (stateless) : sont capables de conserver l'état du bean dans des variables d'instance durant toute

la conversation avec un client. Mais ces données ne sont pas persistantes : à la fin de l'échange avec le client, l'instance de l'EJB est détruite et les données sont perdues.

- ✓ avec état (stateful) : ne peuvent pas conserver de telles données entre chaque appel du client, ce qui signifie qu'il fournit un accès à des méthodes implémentant la logique applicative

### VII .4.2. Les EJB entité

Ces EJB permettent de représenter et de gérer des données enregistrées dans une base de données. Ils implémentent l'interface EntityBean.

L'avantage d'utiliser un tel type d'EJB plutôt que d'utiliser JDBC ou de développer sa propre solution pour mapper les données est que certains services sont pris en charge par le conteneur.

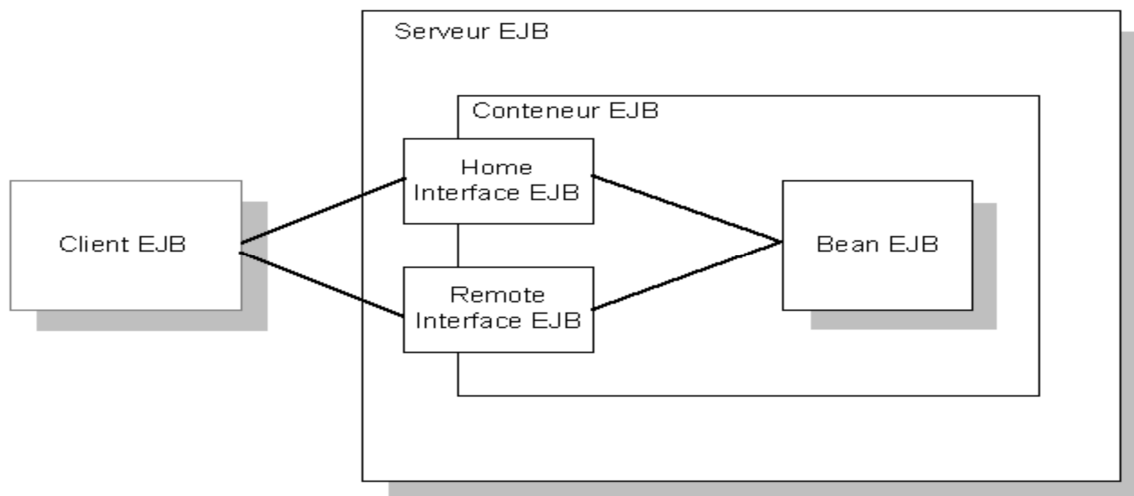
Les beans entités assurent la persistance des données en représentant tout au partie d'une table ou d'une vue. Il existe deux types de bean entité :

- ✓ persistance gérée par le conteneur (CMP : Container Managed Persistence)
- ✓ persistance gérée par le bean (BMP : Bean Managed Persistence).

### VII.4.3. Bean de message (Message-Driven Beans) : [20]

Les EJB message permettant de déclencher un processus côté serveur applicatif lors de la publication d'un message asynchrone. Pour ces composants, le client ne s'adresse pas directement aux composants mais publie un message sur un réceptacle [JMS](#) (queue ou topic) configuré sur le serveur applicatif qui va alors déclencher l'activation par ce serveur d'une instance de l'EJB concerné pour pouvoir traiter ce message.

JMS : est un API permet d'envoyer et de recevoir des messages de manière [asynchrone](#) entre applications ou composants [Java](#)



**Figure N° II.2 : l'architecture générale d'un EJB.**

## VII.5. Serveur EJB [MAS 2011]

EJB Server intègre le conteneur EJB assurant les services requis par le composant de ce dernier.

### ➤ **Client EJB**

Le client EJB fournit généralement la logique de l'interface utilisateur sur la machine cliente. Il passe les appels aux composants EJB distants hébergés sur un serveur et doit savoir comment trouver le serveur EJB et interagir avec les composants EJB. Un composant EJB peut faire office de client EJB en appelant les méthodes d'un autre composant EJB.

Un client EJB ne communique pas directement avec un composant EJB.

Le conteneur fournit les objets proxy mettant en œuvre les Home Interface et Remote Interface des composants. Le Remote Interface des composants définit les méthodes de gestion pouvant être appelées par le client. Ce dernier appelle les méthodes de l'Home Interface pour créer et détruire les proxys pour le Remote Interface.

### ➤ **L'interface remote**

L'interface remote permet de définir les méthodes qui contiendront les traitements proposés par le bean. Cette interface doit étendre l'interface `javax.ejb.EJBObject`.

```
package com.jmdoudoux.ejb;

import java.rmi.RemoteException;
import javax.ejb.EJBObject;

public interface MonPremierEJB extends EJBObject {
 public String message() throws RemoteException;
}
```

➤ **L'interface home**

L'interface home permet de définir des méthodes qui vont gérer le cycle de vie du bean. Cette interface doit étendre l'interface EJBHome.

```
package com.jmdoudoux.ejb;

import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EJBHome;

public interface MonPremierEJBHome extends EJBHome {
 public MonPremierEJB create() throws CreateException, RemoteException;
}
```

**VIII. Les services proposés par la plate-forme J2EE : [ING 2010]**

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

Ces services sont classés par catégories : [ING 2010]

### **VIII.1 Les services d'infrastructures:**

- **JTA/JTS** (*Java Transaction API/Java Transaction Services*) est un API définissant des interfaces standard avec un gestionnaire de transactions.
- **JCA** (*J2EE Connector Architecture*) est une API de connexion au système d'information de l'entreprise, tels que les ERP.
- **JMX** (*Java Management Extension*) fournit des extensions permettant de développer des applications web de supervision d'applications.
- **JNDI** (Java Naming and Directory Interface )

Cette API fournit une interface unique pour utiliser différents services de nommage ou d'annuaires et définit une API standard pour permettre l'accès à ces services.

Un service de nommage permet d'associer un nom unique à un objet et de faciliter ainsi l'obtention de cet objet. Il existe plusieurs types de service de nommage parmi lesquels :

- ✓ DNS (Domain Name System) : service de nommage utilisé sur internet pour permettre la correspondance entre un nom de domaine et une adresse IP
- ✓ LDAP(Lightweight Directory Access Protocol) : annuaire
- ✓ NIS (Network Information System) : service de nommage réseau développé par Sun Microsystems
- ✓ COS Naming (Common Object Services) : service de nommage utilisé par Corba pour stocker et obtenir des références sur des objets Corba□□
- **JDBC** (Java DataBase Connectivity) :

Désigne aussi une API définie par Sun pour permettre un accès aux bases de données avec Java.

### **VIII.2 :Les services de communication :**

- **JAAS** (Java Authentication and Authorization Service) est une API de gestion de l'authentification et des droits d'accès.
- **JavaMail** est une API permettant l'envoi de courrier électronique.

- **JMS** (Java Message Service) fournit des fonctionnalités de communication asynchrone (appelées *MOM* pour *Middleware Object Message*) entre applications.

L'architecture J2EE permet ainsi de séparer la couche présentation, correspondant à l'interface homme-machine (IHM), la couche métier contenant l'essentiel des traitements de données en se basant dans la mesure du possible sur des API existantes, et enfin la couche de données correspondant aux informations de l'entreprise stockées dans des fichiers, dans des bases de données relationnelles ou XML, dans des annuaires d'entreprise ou encore dans des systèmes d'information complexes.

- **RMI** (Remote Method Interface) : est une technologie développée et fournie par Sun à partir du JDK 1.1 pour permettre de mettre en oeuvre facilement des objets distribués.

Le but de RMI est de permettre l'appel, l'exécution et le renvoi du résultat d'une méthode exécutée dans une machine virtuelle différente de celle de l'objet l'appelant. Cette machine virtuelle peut être sur une machine différente pourvu qu'elle soit accessible par le réseau. La machine sur laquelle s'exécute la méthode distante est appelée serveur. Voici un schéma montrons le fonctionnement de RMI.

### **Les amorces (Stub/Skeleton)**

Elles assurent le rôle d'adaptateurs pour le transport des appels distants.

- ✓ Elles réalisent les appels sur la couche réseau.
- ✓ Elles réalisent l'assemblage et le désassemblage des paramètres (*marshalling*, *unmarshalling*).
- ✓ Une référence d'objets distribués correspond à une référence d'amorce.
- ✓ Les amorces sont créées par le générateur **rmic**.

### **Les Stubs**

- ✓ Représentants locaux de l'objet distribué .
- ✓ Initient une connexion avec la JVM distante en transmettant l'invocation distante à la couche des références d'objets.
- ✓ Assemblent les paramètres pour leur transfert à la JVM distante.
- ✓ Attendent les résultats de l'invocation distante.
- ✓ Désassemblent la valeur ou l'exception renvoyée.
- ✓ Renvoient la valeur à l'appelant.
- ✓ S'appuient sur la sérialisation.

### **Les squelettes**

- ✓ Désassemblent les paramètres pour la méthode distante.
- ✓ Font appel à la méthode demandée.
- ✓ Assemblage du résultat (valeur renvoyée ou exception) à destination de l'appelant.

#### **La couche des références d'objets Remote Reference Layer**

- ✓ Permet d'obtenir une référence d'objet distribué à partir de la référence locale au stub.
- ✓ Cette fonction est assurée grâce à un service de noms **rmiregister** (qui possède une table de hachage dont les clés sont des noms et les valeurs sont des objets distants).
- ✓ Un unique **rmiregister** par JVM.
- ✓ **rmiregister** s'exécute sur chaque machine hébergeant des objets distants.
- ✓ **rmiregister** accepte des demandes de service sur le port 1099.

#### **La couche transport**

- ✓ réalise les connexions réseau basées sur les flux entre les JVM
- ✓ emploie un protocole de communication propriétaire (**JRMP**: Java Remote Method Invocation) basé sur TCP/IP
- ✓ Le protocole JRMP a été modifié afin de supprimer la nécessité des squelettes car depuis la version 1.2 de Java, une même classe skeleton générique est partagée par tous les objets distants.

### **IX. Les différentes étapes pour créer un objet distant et l'appeler avec RMI :**

Le développement coté serveur se compose de :

- ✓ La définition d'une interface qui contient les méthodes qui peuvent être appelées à distance.
- ✓ L'écriture d'une classe qui implémente cette interface.
- ✓ L'écriture d'une classe quiinstanciera l'objet et l'enregistrera en lui affectant un nom dans le registre de nom RMI (RMI Registry).

Le développement côté client se compose de :

- ✓ L'obtention d'une référence sur l'objet distant à partir de son nom.
- ✓ L'appel à la méthode à partir de cette référence.

Enfin, il faut générer les classes stub et skeleton en exécutant le programme `rmic` avec le fichier source de l'objet distant.

### **X. L'environnement d'exécution des applications J2EE :**

J2EE propose des spécifications pour une infrastructure dans laquelle s'exécutent ses composants. Ces spécifications décrivent les rôles de chaque élément et précisent un ensemble d'interfaces pour permettre à chacun de ces éléments de communiquer.

Ceci permet de séparer les applications et l'environnement dans lequel elles s'exécutent. Les spécifications précisent à l'aide des API un certain nombre de fonctionnalités que doit implémenter l'environnement d'exécution.

Ces fonctionnalités sont de bas niveau ce qui permet aux développeurs de se concentrer sur la logique métier. Pour exécuter ces composants de natures différentes, J2EE définit des conteneurs pour chacun de ces composants. Il définit pour chaque composant des interfaces qui leur permettront de dialoguer avec les composants lors de leurs exécutions.

Les conteneurs permettent aux applications d'accéder aux ressources et aux services en utilisant les API.

Les appels aux composants se font par des clients via les conteneurs. Les clients n'accèdent pas directement aux composants mais sollicitent le conteneur pour les utiliser.

### **XI. L'assemblage et le déploiement d'applications J2EE :**

J2EE propose une spécification pour décrire le mode d'assemblage et de déploiement d'une application J2EE. Une application J2EE peut regrouper différents modules : modules web, modules EJB ... Chacun de ces modules possède son propre mode de packaging. J2EE propose de regrouper ces différents modules dans un module unique sous la forme d'un fichier EAR (Entreprise ARchive).

Le format de cette archive est très semblable à celui des autres archives :

- ✓ un contenu : les différents modules qui composent l'application (module web, EJB, fichier RAR, ...)
- ✓ un fichier descripteur de déploiement

Les serveurs d'application extraient chaque module du fichier EAR et les déploient séparément un par un.

### **XII. Conclusion :**

Dans ce chapitre, nous avons abordé les notions de base sur l'architecture J2EE.

Ces notions seront utilisées pour le développement de notre application. Dans le chapitre suivant, nous aborderons la conception de notre application en utilisant le langage de modélisation UML.

# Chapitre III

## Analyse et Conception

## **I. Introduction :**

Dans le but d'une meilleure organisation et avant toute réalisation d'une application informatique, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou démarche bien définie, en mettant en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité. Dans ce chapitre nous allons entamer le développement de notre application par une analyse qui mettra en évidence les différents acteurs intervenants dans le système ainsi que leurs besoins.

La phase conception, s'appuyant sur les résultats de cette analyse, donnera la modélisation des objectifs à atteindre.[Lai 04]

De nombreux efforts ont été consentis dans le but d'avoir une meilleure analyse et de rendre la conception plus complète. L'approche objet, s'est avérée un modèle d'analyse et de conception très puissant et se trouve de plus en plus utilisée. Pour cela nous avons adopté la conception avec l'UML (Unified Modelling language) qui permet de bien représenter la dynamique d'une application par la série des diagrammes qu'il offre.

## **II. Organisation de NAFTAL :**

L'organisation de cette entreprise est définie par la décision n°17/CAB du 30 mars 1986, complétée par la décision n°90\*/CAB du 05 novembre 1986.

Dans le cadre du plan national de développement économique et social, NAFTAL est chargé de la commercialisation et de distribution des produits pétroliers, le GPL, les combustibles et les pneumatiques.

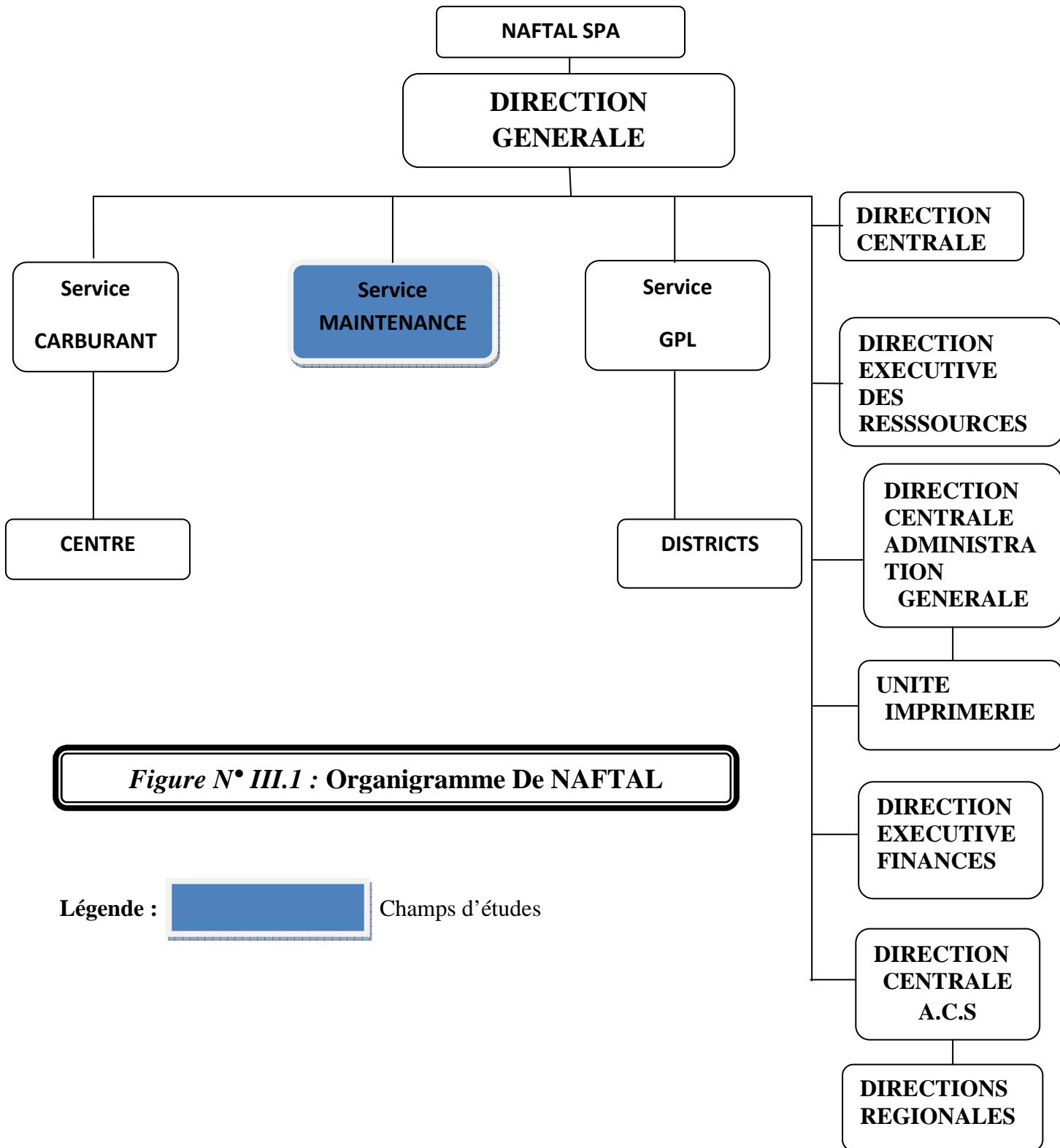
L'identité de l'entreprise NAFTAL est comme suit :

Nom : Entreprise Nationale de Commercialisation Distribution des produits Pétroliers.

Nature : SPA. (Société Par Action) au capital de 15.650.00 DA chacune, qui sont entièrement détenues par SONATRACH.

Siège : rue des Dunes BP 73, CHERAGA, ALGER

### III. Organigramme général de NAFTAL

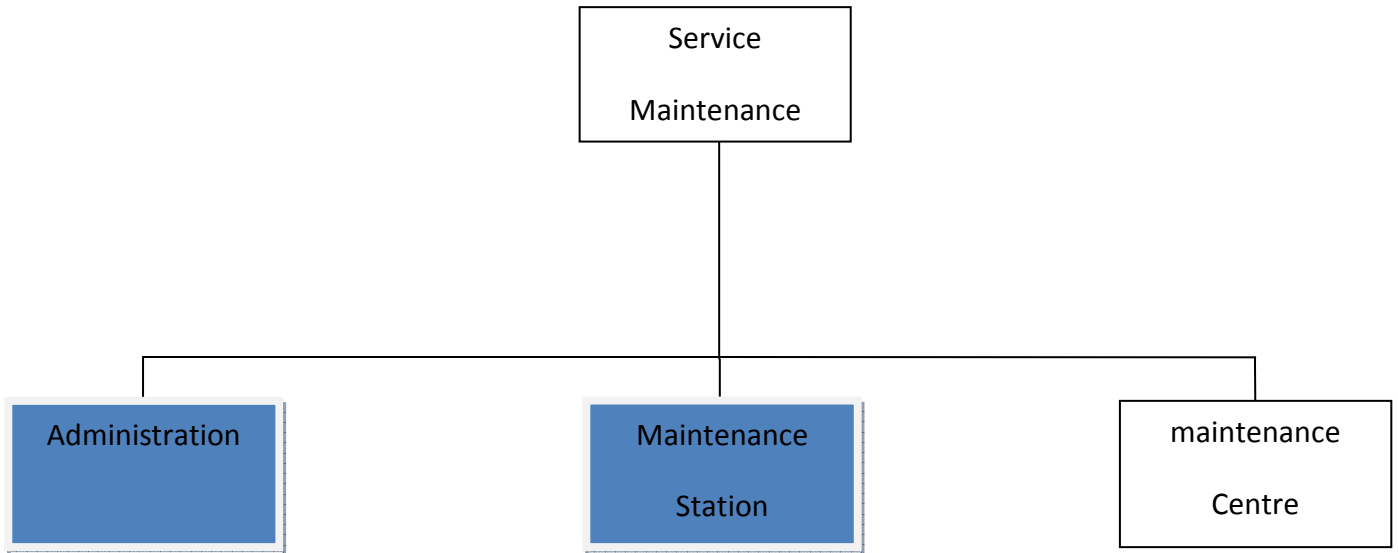


**Figure N° III.1 : Organigramme De NAFTAL**

Légende :   Champs d'études

**Présentation du champ d'étude :** notre champ d'étude est la maintenance des stations ainsi que le l' administration dériver du service maintenance. Comme l'indique l'organigramme suivant

**IV. Organigramme du champ d'étude :**



**Figure N° III.2 : Organigramme du champ d'étude**

**V. Problématique de notre champ d'études:**

Le service Maintenance est confronté à plusieurs problèmes dans la gestion de ses tâches, tels que :

- Retard dans la réception des demandes de travaux (DT) envoyées par les stations.
- Difficultés de suivi de ces dernières une fois les avoir reçus.
- Mauvaise circulation des documents entre les services.
- Perte de temps dans des interventions où la panne peut être réparée sans l'aide du service technique de Naftal.
- Perte et entassement des piles de documents à cause de la non-informatisation de ceux-là.
- Difficulté de gestion des documents établit (compte rendu (CR), ordre de travaux (OT) et contrôle débit (CDB)).
- Retard dans la réalisation des tâches telle que l'élaboration d'un ordre de travaux.

## **VI. Analyse :**

### **VI.1 Spécification des besoins :**

C'est une étape primordiale au début de chaque démarche de développement. Son but est de veiller à développer une application adéquat, sa finalité est la description générale des fonctionnalités du système, en répondant à la question : Que doit faire notre système?

Notre système doit répondre aux exigences suivantes :

1. Le système doit pouvoir éditer :
  - les demandes travaux(DT) : document établi par une station.
  - les ordres de travaux(OT) : document établi après réception d'une DT, pour ordonner l'exécution des travaux.
  - les contrôles débit(CDB) : rapport de vérification du débit des distributeurs de carburant.
  - Les comptes rendu(CR) : rapport des travaux effectués.
2. consulter les DT, OT, CDB.
3. Enregistrer les informations à propos des interventions, des demandes et toutes autres entités, pour mettre à jour la base de données.
4. La suppression et la modification de certaines entités par exemple l'historique de toutes les pannes répertoriés.
5. Gérer la base de données.

Dans l'étape d'analyse, les besoins du client doivent être recensés, formalisés ainsi que l'ensemble des contraintes, puis l'estimation de la faisabilité de ces besoins.

### **VI.2 Identification des acteurs :**

Notre système est composé de deux applications disjointes communiquant entre elles pour former notre système qui interagit avec différents acteurs afin de réaliser un ensemble d'opérations. Un acteur représente un rôle que peut jouer l'utilisateur dans le système. L'acteur est associé à un cas d'utilisation (voir la section suivante), c'est-à-dire qu'il peut interagir avec lui et participer à son scénario, il est représenté par un personnage stylisé.

Les acteurs de notre système sont :

- Le chef de service maintenance et de son secrétaire qu'on regroupe sous le nom utilisateur "**Service Maintenance**".
- Les employés du service personnel que nous identifions sous le nom utilisateur "**Service Personnel**".

- Les chefs de stations et les employés agréés à utiliser notre application (les chefs d'équipes, les comptables) qu'on regroupe sous le nom utilisateur "**Personnel Station**".

### VI.3 Identification des cas d'utilisations :

Un cas d'utilisation (use case) modélise une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système.

Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur.

Pour notre cas, nous avons pu dégager les cas d'utilisation suivants :

| Acteurs                    | Cas d'utilisations                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Service Maintenance</b> | <ul style="list-style-type: none"> <li>- S'authentifier.</li> <li>- Gestion des stations                             <ul style="list-style-type: none"> <li>○ Ajouter une station</li> <li>○ Supprimer une station</li> </ul> </li> <li>- Consultation de la liste des stations</li> <li>- Gestion des pannes                             <ul style="list-style-type: none"> <li>○ Ajouter une panne</li> <li>○ Modifier une panne</li> <li>○ Supprimer une panne</li> </ul> </li> <li>- Consultation des pannes                             <ul style="list-style-type: none"> <li>○ Recherche d'une panne</li> </ul> </li> <li>- Consultation des remarques pannes</li> <li>- Suppression des remarques pannes</li> <li>- Gestion des ordres de travaux                             <ul style="list-style-type: none"> <li>○ Ajouter un nouvel ordre de travail</li> </ul> </li> <li>- Consultation des ordres de travaux</li> <li>- Gestion des contrôles débit                             <ul style="list-style-type: none"> <li>○ Etablissement d'un nouveau contrôle débit</li> </ul> </li> <li>- Consultation des demandes de travaux</li> <li>- Consultation des contrôles débit</li> <li>- Consultation des comptes rendu</li> <li>- Gestion du compte                             <ul style="list-style-type: none"> <li>○ Modifier mot de passe</li> <li>○ Modifier le rappel du mot de passe</li> </ul> </li> <li>- Consultation du rappel du mot de passe.</li> <li>- Impression des documents (OT, CDB, CR)</li> </ul> |
| <b>Personnel Station</b>   | <ul style="list-style-type: none"> <li>- S'authentifier.</li> <li>- Etablissement d'une DT</li> <li>- Consultation des pannes                             <ul style="list-style-type: none"> <li>○ Recherche d'une panne</li> </ul> </li> <li>- Ajouter des remarques pannes</li> <li>- Gestion du compte                             <ul style="list-style-type: none"> <li>○ Modifier mot de passe</li> <li>○ Modifier le rappel du mot de passe</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | - Consultation du rappel du mot de passe.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Service Personnel</b> | - S'authentifier.<br>- Gestion du personnel <ul style="list-style-type: none"><li>○ Ajouter un employé</li><li>○ Modifier un employé</li><li>○ Supprimer un employé</li></ul> - Consultation de la liste des employés <ul style="list-style-type: none"><li>○ Recherche d'un employé</li></ul> - Gestion du compte <ul style="list-style-type: none"><li>○ Modifier mot de passe</li><li>○ Modifier le rappel du mot de passe</li></ul> - Consultation du rappel du mot de passe. |

#### **VI.4 Architecture générale du système :**

Pour avoir une idée globale du fonctionnement de notre système, nous l'avons décomposé en 3 modules :

- Module d'authentification.
- Module de travail.
- Module de résultats.

##### **1. Module d'authentification :**

Ce module permet à l'utilisateur de s'authentifier en donnant son nom d'utilisateur et son mot de passe et ainsi pouvoir accéder aux différentes fonctionnalités de notre système. Il peut aussi se faire rappeler son mot de passe par le système.

##### **2. Module de travail :**

Ce module regroupe les différentes fonctionnalités que propose notre système selon l'utilisateur auquel est destiné chaque application, et qu'on regroupe comme suit :

##### ➤ **Service Maintenance :**

- **Recherche d'une panne :** la recherche d'une panne s'effectue de trois manières : par numéro, par type ou par priorité.
- **Consultation des pannes :** l'utilisateur peut consulter l'historique de toutes les pannes et voir les détails de chacune d'elles.
- **Consultation des remarques pannes :** l'utilisateur peut consulter les remarques pannes qui sont des remarques envoyées par les stations pour commenter la solution ou la description (symptômes) d'une panne.

- **Gestion des pannes :** l'utilisateur a droit aux différentes tâches de gestion des pannes (ajout, modification, suppression) et la suppression des remarques à propos de ces dernières.
  - **Suppression des remarques pannes :** le système permet à l'utilisateur de supprimer les remarques pannes.
  - **Consultation des stations :** l'utilisateur peut consulter la liste de toutes les stations.
  - **Gestion des stations :** qui consiste en l'ajout et la suppression de stations.
  - **Consultation des demandes de travaux :** l'utilisateur a accès à la liste de toutes les demandes de travaux envoyées par les stations.
  - **Consultation des ordres de travaux :** l'utilisateur peut consulter la liste de tous les ordres de travaux.
  - **Etablissement de nouveaux ordres de travaux :** l'utilisateur peut ajouter un nouvel ordre de travaux.
  - **Consultation des comptes rendu :** l'affichage de la liste des comptes rendu établis.
  - **Ajout de compte rendu :** établissement de nouveaux comptes rendu.
  - **Consultation des contrôles débit :** l'utilisateur peut consulter la liste de tous les contrôles débit.
  - **Etablissement de nouveaux contrôles débit :** l'utilisateur peut ajouter de nouveaux contrôles débit.
  - **Impression des documents :** Notre application offre la possibilité d'imprimer des documents propres à l'utilisateur (DT, OT, comptes rendus, contrôles débit).
  - **Gestion du compte :** l'utilisateur peut gérer son compte en modifiant son mot de passe et son rappel du mot de passe..
- **Service Personnel :**
- **Recherche d'un employé :** la recherche d'une personne s'effectue de quatre manières différentes: par numéro, par nom, par fonction, ou le domaine.
  - **Consultation de la liste des employés :** l'utilisateur peut consulter la liste de tous les employés.

- **Gestion du personnel :** l'utilisateur a droit aux différentes tâches de gestion du personnel (ajout, modification, suppression).
  - **Gestion du compte :** l'utilisateur peut gérer son compte en modifiant son mot de passe et son rappel du mot de passe.
- **Personnel Station :**
- **Etablissement de nouvelles demandes de travaux :** l'utilisateur envoie (ajoute) une demande de travaux.
  - **Consultation des pannes :** l'utilisateur peut consulter l'historique de toutes les pannes et voir les détails de chacune d'elles.
  - **Recherche d'une panne :** la recherche d'une panne s'effectue de trois manières : par numéro, par type ou par priorité.
  - **Envoi d'une remarque panne :** l'utilisateur ajoute une remarque pour commenter la solution ou la description d'une panne.
  - **Impression des demandes de travaux :** notre application permet au personnel de la station de sauvegarder une copie des demandes de travaux envoyées.
  - **Gestion du compte :** l'utilisateur peut gérer son compte en modifiant son mot de passe et son rappel du mot de passe.

### **3. Module de résultats :**

Il représente les résultats du module précédent. Il englobe les résultats des recherches, les listes retournées, les mises à jour effectuées et les documents imprimés.

### **VI.5 : Spécification des scénarios :**

Un scénario est une requête de cas d'utilisation, tout résultat possible de l'exécution d'un cas d'utilisation est appelé scénario.

Les scénarios doivent être définis par les utilisateurs eux même, chacune des tâches effectuées par un ou plusieurs acteurs sera décrite par un ensemble des scénarios.

## [Titre du document]

➤ Voici quelques scénarios pour quelques cas d'utilisations :

| Taches                                                    | Scénarios par tache                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T01 : S'authentifier                                      | <p>S01 : Le système affiche un formulaire d'authentification.</p> <p>S02 : L'utilisateur saisit son nom utilisateur et son mot de passe, puis il click sur « connexion ».</p> <p>S03 : Le système vérifie la validité des informations reçues et affiche l'espace demandé, sinon il retourne un message d'erreur</p> |
| T02 : Ajouter une station, personne, ou une panne, une DT | <p>S04 : L'utilisateur demande le formulaire d'ajout</p> <p>S05 : Le système lui affiche le formulaire souhaité.</p> <p>S06: Le système vérifie les champs des informations entrées et enregistre dans la BDD, sinon affiche un message d'erreur.</p>                                                                |
| T03 : Consultation d'une DT.                              | <p>S07 : L'utilisateur demande à consulter la liste des DT.</p> <p>S08 : Le système affiche la liste des DT</p> <p>S09 :L'utilisateur choisit une DT pour afficher les détails.</p>                                                                                                                                  |
| T04 : Consultation d'une nouvelle DT.                     | <p>S10 : L'utilisateur accède à l'espace DT</p> <p>S11 : Le system affiche le nombre de nouvelle DT</p> <p>S12 : L'utilisateur demande à afficher la liste des nouvelles DT et il click sur l'une d'elles pour voir les détails</p>                                                                                  |
| T05 : Consultation des informations d'une personne.       | <p>S13 : L'utilisateur demande à consulter la liste des personnes.</p> <p>S14 : Le système retourne la liste de toutes les personnes</p> <p>S15 :L'utilisateur choisit une personne pour afficher les détails.</p>                                                                                                   |
| T06 : Consultation de la liste d'OT.                      | <p>S16 : L'utilisateur demande à consulter la liste des OT.</p> <p>S17 : Le système retourne la liste des OT</p> <p>S18 :L'utilisateur choisit un OT pour afficher les détails.</p>                                                                                                                                  |

## [Titre du document]

|                                                    |                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T07 : Ajouter un OT                                | <p>S19 : L'utilisateur consulte une DT et il click sur Etablir un OT ou bien il demande le formulaire d'ajout d'un OT Dans l'espace OT</p> <p>S20 : Le système lui affiche le formulaire souhaité.</p> <p>S21: Le système vérifie les champs des informations et les enregistre, sinon il affiche un message d'erreur</p>                               |
| T08 : Etablir un contrôle débit ou un compte rendu | <p>S22 : L'utilisateur consulte un OT</p> <p>S23 : Si la mission de l'OT est une réparation il accède à l'espace ajout compte rendu sinon si la mission est un contrôle débit il accède à l'espace ajout d'un contrôle débit</p> <p>S24: Le système vérifie les champs des informations entrées et l'enregistre, sinon affiche un message d'erreur.</p> |
| T09 : Consultation d'une Panne.                    | <p>S25 : L'utilisateur demande à voir la liste des pannes.</p> <p>S26 : Le système affiche la liste des pannes</p> <p>S27 :L'utilisateur choisit une panne pour afficher les détails.</p>                                                                                                                                                               |
| T10: Ajout d'une remarque panne                    | <p>S28 : L'utilisateur consulte une panne et il click sur le bouton avez-vous une remarque</p> <p>S29 : Le système lui affiche le formulaire souhaité.</p> <p>S30: Le système vérifie les champs des informations et les enregistre, sinon il affiche un message d'erreur</p>                                                                           |
| T11 : Consultation des remarques des Panne.        | <p>S31 : L'utilisateur accède à l'espace panne</p> <p>S32 : Le system affiche le nombre de nouvelle remarque</p> <p>S33 : L'utilisateur demande à afficher la liste des remarques et il click sur l'une d'elles pour voir les détails</p>                                                                                                               |
| T12 : Recherche d'une personne                     | <p>S34 : L'utilisateur choisit le type de recherche et remplit les champs appropriés</p>                                                                                                                                                                                                                                                                |

## [Titre du document]

|                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                          | <p>S35 : Si la recherche a été fructueuse le système retourne la liste des personnes trouvées sinon il affiche un message d'erreur</p>                                                                                                                                                                                                                                                                                                      |
| T13 : Recherche d'une panne              | <p>S36 : L'utilisateur choisit le type de recherche et remplit les champs appropriés</p> <p>S37 : Si la recherche a été fructueuse le système retourne la liste des pannes trouvées sinon il affiche un message d'erreur</p>                                                                                                                                                                                                                |
| T14 : Supprimer une personne             | <p>S38 : L'utilisateur consulte le dossier d'une personne et click sur le bouton Supprimer</p> <p>S39 : Le system affiche un message de confirmation de suppression</p> <p>S40 : Si la suppression est validée la personne est alors supprimer</p>                                                                                                                                                                                          |
| T15 : Modifier le dossier d'une personne | <p>S41 : L'utilisateur consulte le dossier d'une personne et click sur le bouton modifier</p> <p>S42 : Il modifie les champs appropriés et click sur enregistrer</p> <p>S43 : Le system vérifie les champs des informations entrées et affiche un message de confirmation de sauvegarde sinon il affiche un message d'erreur</p> <p>S44 : Si le message de confirmation de sauvegarde est validé le system enregistre les modifications</p> |
| T16 : Supprimer une panne                | <p>S45 : L'utilisateur consulte une panne et click sur le bouton Supprimer</p> <p>S46 : Le system affiche un message de confirmation de suppression</p> <p>S47 : Si la suppression est validée la panne est alors supprimer</p>                                                                                                                                                                                                             |
| T17 : Modifier une panne                 | <p>S48 : L'utilisateur consulte une panne et click sur le bouton modifier</p>                                                                                                                                                                                                                                                                                                                                                               |

## [Titre du document]

|                                    |                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                    | <p>S49 : Il modifie les champs appropriés et click sur enregistrer</p> <p>S50 : Le system vérifie les champs des informations entrées et affiche un message de confirmation de sauvegarde sinon il affiche un message d'erreur</p> <p>S51 : Si le message de confirmation de sauvegarde est validé le system enregistre les modifications</p>                    |
| T18 : Imprimer la DT.              | <p>S52 : L'utilisateur demande l'impression de la DT en cliquant sur « Imprimer la demande ».</p> <p>S53 : Le system génère un fichier PDF</p>                                                                                                                                                                                                                   |
| T19 : Imprimer l'OT.               | <p>S54 : L'utilisateur demande l'impression de l'OT en cliquant sur « Imprimer l'ordre ».</p> <p>S55 : Le system génère un fichier PDF</p>                                                                                                                                                                                                                       |
| T20 : Imprimer le Compte Rendu.    | <p>S56 : L'utilisateur demande l'impression du Compte rendu en cliquant sur « Imprimer le Compte rendu ».</p> <p>S57 : Le system génère un fichier PDF</p>                                                                                                                                                                                                       |
| T21 : Imprimer la CD.              | <p>S58 : L'utilisateur demande l'impression de la CR en cliquant sur « Imprimer le Contrôle debit ».</p> <p>S59 : Le system génère un fichier PDF</p>                                                                                                                                                                                                            |
| T22 : Rappel du mot de passe       | <p>S60 : L'utilisateur click sur le bouton mot de passe oublié</p> <p>S61 : Le system affiche le formulaire du rappel du mot de passe avec la question de rappel</p> <p>S62 : L'utilisateur donne sa réponse et click sur ok</p> <p>S63 : Si L'utilisateur a donné la bonne réponse le system affiche son mot de passe, sinon il affiche un message d'erreur</p> |
| T23 : Modification du mot de passe | <p>S64 : L'utilisateur click sur le bouton gestion du compte</p> <p>S65 : Le system affiche le formulaire demandé</p> <p>S66 : L'utilisateur remplit les champs appropriés et click sur valider</p> <p>S67 : Le system vérifie les champs et affiche un message</p>                                                                                              |

|                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                              | de confirmation de modification, sinon il affiche un message d'erreur<br>S68 : Si le message de confirmation de modification est validé le system enregistre les nouvelles données                                                                                                                                                                                                                                                  |
| T24 : Modification du rappel du mot de passe | S69 : L'utilisateur click sur le bouton gestion du compte<br>S70 : Le system affiche le formulaire demandé<br>S71 : L'utilisateur remplit les champs appropriés et click sur valider<br>S72 : Le system vérifie les champs et affiche un message de confirmation de modification, sinon il affiche un message d'erreur<br>S73 : Si le message de confirmation de modification est validé le system enregistre les nouvelles données |

## **VII : Conception :**

### **VII.1 : Diagramme de cas d'utilisation :**

Bien souvent, les utilisateurs ne sont pas des informaticiens. Il leur faut donc un moyen simple d'exprimer leurs besoins. C'est précisément le rôle des diagrammes de cas d'utilisation qui permettent de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML d'analyse d'un système.

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs.

Il ne faut pas négliger cette première étape pour produire un logiciel conforme aux attentes des utilisateurs.

- Diagrammes détaillés de quelques cas d'utilisations :
  - ✓ Diagramme de cas d'utilisation General :

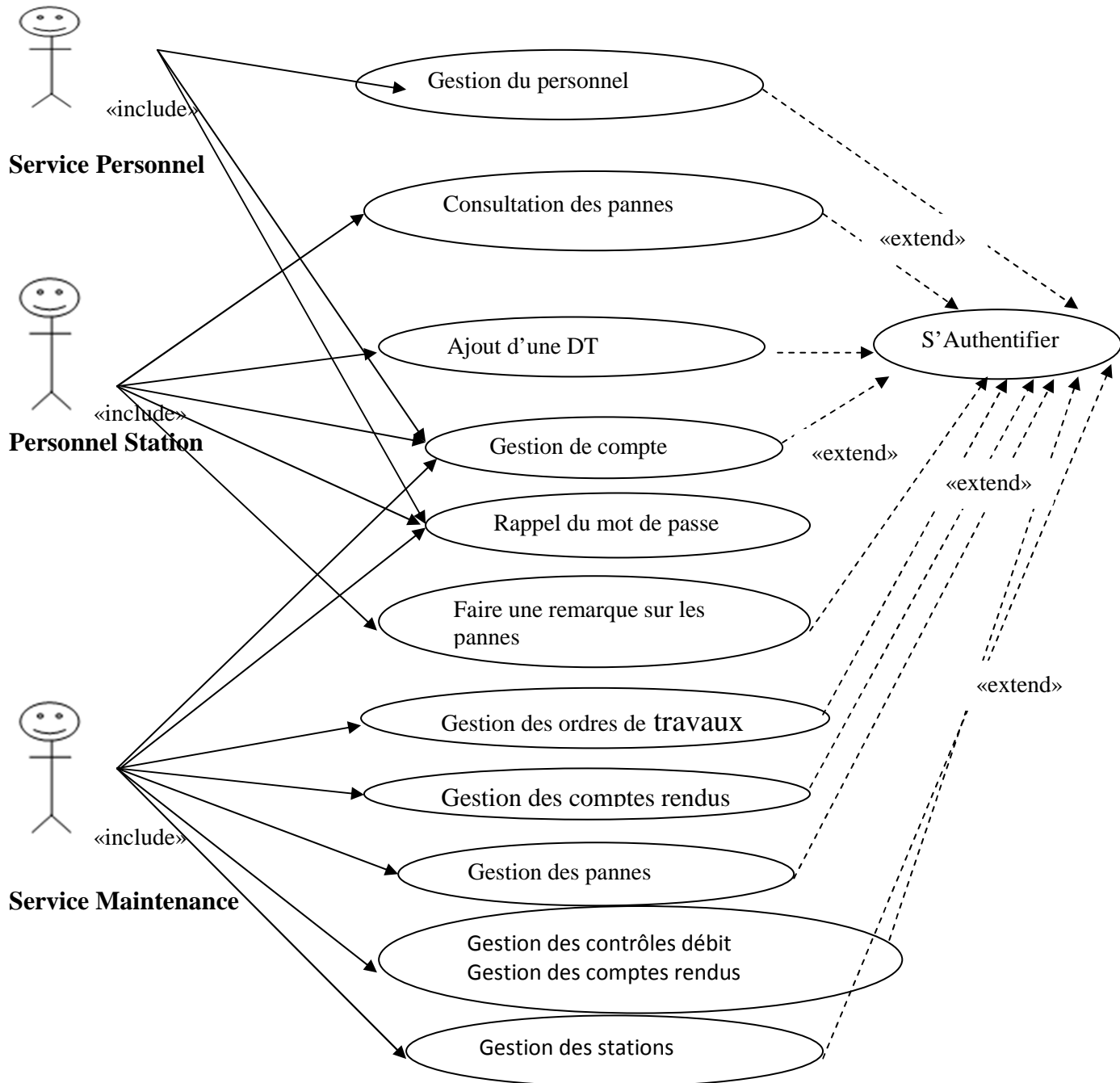
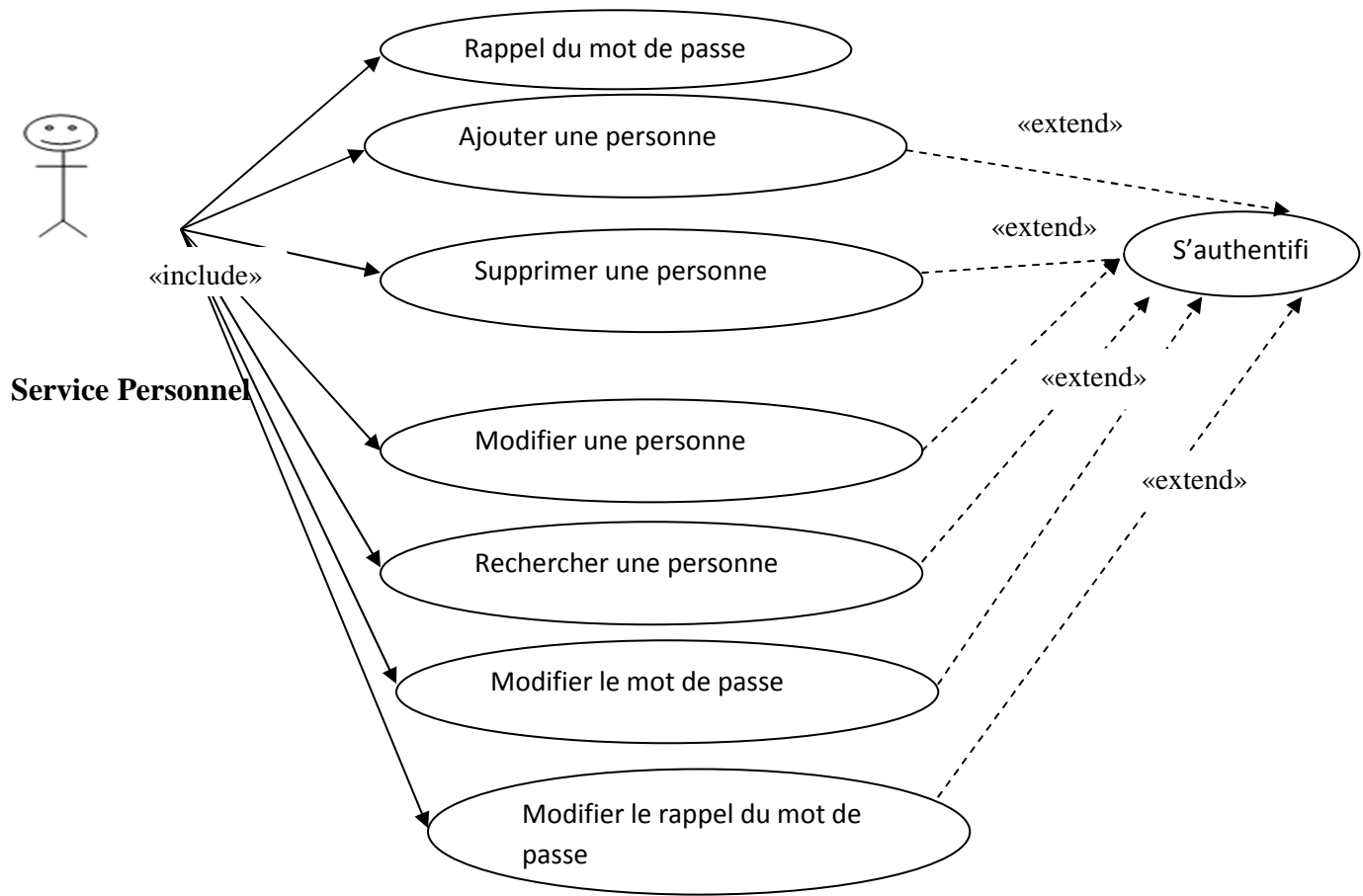


Figure N° III.3 : Diagramme de cas d'utilisation général

✓ Diagramme du cas d'utilisation « Personnel » :



**Figure N° III.4 :Diagramme de cas d'utilisation Personnel**

✓ Diagramme du cas d'utilisation « Service Maintenance » :

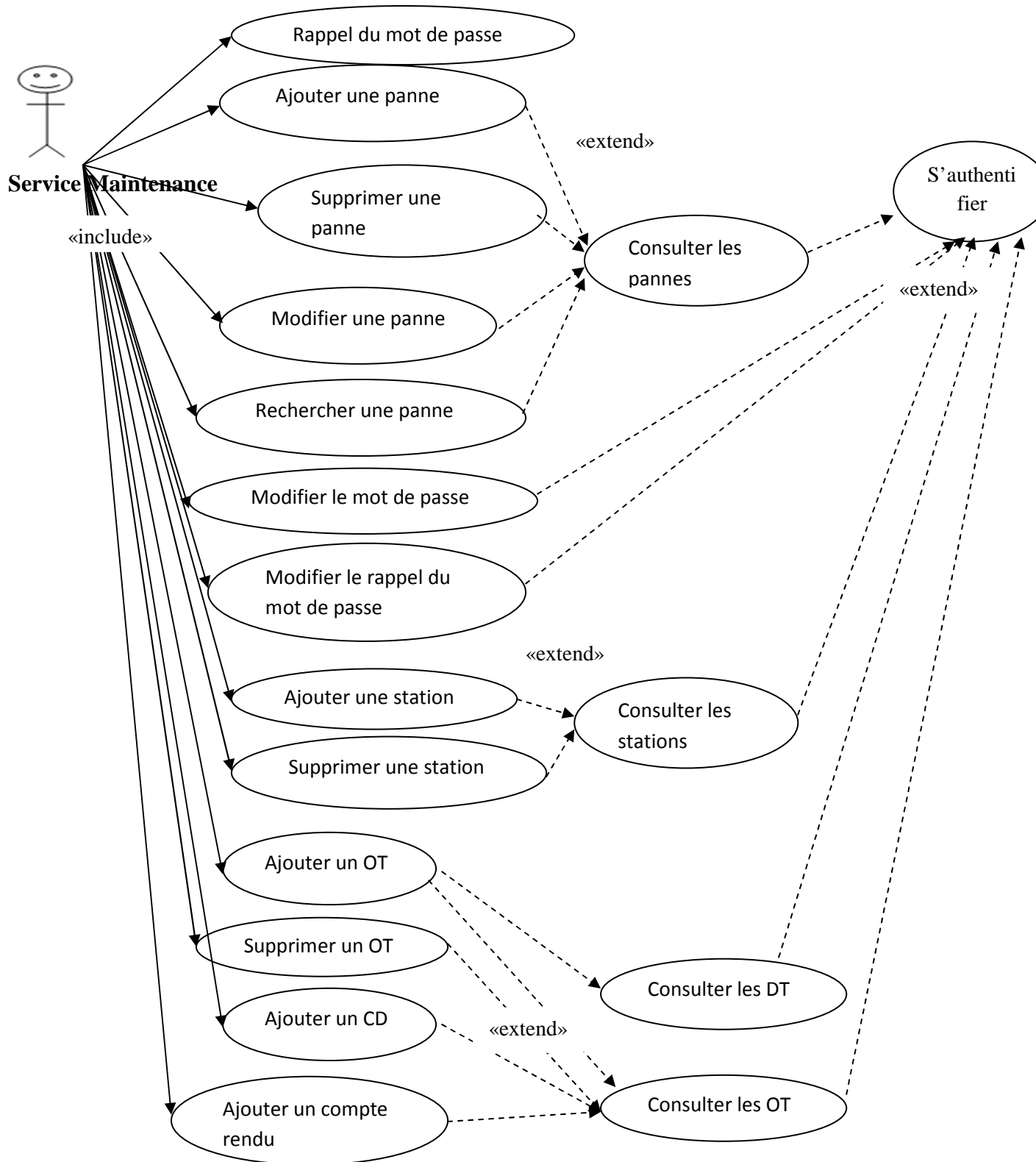
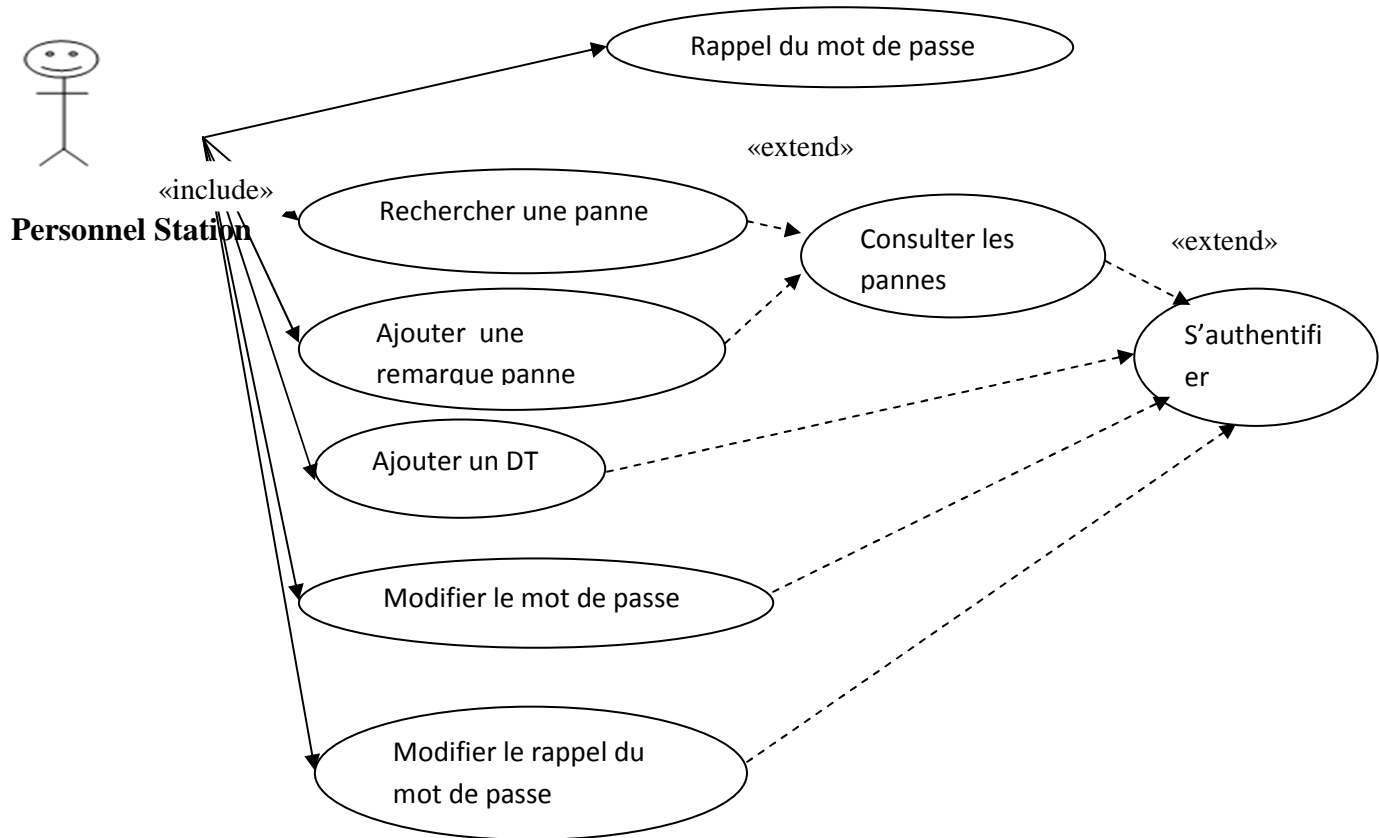


Figure N° III.5 : Diagramme de cas d'utilisation Service Maintenance

✓ Diagramme du cas d'utilisation « Personnel Station » :



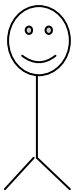
**Figure N° III.6 : Diagramme de cas d'utilisation Personnel**

## VII.2 : Diagramme de séquence :

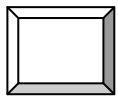
Il permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets. Un diagramme de séquence montre une interaction présentée en séquence dans le temps. Voici quelques notions de base du diagramme :

1. Scénario : une liste d'actions qui décrivent une interaction entre un acteur et le système.
2. Interaction : un comportement qui comprend un ensemble de messages échangés par un ensemble d'objets dans un certain contexte pour accomplir une certaine tâche.
3. Message : Un message représente une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une réaction chez le récepteur.

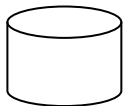
### Remarque :



**Utilisateur**

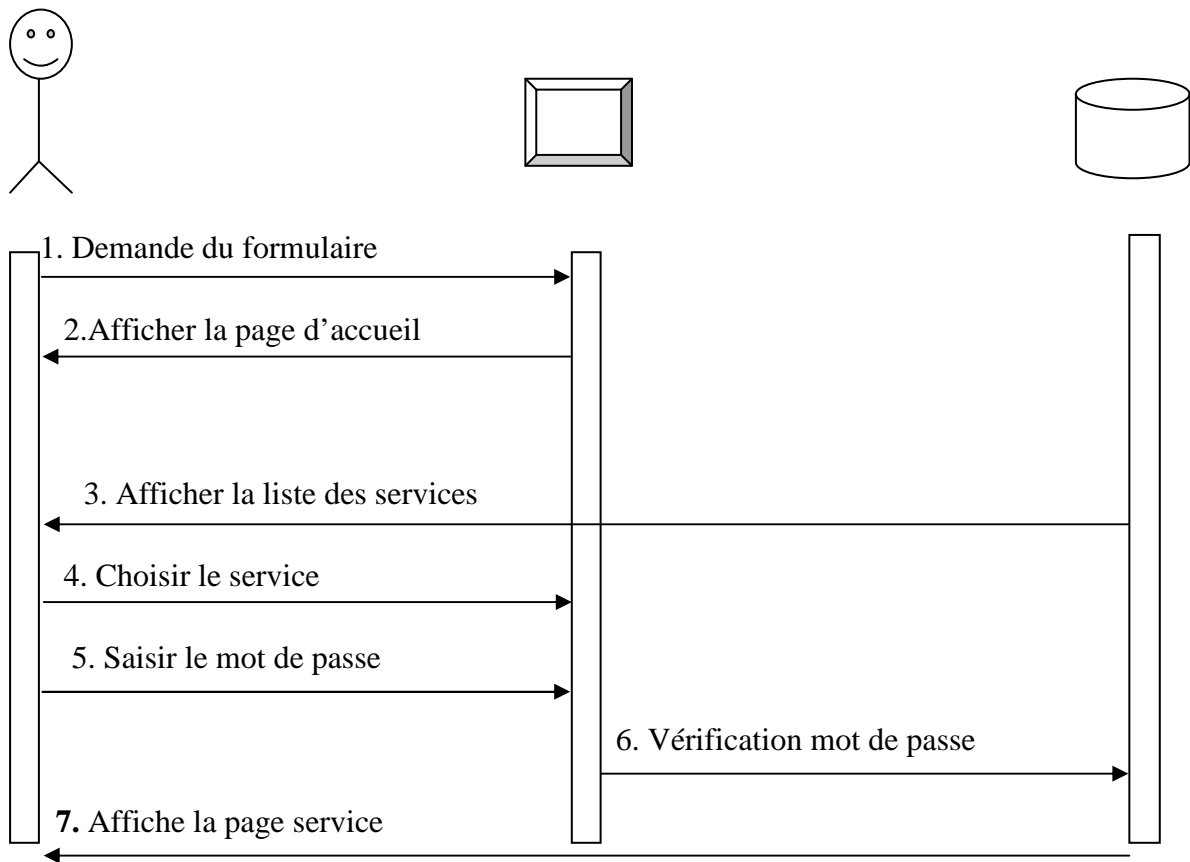


**Système**



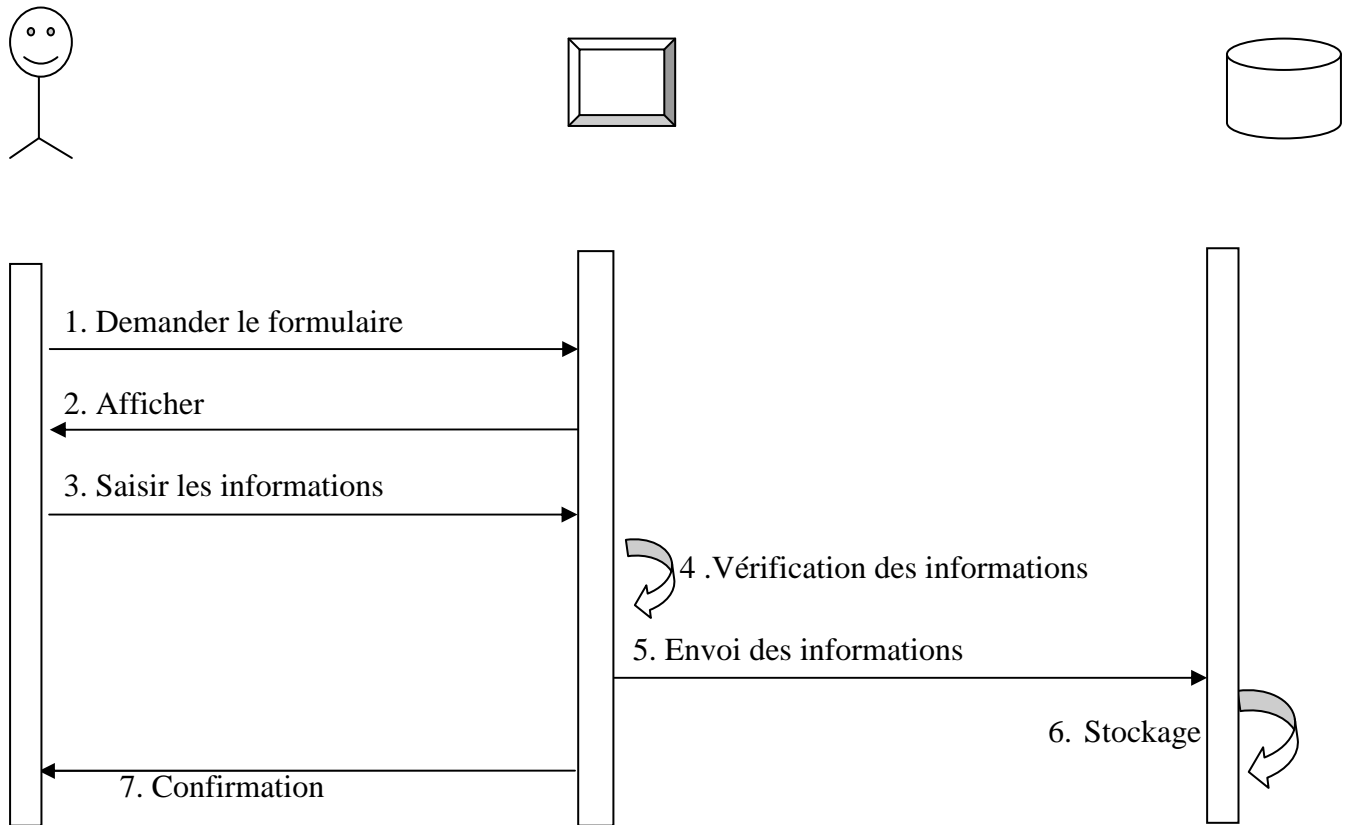
**Base de données**

✓ Diagramme de séquence « cas Authentification » :



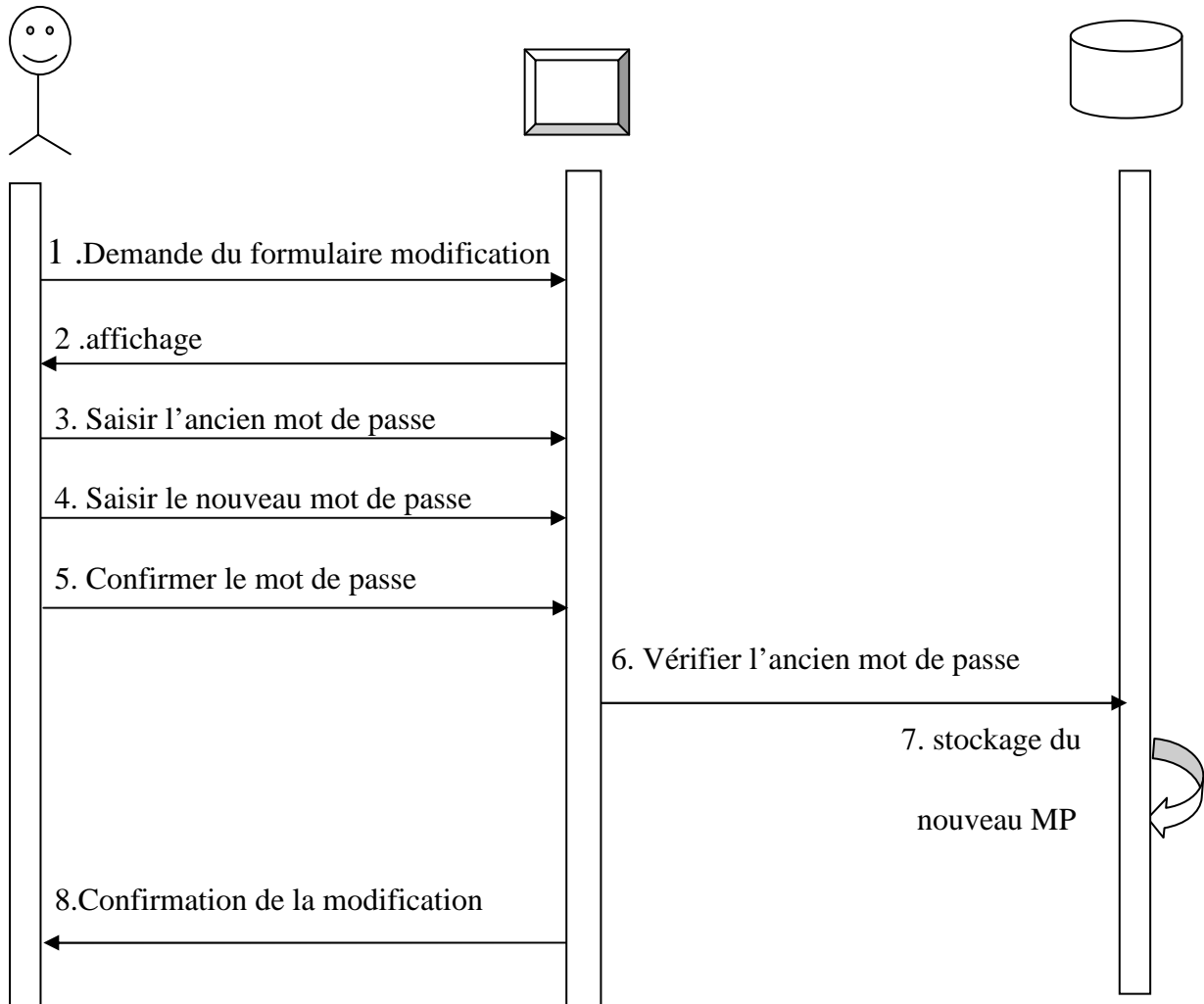
**Figure N° III.7 : Diagramme de séquence «cas Authentification »**

✓ Diagramme de séquence « cas Ajout » :



**Figure N° III.8 : Diagramme de séquence «cas Ajout »**

✓ Diagramme de séquence « cas modification du mot de passe »



**Figure N° III.9 : Diagramme de séquence «cas modification du mot de passe »**

### VII.3 : Diagramme d'activité :

Le diagramme d'activité donne une vision des enchaînements des activités propre à une opération ou à un cas d'utilisation.

Le diagramme d'activité est attaché à une catégorie de classes et décrit le déroulement des activités de cette catégorie. Le déroulement s'appelle "flot de contrôle". Il indique la part prise par chaque objet dans l'exécution d'un travail. Il sera enrichi par les conditions de séquence.

#### Remarque :

Utilisateur qui exécute l'application



Le résultat affiché après exécution de l'application

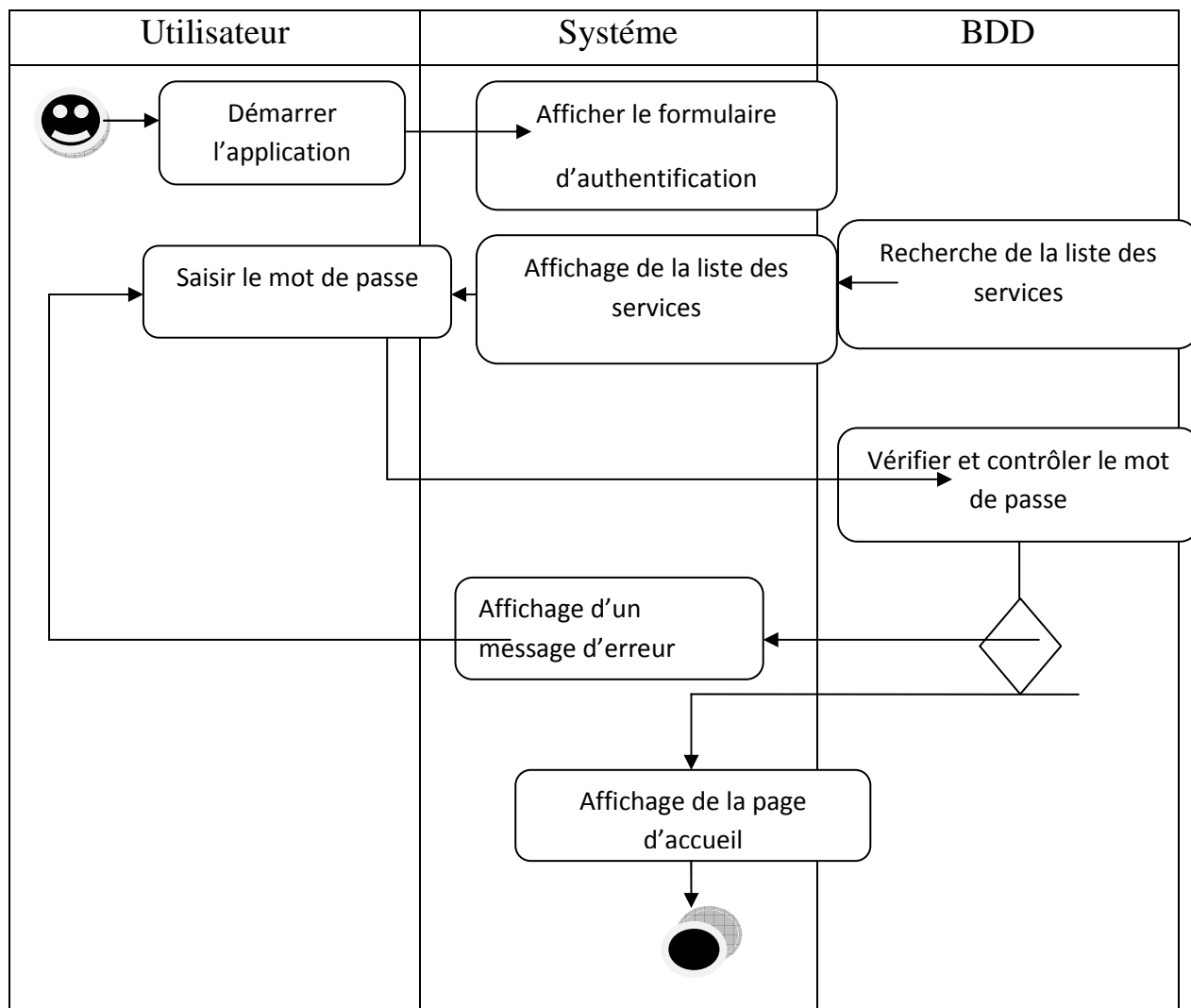


Le teste de validation (oui ou non)



✓ **Diagramme d'activité de l'authentification :**

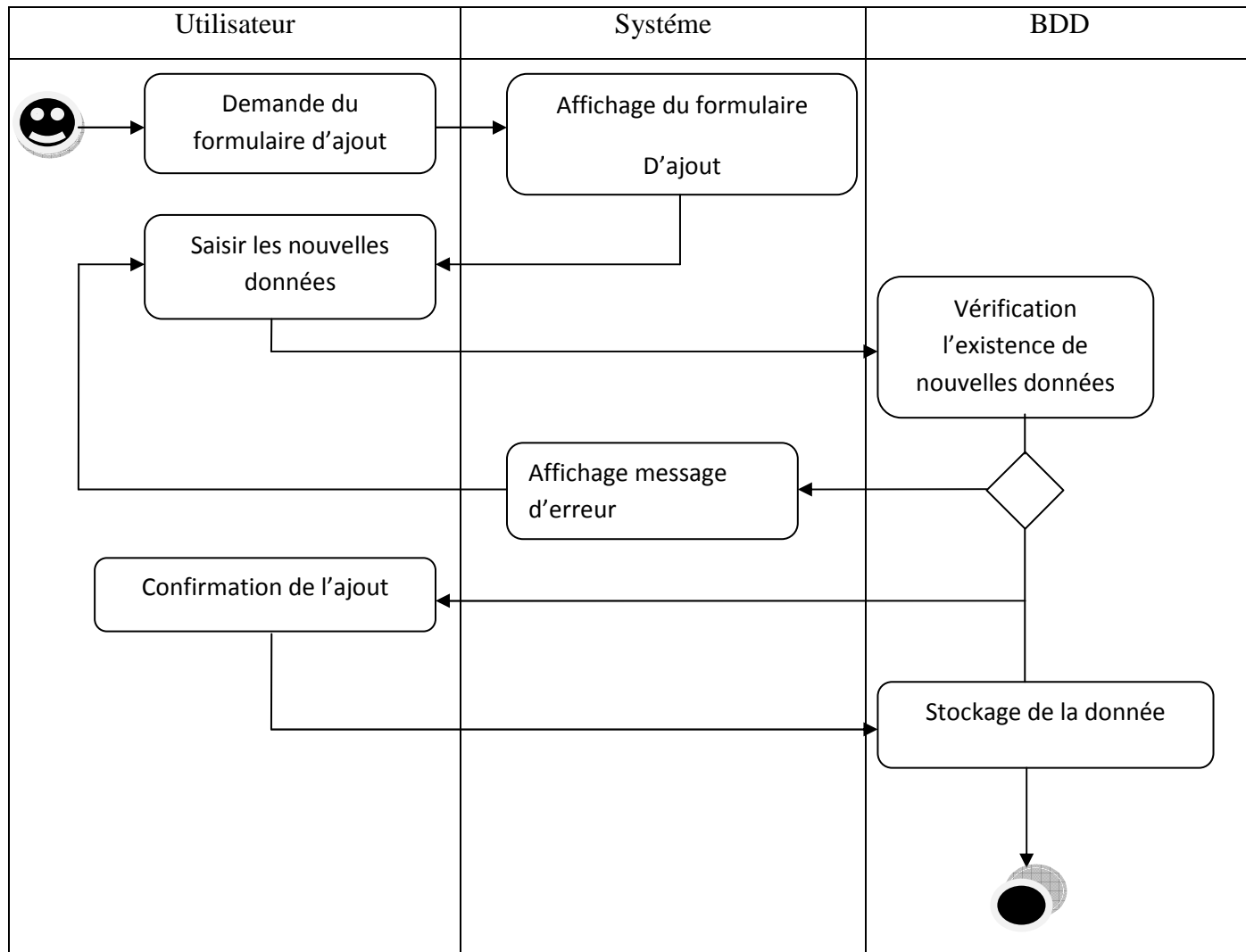
Le diagramme d'activité d'authentification nous permet de voir les comportements internes du système, lors du démarrage de l'application par l'utilisateur, le système lui affiche le formulaire d'authentification, après que le mot de passe soit saisi le système vérifie sa validité et affiche la page d'accueil sinon il affiche un message d'erreur



**Figure N° III.10 : Diagramme d'activité de l'authentification**

✓ **Diagramme d'activité d'ajout :**

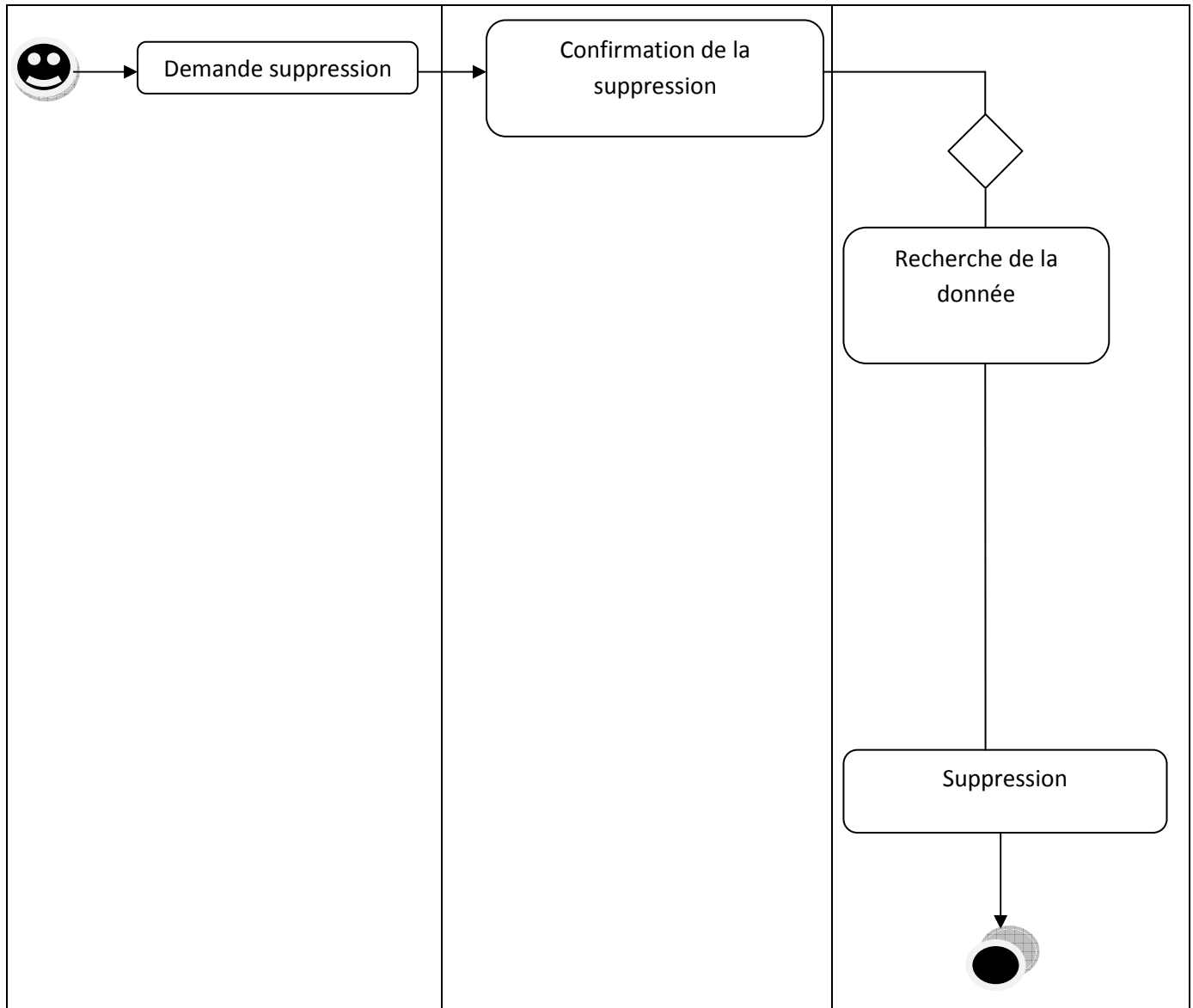
Après une demande d'ajout d'une donnée par l'utilisateur (une DT, une panne, une station, un personne), le système lui affiche le formulaire d'ajout pour qu'il puisse saisir ces données et confirmer leur enregistrement au niveau de la base de données.



**Figure N°III. 11 : Diagramme d'activité d'ajout**

✓ **Diagramme d'activité de suppression :**

Après une demande de suppression d'une donnée par l'utilisateur (une personne), le système lui affiche le formulaire de suppression pour qu'il puisse saisir le code et le faire supprimer puis confirmer la suppression au niveau de la base de données.



**Figure N° III.12 : Diagramme d'activité de suppression**

✓ Diagramme d'activité de recherche :

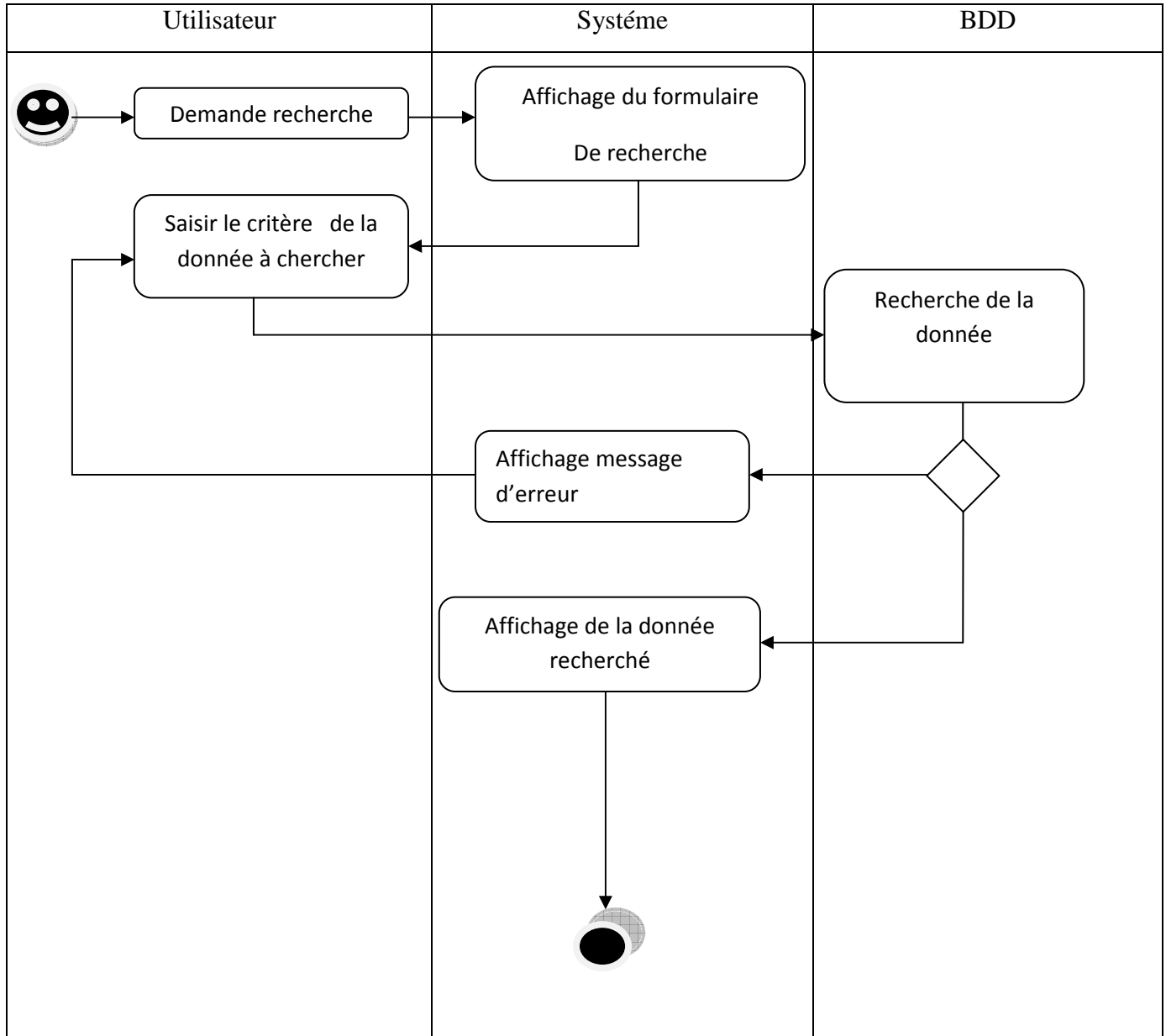
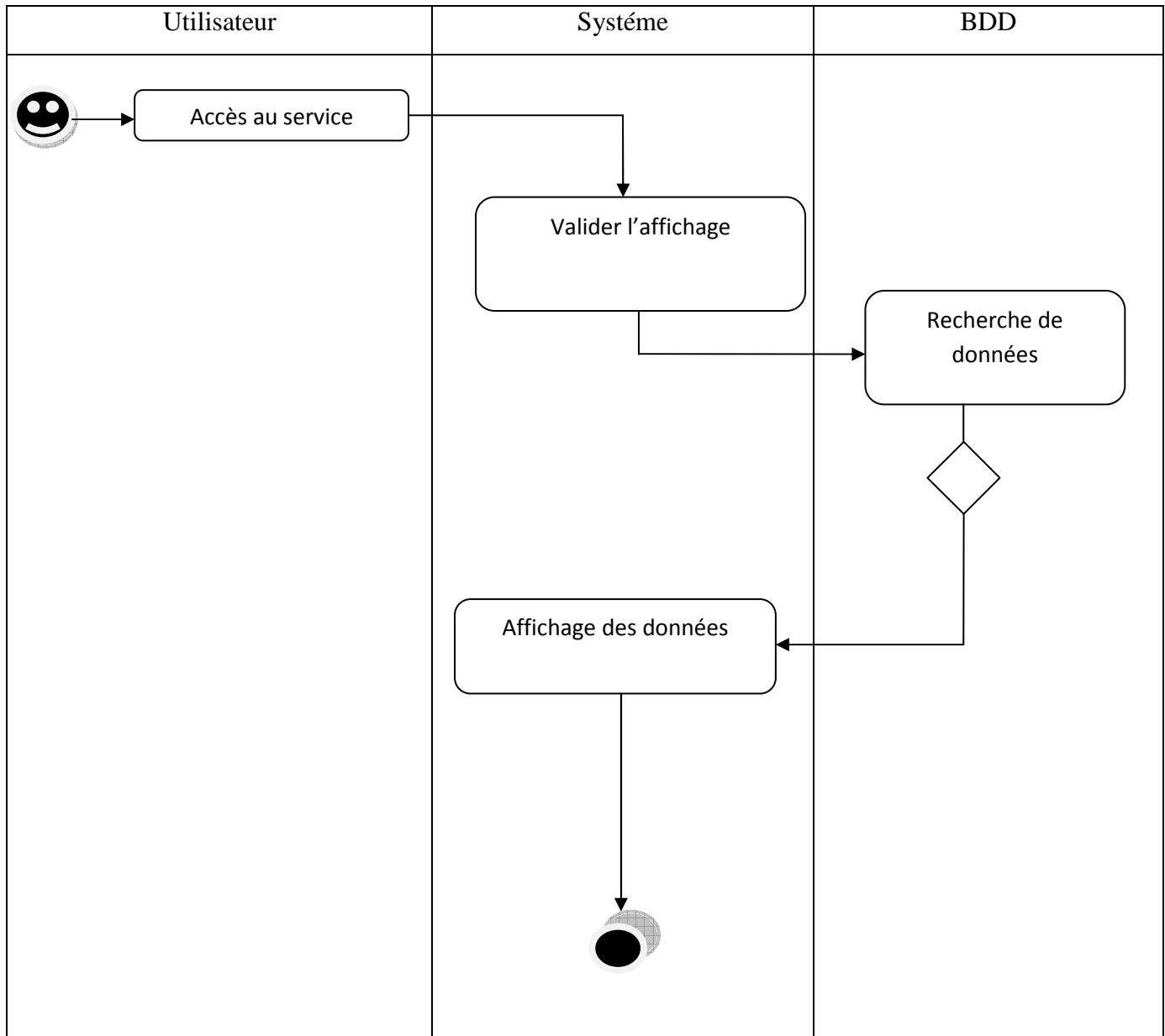


Figure N° III.13 : Diagramme d'activité de recherche

✓ Diagramme d'activité de consultation :



**Figure N° III.14: Diagramme d'activité de consultation**



**Règles de gestion :**

Les relations entre les tables doivent respecter les relations suivantes :

- Le service personnel gère tous les employés, et les employés sont gérés uniquement par le service personnel.
- Le service maintenance gère toutes les pannes, et les pannes sont gérées uniquement par le service maintenance.
- Les remarques sur les pannes sont consultées uniquement par le service maintenance, et ce dernier peut consulter toutes les remarques pannes.
- Le service maintenance peut ajouter ou supprimer une ou plusieurs stations, une station n'est ajoutée ou supprimée que par le service maintenance.
- Le service maintenance consulte toutes les DT, une DT n'est consultée que par le service maintenance.
- Les OT sont gérés uniquement par le service maintenance, et le service gère tous les OT.
- Une DT peut ne pas concerner une panne comme elle peut en concerner plusieurs, et une panne peut être mentionnée dans une ou plusieurs DT.
- Une remarque panne concerne une et une seule panne, et une panne peut être l'objet de plusieurs remarques.
- Une station établit une ou plusieurs remarques pannes.
- Une station établit une ou plusieurs DT, et une DT n'est établie que par une et une seule station.
- Une station possède plusieurs employés, et un employé peut ne pas travailler dans une station.
- Une DT donne lieu à un et un seul OT, un OT n'est établi qu'avec l'existence d'une DT, et il ne concerne que celle-ci.
- Un OT donne lieu à un ou plusieurs comptes rendus, et un compte rendu n'est établi qu'après parution d'un OT.
- Un OT donne lieu à un ou plusieurs CDB, et un CDB n'est établi qu'après parution d'un OT.

**VII.5 : Conception de la base de données**

Dans ce niveau, le travail consiste à traduire le diagramme de classe de façon à optimiser l'implémentation de la base de données pour finir par le modèle physique qui représente l'implémentation des tables de la base de données.

### **VII.6 : Le modèle relationnel :**

A partir de la description conceptuelle que nous avons effectuée, on peut réaliser le modèle relationnel; ça en utilisons des règles de passages de l'UML vers le relationnel.

Quelques notions essentielles :

- **Domaine** : c'est l'ensemble des valeurs d'un attribut.
- **Relation** : c'est un sous ensemble du produit cartésien d'une liste de domaines.

C'est en fait un tableau à deux dimensions dont les colonnes correspondent aux Domaines et dont les lignes contiennent des tuples. On associe un nom à chaque colonne.

- **Attribut** : c'est une colonne d'une relation, caractérisé par un nom.
- **Tuple** : c'est la liste des valeurs d'une ligne d'une relation.
- **Cardinalité** : elle permet de définir les conditions de participation d'une entité à une relation. Toutefois, une entité peut participer à plusieurs relations.
- **L'arité** : est le nombre d'attributs d'une relation.
- **Clé** : On distingue deux types de clés:

**Clé primaire** : ensemble d'attributs dont les valeurs permettent de distinguer les n-tuplets les uns des autres (notion d'identifiant).

**Clé étrangère** : Attribut qui est clé primaire d'une autre entité.

**NB** : pour la notation, nous avons choisi de mettre en gras-souligné les clés primaires et de mettre une \* à la fin de chaque clé étrangère :

- Personnel (Idpersonne, Nompersonne, Prenompersonne, Fonctionpersonne, Domainepersonne, Datenaiss, Lieunaiss, Situationpersonne, Telpersonne, Adressepersonne, Sexe).
- CDB (IdCDB, DateCDB, IdOT\*, Idstation\*, Idpistolet, Essai1, Essai2, Essai3).
- Crendu (Idcrendu, Idpersonne\*, IdOT\*, Datecrendu, Idpiece, Duretrav, Observation).
- DT (IdDT, DateDT, Idpersonne\*, Idstation\*, Typemission, Remarque, Idpanne\*, IndexDT).
- OT (IdOT, IdDT\*, Idpersonne\*, mission, dateOT).
- Panne (Idpanne, Typepanne, , Solpanne, Prioritepanne, Descpanne).
- Station (idstation\*, pwdstation, localitestation, distancestation, questionsec, reponsesec).
- Admin(Idadmin, Pwdadmin, Questionsecadmin, Reponsesecadmin).
- Rqpanne(Idrq, Idpanne\*, Idstation, Solpanne, Descpanne, etatrq).

**VII.7 : Dictionnaire de données :**

| Codification      | Désignation                  | Type      | Taille | Observation   |
|-------------------|------------------------------|-----------|--------|---------------|
| Idpersonne        | Identifiant personne         | Caractère | 45     |               |
| Nompersonne       | Nom personne                 | Caractère | 45     |               |
| Prenompersonne    | Prénom personne              | Caractère | 45     |               |
| Fonctionpersonnee | Fonction personne            | Caractère | 45     |               |
| Domainpersonne    | Domain personne              | Caractère | 45     |               |
| Idcdb             | Identifiant contrôle débit   | Caractère | 45     |               |
| Datecdb           | Date contrôle débit          | Date      | 10     | jj/mm/aaaa    |
| Idot              | Identifiant ordre de travaux | Caractère | 45     |               |
| Idcrendu          | Identifiant compte rendu     | Caractère | 45     |               |
| Datecrendu        | Date compte rendu            | Date      | 10     | jj/mm/aaaa    |
| Duretrav          | Durée travaux                | Integer   | 10     | Hh/mm         |
| Datedt            | Date demande travaux         | Date      | 10     | jj/mm/aaaa    |
| Idstation         | Identifiant station          | Caractère | 45     |               |
| Idpanne           | Identifiant panne            | Caractère | 45     |               |
| Index             | Index                        | entier    | 9      | XXXXXX<br>XXX |
| Mission           | Mission                      | Caractère | 45     |               |
| Typepanne         | Typepanne                    | Caractère | 45     |               |
| Solpanne          | Solustionpanne               | Caractère | 200    |               |
| Priopanne         | Priorité de la panne         | Caractère | 45     |               |
| Idpistolet        | Identifiant pistolet         | Caractère | 45     |               |

**Figure N° III.15: Dictionnaire de données**

### **VIII. Conclusion :**

Dans ce chapitre, nous avons pu cerner les besoins de notre champ d'étude et entrevoir les différents aspects de l'application à réaliser, en nous basant sur les diagrammes du langage UML à savoir le diagramme de cas d'utilisation, le diagramme de collaboration, le diagramme de séquence et le diagramme de classe.

A ce niveau du développement, on est prêt à mettre en œuvre l'application qui sera présentée dans le prochain chapitre.

# Chapitre IV

## Réalisation

## **I- Introduction**

Après avoir présenté dans le chapitre précédent la conception et le fonctionnement global du système, nous arrivons dans ce chapitre à la mise en œuvre de notre application.

Ce chapitre sera divisé en deux parties : la première partie sera consacrée à la description des outils de développement de notre application. La deuxième partie contiendra les détails techniques d'implémentation des différentes fonctionnalités offertes par notre système.

## **II- Partie I : Les Outils de développement**

### **II.1.1- PostgreSQL [A&J&C 90]**

L'information est une ressource capitale pour le bon fonctionnement de n'importe quelle structure, ce qui recommande l'utilisation d'un SGBD pour sa sécurisation et sa structuration. Le SGBD PostgreSQL que nous présentons dans ce qui suit est l'un des systèmes les plus recommandés.

#### **II.1.1.1- Présentation de PostgreSQL**

PostgreSQL™ est un système de gestion de bases de données relationnelles objet (ORDBMS) fondé sur POSTGRES, Version 4.2™. Ce dernier a été développé à l'université de Californie au département des sciences informatiques de Berkeley. POSTGRES est à l'origine de nombreux concepts qui ne seront rendus disponibles au sein de systèmes de gestion de bases de données commerciales que bien plus tard. PostgreSQL™ est un descendant Open Source du code original de Berkeley. Il supporte une grande partie du standard SQL tout en offrant de nombreuses fonctionnalités modernes :

- requêtes complexes ;
- clés étrangères ;
- triggers ;
- vues ;
- intégrité transactionnelle ;
- contrôle des versions concurrentes (MVCC ou multiversion concurrency control).

De plus, PostgreSQL™ peut être étendu par l'utilisateur de multiples façons. En ajoutant, par exemple :

- de nouveaux types de données ;
- de nouvelles fonctions ;
- de nouveaux opérateurs ;
- de nouvelles fonctions d'agrégat ;

- de nouvelles méthodes d'indexage ;
- de nouveaux langages de procédure.
- Et grâce à sa licence libérale, PostgreSQL™ peut être utilisé, modifié et distribué librement, quel que soit le but visé, qu'il soit privé, commercial ou académique.

### II.1.1.2- Vue sur notre base de données sur POSTGRES

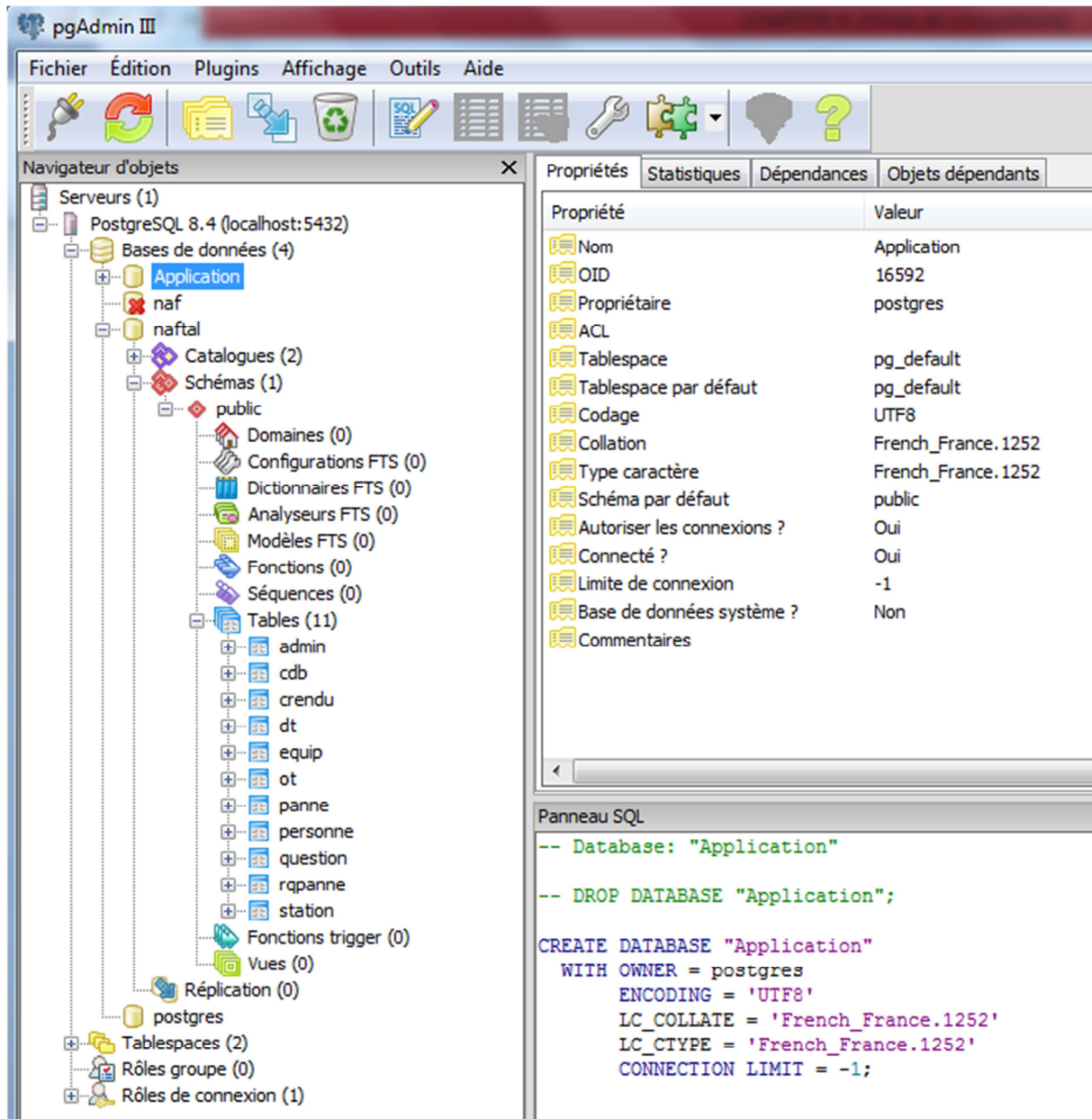


Figure IV.1 interface de notre BDD sur POSTGRES

## II.1.2-Le langage Java

Le sujet de notre travail n'étant pas java lui-même en tant que langage mais juste le langage dans lequel seront écrits les programmes de notre application, nous ferons tout de même un tour d'horizon de ce puissant langage tout en restant le plus près possible de notre sujet.

### II.1.2.1-La programmation graphique

Dans la création d'IHM, on distingue deux grandes parties :

➤ **la partie statique :**

qui constitue l'aspect graphique de l'interface (menu, boutons, boites de dialogue, ...etc.) et

➤ **la partie dynamique :**

qui définit et gère les actions à effectuer en fonction d'événements générés par l'utilisateur (ex : un clic souris sur un bouton provoque l'ouverture d'une boite de dialogue ou une fenêtre).

Java met à la disposition du programmeur

- ✓ **le package Swing** (extension de l'ancien package **AWT** –Abstract Windowing Toolkit) offre tous les composants nécessaires à la création des IHM. Il contient de nombreuses classes permettant de réaliser des interfaces graphiques. Sa classe de base est la classe **JComponent**. Elle contient quatre classes définissant des composants de «haut niveau» :

**a. JFrame** est la classe permettant de faire une «application». Une instance de **JFrame** est composée d'un **JRootPane** lui-même composé de :

✚ **LayeredPane**

✚ **MenuPane**: le menu

✚ **ContentPane**: les composants

✚ **GlassPane**: peut servir à intercepter des événements souris.

✚ **Décorations** : bordure, titre, boutons d'icônes et fermeture, icône

**b. JWindow**: définit une fenêtre sans le cadre.

**c. JDialog**: définit une fenêtre de dialogue, qui peut être modale ou non.

**d. JApplet**: définit un applet.

- ✓ La classe **Event** est utilisée pour la gestion des événements.

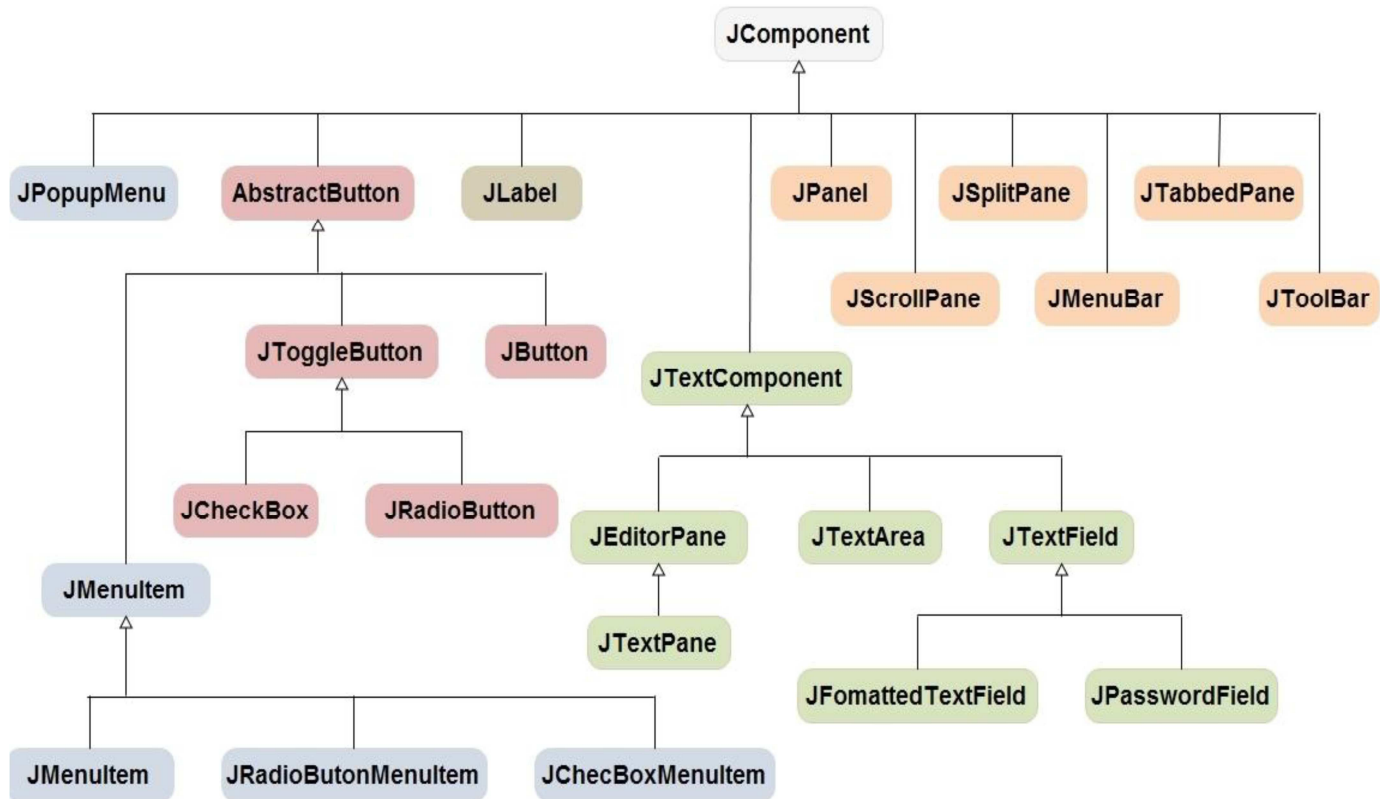


Figure IV.2: les classes de base de la classe JComponent.

### II.1.2.2-Environnement de développement NetBeans

NetBeans est un projet open source fondé par Sun Microsystems. L'IDE NetBeans est un environnement de développement permettant d'écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java – mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'IDE NetBeans. L'IDE NetBeans est un produit gratuit.

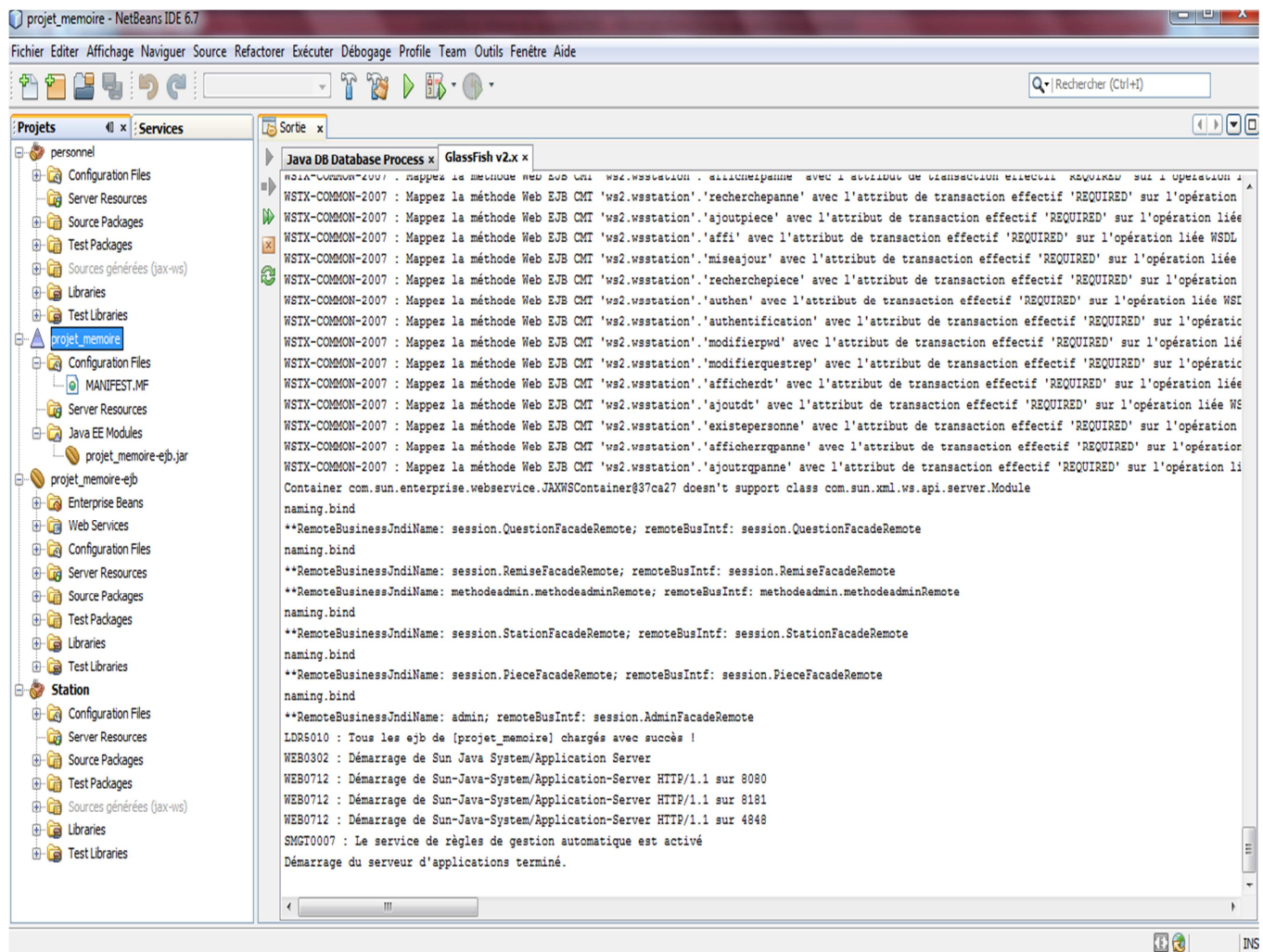
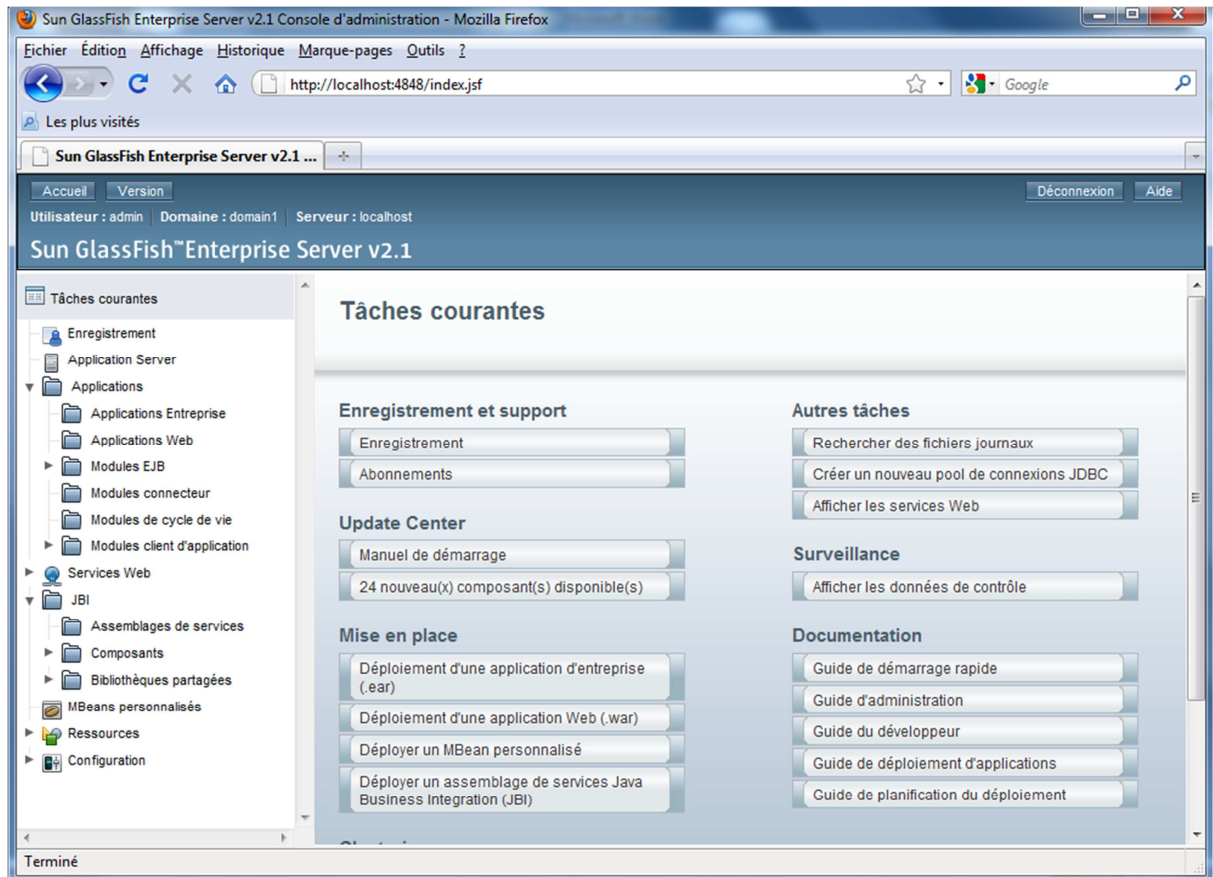


Figure IV.3: l'interface graphique de NetBeans.

### II.1.3-Le serveur d'application (Glassfish)

Glassfish est le nom du [serveur d'applications](#) “[Open Source](#)” [JavaEE 5](#) et désormais [Java EE 6](#) avec la version 3 et qui sert de fondation au produit "Oracle GlassFish Server" (anciennement Sun Java System Application Server de [Sun Microsystems](#)) . Sa partie Toplinkpersistance provient d'[Oracle](#). C'est la réponse aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération.



**Figure IV.4 Interface administrateur du serveur d'application GlassFish.**

### III- Partie II

#### III.1. Implémentation des différentes parties de l'application

##### III.1.1. Partie stockage

Pour l'implémentation de cette partie, Nous avons créé avec le SGBD POSTGRES une base de données [naftal] contenant les différentes tables de chapitre conception.

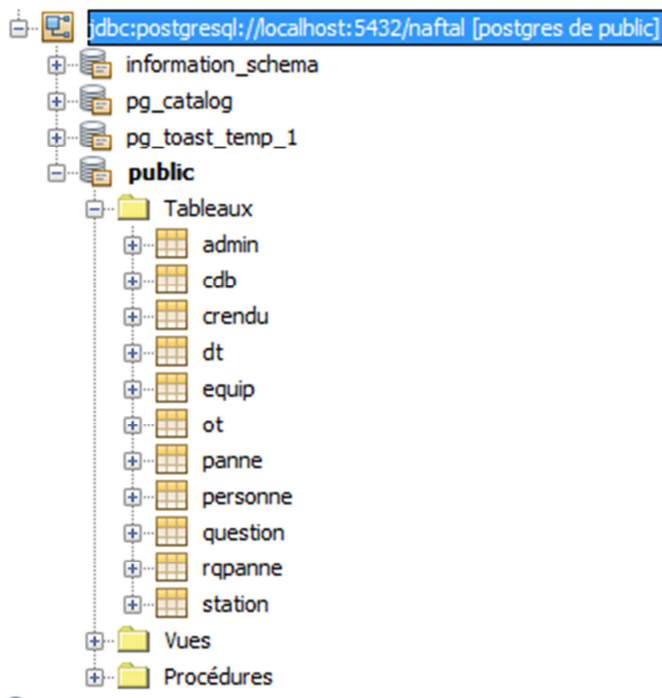
Voici un exemple de script de création d'une des tables :

```
create table panne (idpanne(45) primary key auto_increment,
typepanne(45) not null ,
solpanne(45) not null,
priopanne(45) not null,
descpanne(45)not null
)ENGINE=InnoDB;
```

**Figure IV.5. Script de création de la table panne.**

Après avoir créé cette base de données, on a établie une connexion NetBeans /MYSQL à l'aide de pilote JDBC.

L'image suivante est une capture d'écran montrant la connexion établie, ainsi que les différentes tables de notre BDD.



**Figure IV.6. pilote de connexion JDBC .**

### III.1.2. Partie Traitement

Dans cette section, nous allons faire une description des différents projets NetBeans implémentés.

#### III.1.2.1. le projet [projet\_memoire]

Ce projet est un projet JEE de type Entreprise Application, c'est lui qui nous permet le déploiement de notre service web au sein de serveur Glassfish.

Ce dernier est composé de deux modules :

- Un module de type client application.
- Un module EJB dont dépendent nos web services.

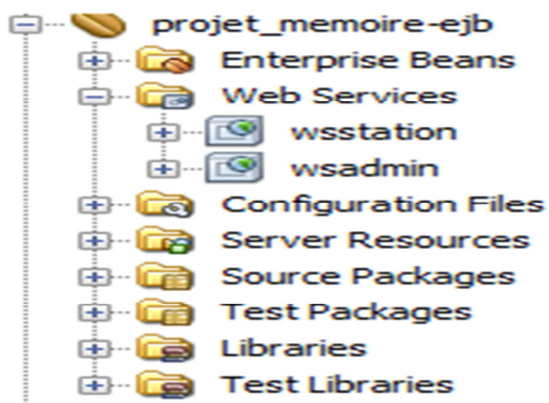
La figure suivante est une capture d'écrans montrant le projet\_memoire.



**Figure IV.7. projet\_memoire.**

### III.1.2.2. le module projet\_memoire-ejb

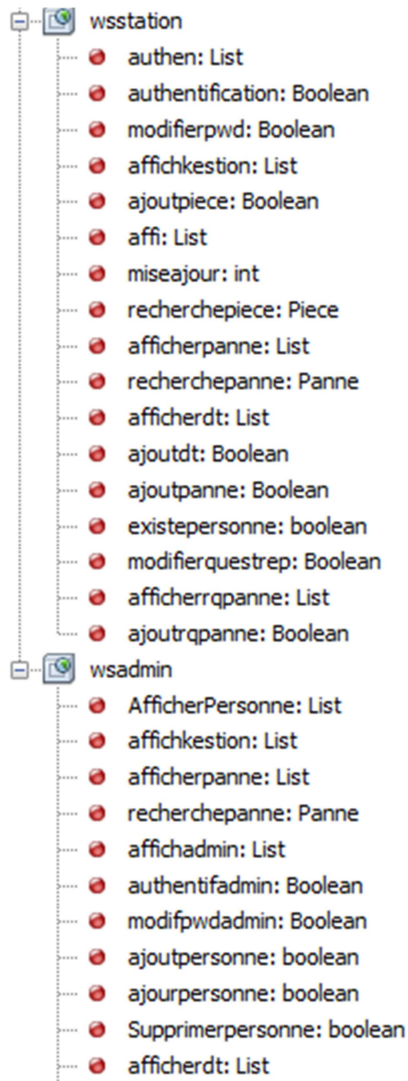
C'est un projet de type EJB Module de la catégorie Java EE des projets NetBeans.



**Figure IV.8. le module projet\_memoire-ejb .**

### III.1.2.3. Les web services

La figure suivante est une capture d'écrans montrant les modules web services.



**Figure IV.9 Les web services.**

➤ **Le déploiement des web services**

L'image suivante est une capture d'écran de la console administrateur du serveur Glassfish montrant les web services déployés, ainsi que les diverses informations qui les caractérisent.

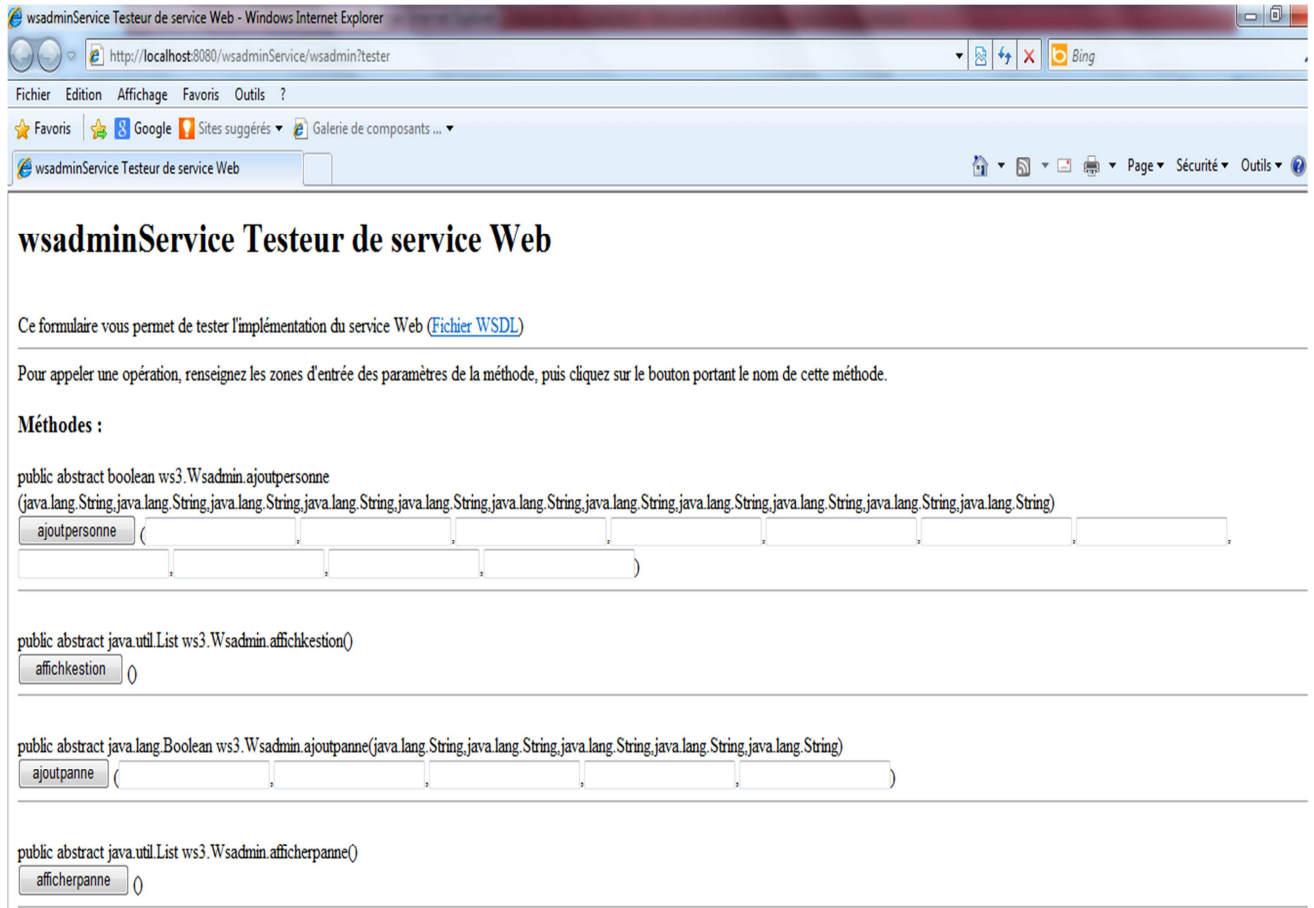


**Figure IV.10. Le déploiement des web services**

➤ **Tester les web méthodes**

Cette console nous permet de tester nos web services en cliquant sur le bouton **tester**, la page de test s'ouvre. Toutes les méthodes (**@WebMethod**) exposées par le service web sont affichées et peuvent être testées.

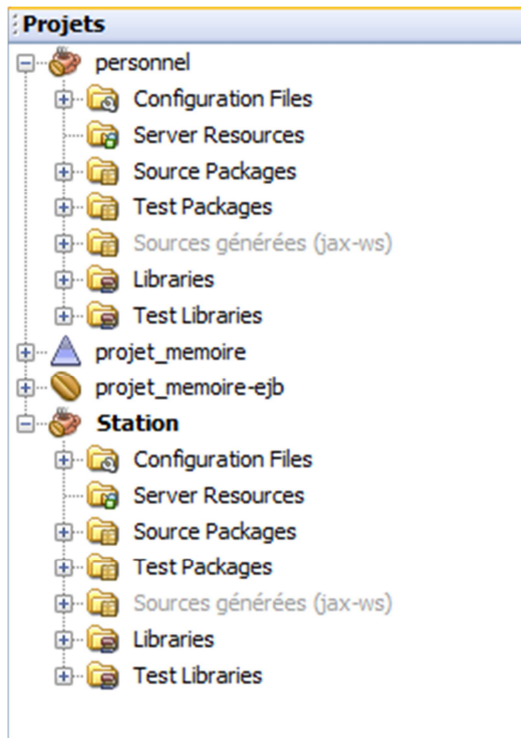
La figure suivante est une capture d'écran montrant cette page de test, ainsi quelques web méthodes qu'elle expose :



**Figure IV.11. page test du web service**

**III.1.2.4. les modules client:**

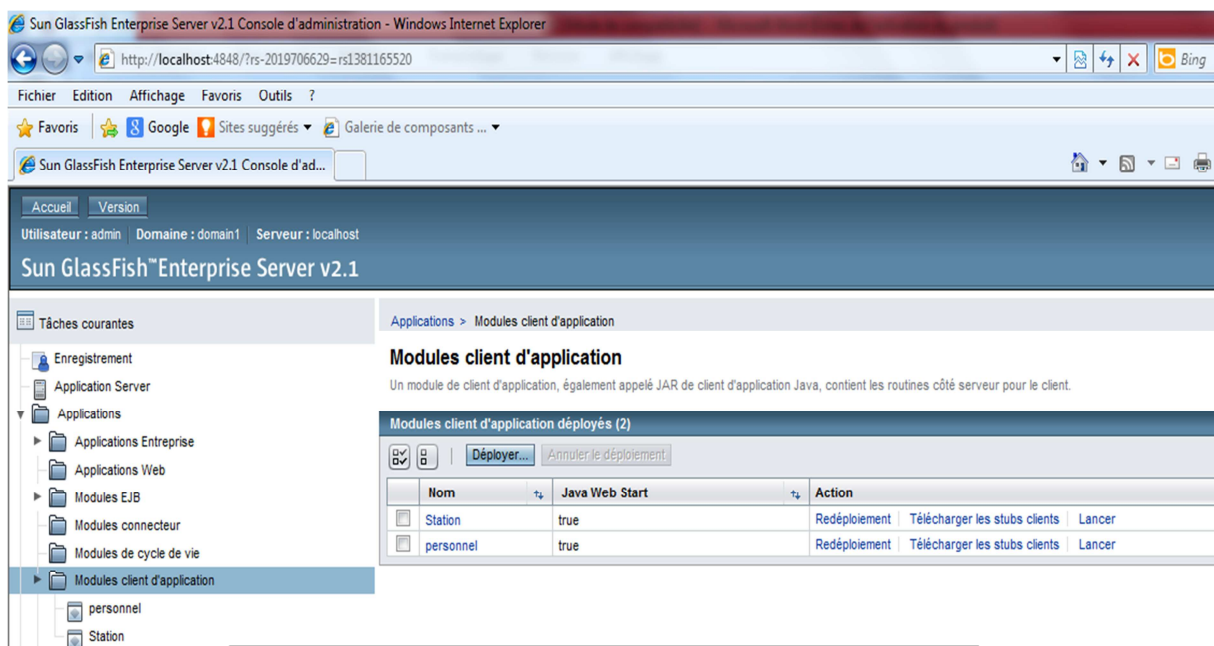
sont des projet de type [Enterprise Application Client] qui s'exécutent sur les machines des utilisateurs, La figure suivante est une capture d'écrans montrant les modulesClient



**Figure IV.13. Module Client.**

➤ **Le déploiement des clients :**

L'image suivante est une capture d'écran de la console administrateur du serveur Glassfish montrant les clients déployés.

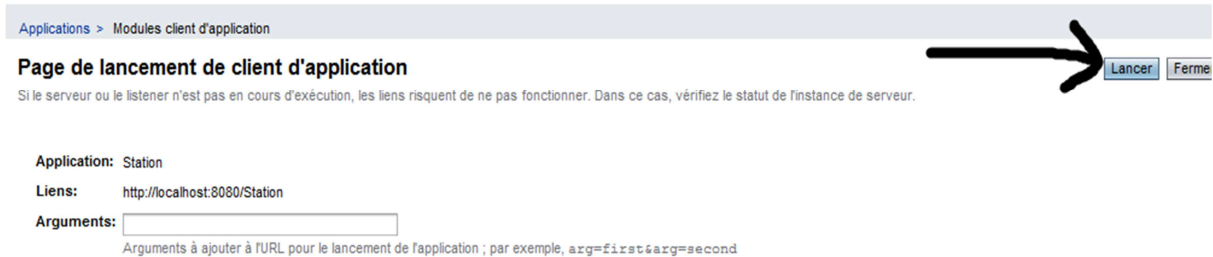


**Figure IV.14. Le déploiement des client**

➤ **Lancer les clients :**

Cette console nous permet de lancer le téléchargement de notre application cliente en cliquant sur le bouton **Lancer**, la page de Lancement s'ouvre puis en télécharge, notre application.

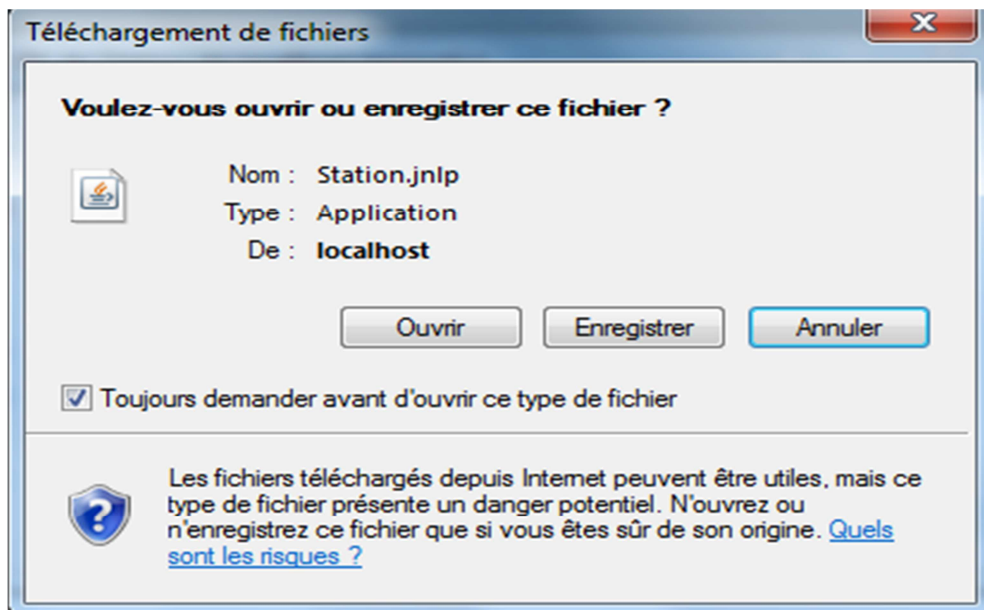
La figure suivante est une capture d'écran montrant cette page de Lancement :



**Figure IV.15. page lancement des client**

En suite le téléchargement sera lancer automatiquement.

La figure suivante est une capture d'écran montrant le téléchargement :



**Figure IV.16. page d'enregistrement ou de lancement du client**

**IV- Présentation de l'application :**

Pour illustrer l'échange de données entre le client et notre système, nous avons réalisé deux applications client : application client pour les stations et application client pour le service maintenance ainsi que le service personnel. Ces applications font appel à des web

méthodes que nous avons développées dans nos web services. À travers les interfaces présentées ci-dessous, nous visons à donner une vue générale de notre application conçue.

#### IV.1. Interfaces authentification :

.L'authentification se fait à travers des interfaces représentée dans les figures IV.5 et IV.6



Figure IV.17: L'interface d'authentification du service maintenance et personnel.



Figure IV.18: L'interface d'authentification des stations.

#### IV.2. Interfaces de gestion des paramètres de compte :

Cette interface permet de modifier le mot de passe ainsi que le rappel de ce dernier.

The image shows a software window titled "Gestion du compte". It contains two distinct sections for account management:

- Modifier votre mot de passe (Yellow background):** This section includes four input fields labeled "Login", "Ancien mot de passe", "Nouveau mot de passe", and "Confirmation du mot de passe". Below these fields are two buttons: "Valider" (with a green checkmark icon) and "Annuler" (with a red 'X' icon).
- Modifier votre rappel du mot de passe (Blue background):** This section features a dropdown menu and a text input field. Below them are also two buttons: "Valider" and "Annuler".

ERROR: ioerror  
OFFENDING COMMAND: image

STACK: