

# MEMOIRE DE STAGE DE FIN D'ETUDE

Pour l'obtention du

## MASTERE PROFESSIONNEL

« Nouvelles Technologies des Télécommunications et Réseaux »

Présenté par :

**Daly Seif Allah**

### Titre

Etude, conception et réalisation d'un tableau de bord pour la supervision  
réseau à l'aide d'un outil open source « Nagios »

Soutenu le : 08/02/2014

Devant le jury :

Mme Hanen Idoudi

Mme Ahlem Ben Hsin

Mme Houda Rekaya Houissa

# **Remerciements**

*A la clôture de ce travail et avant d'entamer la rédaction de ce rapport, nous tenons à exprimer dans un premier temps, nos remerciements les plus sincères à **Mr Nacer Ben Naji** pour la proposition de ce sujet, sa disponibilité, son soutien et sa précieuse aide lors de l'élaboration du projet.*

*Nous remercions aussi l'ensemble du personnel du service INM qui a été accueillant, chaleureux et compréhensif.*

*Nous remercions très sincèrement, les membres du jury d'avoir bien voulu accepter de faire la commission d'examineurs.*

*Enfin nous adressons nos plus sincères remerciements à tous nos proches et amis qui nous ont toujours soutenu et encouragés au cours de la réalisation de ce projet ainsi qu'à nos prof qui ont contribué à la réussite de cette formidable année universitaire.*

*Nous adressons aussi nos sincères remerciements à tous ceux qui ont participé de près ou de loin à la réalisation de ce projet, aux personnes qui, malgré que leurs noms ne figurent pas dans ce document, étaient toujours prêtes à aider et à contribuer dans le bon déroulement de ce travail.*

# Dédicaces

---

*Je dédie ce modeste travail comme un témoignage d'affection, de respect et d'admiration :*

*À mes très chers parents,*

*Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi, Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir. Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de vos prières quotidiennes.*

*À mon frère et ma sœur,*

*Qu'ils puissent trouver dans ce modeste mémoire l'expression de mon attachement et de ma profonde reconnaissance.*

*À toute la famille et à tous mes chers amis,*

*Qu'ils trouvent dans ce travail l'expression de mon profond respect et reconnaissance.*

*Daly Seif Allah*





# INTRODUCTION GÉNÉRALE

Les notions de science et de technique se transforment à un rythme rapide depuis la grande révolution scientifique du 18ème siècle. Ces changements ont apporté de nouvelles satisfactions, une autonomie renouvelée et une forte dose d'innovation. Mais, de nos jours on parle plutôt des nouvelles technologies de communication et de l'information avec l'informatique qui devient de plus en plus incontournable. Science du traitement rationnel de l'information, l'informatique s'impose plus que jamais, comme un outil précieux dont a besoin toute entreprise qui veut accroître sa productivité et rester compétitive sur le plan international. Ainsi, dans le domaine des entreprises modernes, l'adoption d'un système informatique facilite énormément les processus d'administration (personnelle et matérielle) et de productivité. Un tel outil aide à bien satisfaire les exigences croissantes non seulement des administrateurs, mais aussi des utilisateurs, c'est pourquoi les développeurs conçoivent ce système d'une telle façon que tous les besoins soient satisfaits à travers une bonne gestion de toutes les branches de l'entreprise.

En effet, toutes les entreprises, de nos jours, sont équipées d'un réseau local au minimum, et pour les plus importantes d'entre elles de réseaux longue distance (Wide Area Network), leurs parcs informatiques englobent des centaines, voire des milliers de terminaux engendrés par des serveurs de bases de données et des serveurs de traitements. L'apparition de ces nouveaux environnements informatisés rend la surveillance des éléments clefs de réseaux et de systèmes une opération indispensable, afin de minimiser la perte d'exploitation et garantir que les utilisateurs ne s'aperçoivent pas des anomalies de fonctionnement. Ces anomalies provoquent des conséquences variables en degré pour le fonctionnement au niveau de système d'entreprise. En effet, l'arrêt d'un service de messagerie n'est pas aussi coûteux que la perte de la base de données de son entreprise due à un disque défectueux.

Vu que le système informatique est au cœur des activités des entreprises, sa maîtrise devient primordiale, puisqu'il doit fonctionner pleinement et en permanence pour garantir la



fiabilité et l'efficacité exigées, d'une part et d'autre part, les problèmes liés au système informatique tels que les défaillances, les pannes, les coupures et les différents problèmes techniques doivent être réduits, du fait qu'une indisponibilité du système ou du réseau peut causer des pertes considérables.

C'est pourquoi, les administrateurs réseaux font appel à des solutions de surveillance et de supervision de réseaux qui sont l'une des disciplines modernes qui connaissent actuellement le plus fort essor.

La supervision de réseaux peut être définie comme l'utilisation de ressources réseaux adaptées dans le but d'obtenir des informations (en temps réel ou non) sur l'utilisation ou la condition des réseaux et de leurs éléments afin d'assurer le bon fonctionnement d'un système ou d'une activité, rapporter et alerter les fonctionnements normaux et anormaux des systèmes informatiques, pour garantir une répartition optimale.

D'où l'intérêt de notre projet de fin d'études qui consiste à développer une application web performante et facile à utiliser qui permet la supervision réseaux l'aide d'un outil open source "Nagios".

Notre rapport est composé de quatre chapitres :

- ✓ Chapitre 1 : Etude préalable
- ✓ Chapitre 2 : Phase d'incubation
- ✓ Chapitre 3 : Conception
- ✓ Chapitre 4 : Réalisation



# ETAT DE L'ART

## I. Présentation de Nagios

Nagios est un logiciel de supervision de réseau libre sous licence GPL qui fonctionne sous Linux.

Il a pour fonction de surveiller les hôtes et services spécifiés, alertant l'administrateur des états des machines et équipements présents sur le réseau.

Bien qu'il fonctionne dans un environnement **Linux**, ce logiciel est capable de superviser toutes sortes de systèmes d'exploitation (Windows XP, Windows 2000, Windows 2003 Server, Linux, Mac OS entre autres) et également des équipements réseaux grâce au protocole SNMP.

Cette polyvalence permet d'utiliser Nagios dans toutes sortes d'entreprises, quel que soit la topologie du réseau et les systèmes d'exploitation utilisés au sein de l'entreprise.

Ce logiciel est composé de trois parties:

- ❖ Le moteur de l'application, qui gère et ordonnance les supervisions des différents équipements.
- ❖ Les Plugins qui servent d'intermédiaire entre les ressources que l'on souhaite superviser et le moteur de Nagios. Il faut bien noter que pour accéder à une certaine ressource sur un hôte, il faut un plugin coté Nagios et un autre coté hôte administré.
- ❖ L'interface web qui permet d'avoir une vue d'ensemble des états de chaque machine du parc informatique supervisé et pouvoir ainsi intervenir le plus rapidement possible en ciblant la bonne panne.

## II. Fonctionnement de Nagios

Le principe de supervision de Nagios repose sur l'utilisation de plugins, l'un installé sur la machine qui supporte Nagios, et l'autre sur la machine que l'on souhaite superviser. Un plugin est un programme modifiable, qui peut être écrit dans plusieurs langages possibles, selon les besoins, et qui servent à récupérer les informations souhaitées.



Nagios, par l'intermédiaire de son plugin, contact l'hôte souhaité et l'informe des informations qu'il souhaite recevoir.

Le plugin correspondant installé sur la machine concernée reçoit la requête envoyée par Nagios et ensuite va chercher dans le système de sa machine les informations demandées. Il renvoie sa réponse au plugin Nagios, qui le transmet au moteur de Nagios afin d'analyser le résultat obtenu et ainsi mettre à jour l'interface web.

### **III. Fonctionnalités [1]**

Les fonctionnalités de Nagios sont très nombreuses, parmi les plus communes nous pouvons citer les suivantes :

- ❖ La supervision des services réseaux (SMTP, http...), des hôtes et des ressources systèmes (CPU, charge mémoire...)
- ❖ Génération de rapports.
- ❖ Gestion des alertes.
- ❖ Surveillance des processus (sous Windows, Unix...).
- ❖ Superviser des services réseaux : (SMTP, POP3, HTTP, NNTP, ICMP, SNMP, LDAP, etc.)

Toutes ces fonctionnalités sont assurées grâce à la gestion et supervision du réseau et ses différentes entités d'une manière centralisée. La figure 1 modélise cet aspect :



Figure1 : Fonctionnalité de Nagios

#### IV. Architecture

L'architecture de Nagios se base sur le paradigme serveur-agent. D'une manière spécifique, un serveur faisant office de point central de collecte des informations tandis que les autres machines du réseau exécutent un agent chargé de renvoyer les informations au serveur.

L'architecture globale de Nagios peut être décomposée en 3 parties coopératives entre elles :

- ❖ **Un noyau** qui est le cœur du serveur Nagios, lancé sous forme de démon et responsable de la collecte et l'analyse des informations, la réaction, la prévention, la réparation et l'ordonnancement des vérifications (quand et dans quel ordre). C'est le principe de répartition des contrôles au mieux dans le temps qui nous évite la surcharge du serveur et des machines à surveiller.
- ❖ **Des exécutants** : ce sont les plugins dont un grand nombre est fourni de la base, responsables de l'exécution des contrôles et tests sur des machines distantes ou locales et du renvoi des résultats au noyau du serveur Nagios.
- ❖ **Une IHM** : C'est une interface graphique accessible par le web conçue pour rendre plus exploitable les résultats. Elle est basée sur les CGI (Common Gateway Interface)

fournis par défaut lors de l'installation de Nagios qui interprètent les réponses des plugins pour les présenter dans l'interface.

Il est possible de coupler Nagios à une base de données MySQL ou PostgreSQL, lorsque le nombre d'objets à superviser devient conséquent. La figure suivante modélise l'architecture de Nagios.

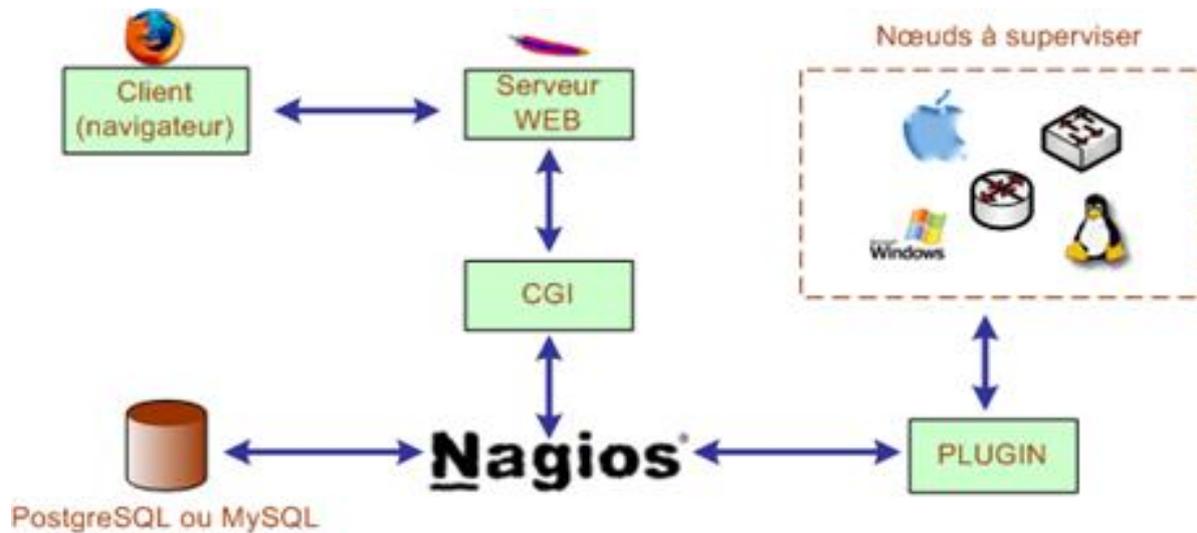


Figure2 : Architecture de Nagios



# CHAPITRE

# 1

# ETUDE PRÉALABLE



## Introduction

Dans ce chapitre, nous présentons le cadre général de notre projet. En effet, nous commençons par la présentation du projet, ensuite la présentation de l'organisme d'accueil, l'étude de l'existence de notre projet et le cycle de vie suivie.

Il s'agit enfin de réaliser une étude sur la méthodologie de conception.

### I. Présentation du projet :

Les systèmes d'information sont tous différents de par leurs tailles, leurs natures, leurs criticités. Ils ont cependant pour point commun d'être le théâtre de l'incident, à un moment où à un autre. Ils sont beau être gérés par les meilleurs administrateurs du globe et être à jour.

La loi de Murphy est immuable. Si quelque chose peut mal tourner, alors elle finira infailliblement par mal tourner. Un des rôles des administrateurs est justement de gérer cela. Ils doivent concevoir l'architecture du système d'information de telle manière qu'une panne ait un impact minimal sur le reste du système. Ils doivent aussi gérer les éventuels problèmes – ce qui reste une part importante de leur charge de travail.

Ainsi un tableau de bord complet, convivial servira comme un outil de la supervision et déclenche une intervention immédiate des administrateurs.

### Objectifs :

Notre projet consiste à Etudier et concevoir réaliser un tableau de bord pour la supervision réseau à l'aide d'un outil open source "Nagios".

Il s'agit de développer une application web qui permet la supervision réseau. Cette interface doit être une solution modulaire simple et intuitive possible de façon à ne nécessiter aucun apprentissage particulier.

La maintenance et la mise à jour de cette interface doivent être faciles dès qu'on possède les fichiers sources.



## II. Présentation de l'organisme d'accueil

L'Institut national de la météorologie (INM) est le service national de météorologie et de sismologie de la Tunisie. Fondé en 1974, mais héritier d'un réseau établi depuis le XIX siècle, il maintient des stations d'observations des éléments du temps, des tremblements de terre et des observations astronomiques, en collaboration avec les organismes internationaux. Il émet les prévisions météorologiques publiques, maritimes et aéronautiques pour le pays et supporte des groupes de recherche en météorologie et géophysique.

**Organisation :** L'INM s'occupe de deux domaines principaux, la météorologie et la géophysique, mais donne un service accessoire en astronomie. Il possède son quartier général à Tunis et six subdivisions régionales : Tunis, Jendouba, Sousse, Sfax, Tozeur et Médenine. Ces subdivisions gèrent le réseau d'observation et donnent les services au public, à l'aviation et aux intérêts maritimes sur leur territoire.

La météorologie est le domaine principal d'activité ; elle regroupe l'observation, la climatologie et la prévision. L'INM maintient 26 stations météorologiques classiques prenant des données horaires, 31 stations d'agro météorologie, 58 stations climatologiques et 208 de pluviométrie prenant des données quotidiennes, huit stations maritimes ainsi qu'un radar météorologique et des stations de radiosondage. Finalement, il existe une station de mesure des polluants. L'INM émet des prévisions météorologiques sous forme de bulletins, directives ou dossiers via différents supports de communication aux différents usagers.

Elles sont établies pour une période allant jusqu'à sept jours. Pour ses activités en sismologie, l'INM dispose de quinze stations de mesure. Sa division d'astronomie assure la prise de mesure du rayonnement solaire l'observation de la phase lunaire à l'échelle du pays pour les besoins religieux l'établissement des dates des éclipses lunaires et solaires et l'élaboration des éphémérides.

### **Missions:**

- ✓ L'observation météorologique.
- ✓ L'enregistrement et la localisation des séismes.
- ✓ L'observation astronomique et le calcul des éphémérides.
- ✓ La prévision du temps.



- ✓ L'exécution de recherches fondamentales et appliquées pour le développement des sciences météorologiques et géophysiques.
- ✓ La satisfaction des besoins généraux en données météorologiques et géophysiques intéressant les divers secteurs de l'économie nationale.
- ✓ La préparation et l'application des accords internationaux dans les domaines relevant de sa compétence et la coopération technique avec les services étrangers et organisations spécialisées.

### **III. Cahier de charge**

Le cahier des charges se définit comme un acte, un document de référence qui permet à un dirigeant d'entreprise de préciser les conditions, les règles et les exigences d'une mission, d'une intervention, d'un travail à accomplir ou d'une tâche à exécuter par un consultant en management, en vue de résoudre un problème spécifique ou d'améliorer une situation donnée, tout en déterminant les résultats attendus.

### **IV. Objectifs**

Le cahier des charges est une demande de service détaillé, élaborée de façon à protéger les intérêts du dirigeant d'entreprise et à améliorer la qualité de l'offre présentée par le consultant.

Il permet plus particulièrement :

- ❖ de définir précisément le problème ou la situation à améliorer;
- ❖ d'identifier clairement les besoins de l'entreprise;
- ❖ de préciser les objectifs de l'intervention et les résultats attendus;
- ❖ d'établir les règles et les exigences de base pour la réalisation de l'intervention;
- ❖ d'obliger la société-conseil et ses conseillers à préciser les services à fournir et les biens livrables;
- ❖ de présenter les éléments d'une entente pouvant lier les deux parties;
- ❖ de servir de document de référence pour éviter les problèmes d'interprétation et les outils;
- ❖ de définir les responsabilités et les rôles respectifs des parties engagées dans l'intervention;
- ❖ de déterminer les modalités de règlement en cas de litiges.



## V. Etude de l'existence

Cette phase consiste à analyser l'existant afin de fournir le plan de développement du projet. Elle permet d'assurer un balayage global des besoins.

### 1. Critique de l'existence

Pour simplifier la mission de la supervision et pour avoir une interface claire, simple et utilisable sur n'importe quel système d'exploitation sans avoir besoin d'installer des plugins sur la machine à superviser.

### 2. Solution

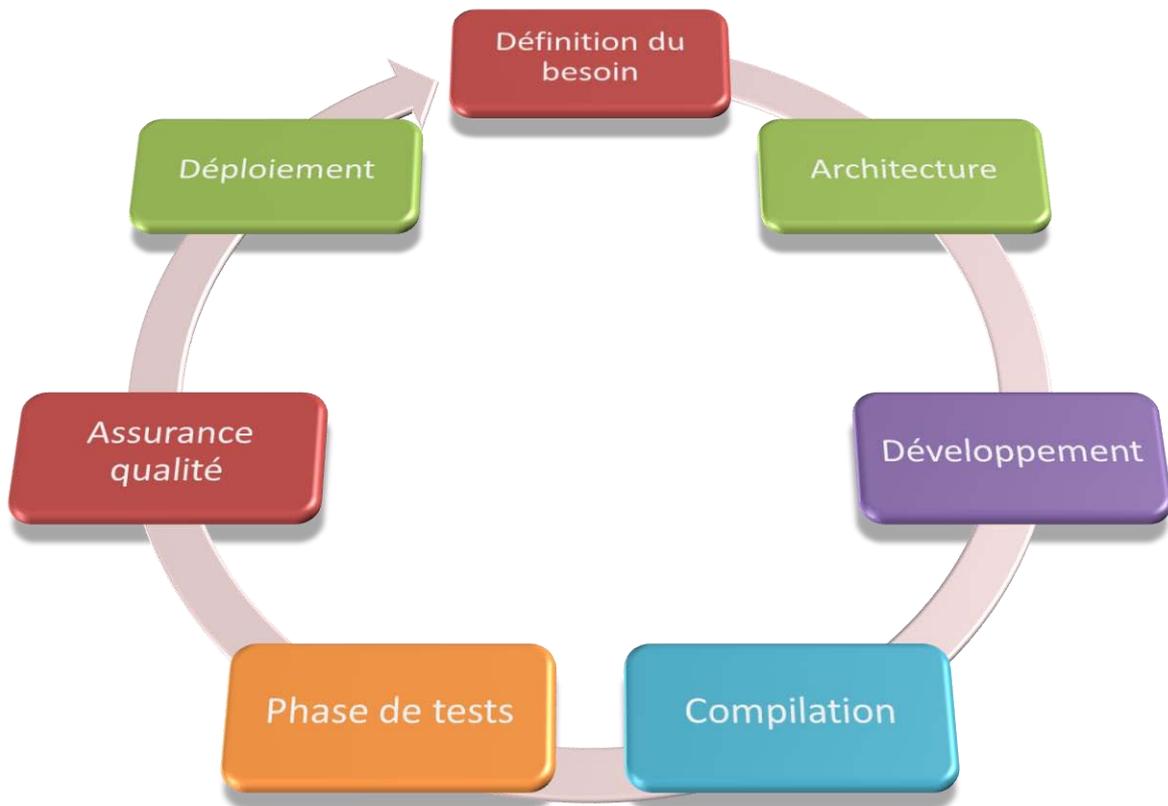
En tenant compte des critiques du système existant et des besoins de l'institut national de météorologie et en se basant sur Nagios, le but de notre travail est de réaliser un tableau de bord de supervision claire et simple à utiliser sous n'importe quel système d'exploitation.

## VI. Cycle de vie

Pour chaque produit (ou application) conçu et développé on choisit une démarche pour le suivre tout au long du projet que l'on appelle cycle de vie du produit. Le cycle de vie se base sur la notion de développement durable en fournissant un moyen efficace pour évaluer les impacts environnementaux d'un produit.

Le concept de cycle de vie est un concept extrêmement important dans le cadre de l'industrialisation de solutions logicielles en général.

Il faut d'abord étudier les systèmes impliqués par les produits à comparer, les flux de matières, après les impacts environnementaux connus, pour chaque étape de cycle de vie.



**Figure 3: Cycle de vie**



## VII. Méthodologie de conception

La conception d'un système d'information est une étape cruciale et essentielle pour le lancement d'un projet informatique, elle s'effectue préalablement à la réalisation et nécessite une méthodologie de travail adéquate répondant aux spécifications de l'application et aux exigences des utilisateurs dans l'objectif d'aboutir à la solution adéquate.

La phase de conception permet de décrire le fonctionnement du système à venir et de formaliser les étapes préparatoires de son développement, en vue d'en faciliter la réalisation et d'assurer la conformité aux besoins de l'utilisateur.

La phase de modélisation nécessite des méthodes d'analyse et de conception permettant de mettre en place une représentation virtuelle d'une réalité.

### 1. Processus unifié

Le processus unifié plus couramment appelé (UP : Unified Process), est une méthode générique de développement logiciel s'appuyant sur le langage de modélisation UML (Unified Modeling Language).

Le principal intérêt du processus unifié est de limiter les risques lors du développement de projets.

Les caractéristiques du Processus unifié :

- **Centré sur l'architecture** : l'architecture peut être considérée comme l'ensemble de vues du système qui vont provenir des besoins de l'entreprise et des différents intervenants.
- **Piloté par les cas d'utilisation** : le modèle des cas d'utilisation guide le processus unifié et décrit les fonctionnalités du système.
- **Itératif et incrémental** : les itérations se succèdent dans un ordre logique permettant de donner lieu à un incrément et donc d'établir un développement le plus optimisé et efficace.

Le cycle de vie du processus unifié comprend 4 phases :

- Incubation/ création
- Elaboration



- Construction
- Transition

## 2. UML

### 2.1 Définition

UML n'est pas une méthode (C'est une description normative des étapes de la modélisation) : ses auteurs ont en effet estimé qu'il n'était pas opportun de définir une méthode en raison de la diversité des cas particuliers. Ils ont préféré se borner à définir un langage graphique qui permet de représenter et de communiquer les divers aspects d'un système d'information aux graphiques qui sont bien sûr associés des textes qui expliquent leur contenu. UML est donc un métalangage car il fournit les éléments permettant de construire le modèle qui, lui, sera le langage du projet.

### 2.2 Les avantages de l'UML

UML est un langage formel et normalisé. Il permet ainsi : Un gain de précision, un gage de stabilité et l'utilisation d'outils.

UML est un support de communication performant : Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse lui font un langage universel.

#### Objectifs :

- Construire des modèles de systèmes,
- Organiser le travail,
- Gérer le cycle de vie de A à Z,
- Gérer le risque,
- Obtenir de manière répétitive des produits de qualité constante.

En conclusion, nous avons choisi de travailler avec UML parce qu'il exprime mieux la vue statique et dynamique du système d'information et pour notre application web, il est nécessaire de faire une analyse très approfondie pour pouvoir dégager les nécessités de développement ainsi que quelques scénarios d'exécution.



Maintenant dans une conception d'une application web, les diagrammes de classes obtenus doivent être encore modifiés selon la technologie (.net, java/j2ee) ou langage (PHP) utilisés ce qui rend le passage à la génération de code moins évidente...

### **3. Logiciel de modélisation**

StartUML est un logiciel de modélisation UML/MDA rapide, flexible, disponible en open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU CPL.

L'objectif du projet StartUML est de construire un outil de modélisation de logiciels et aussi la plate-forme qui est un remplacement convaincante d'outils UML commerciaux tels que Rational Rose, Together et ainsi de suite.

#### **Conclusion :**

A travers ce 1<sup>er</sup> chapitre, nous avons pu se placer dans le contexte général de notre projet en introduisant le domaine de l'application ainsi que le cadre du travail.

La détermination des besoins fonctionnels et non fonctionnels va nous permettre de mieux clarifier notre application, de bien mettre au point nos objectifs au cours de la phase suivante, qui est la phase d'incubation.



# CHAPITRE

# 2

# PHASE D'INCUBATION



## Introduction

La phase d'incubation consiste à comprendre le contexte du système. Dans cette phase, nous allons déterminer les besoin fonctionnels et non fonctionnels de l'application et d'identifier les cas d'utilisation initiaux.

### I. Capture des besoins

La phase de capture des besoins est la première phase formelle obligatoire dans le développement d'une application informatique puisque la capacité de persuasion d'un produit ne peut se réaliser parfaitement sans une spécification préalable élaborée des besoins et des exigences

#### 1. Spécification des besoins fonctionnels

Les besoins fonctionnels servent à présenter les actions que doit effectuer le système en réponse à une demande présentée par un utilisateur.

- ❖ Contrôler les accès selon les rôles et les droits de chacun des utilisateurs.
- ❖ Gestion des profils des différents acteurs.
- ❖ supervision des systèmes et réseaux.

#### 2. Présentation des acteurs

Nous précédons par une détermination des acteurs et des cas d'utilisation :

Un acteur est toute entité qui interagit avec notre système dans le but de réaliser une plus-value et qui a toujours le même comportement.

Un cas d'utilisation est la description d'un ensemble de séquences d'opérations qu'un système effectue pour répondre au besoin d'un acteur.

Le modèle de diagramme d'utilisation ainsi que les interactions entre les composants du système.

Notre application interagit avec les acteurs suivants :

#### ✓ Superviseur

C'est un agent qui assure la supervision des systèmes et la résolution des alertes.

Le logiciel doit permettre :

- ❖ L'accès et l'identification.



Le superviseur a comme rôle :

- ❖ La consultation des historiques.
- ❖ Afficher la liste des logiciels installés.
- ❖ Consulter la taille mémoire.
- ❖ Gérer la gestion des processus.
- ❖ surveillance des services réseaux.
- ❖ Gérer le système.

### 3. Spécification des besoins non fonctionnels

Les besoins non fonctionnels présentent les exigences internes pour le système et cachées vis à vis des utilisateurs.

Les impératifs les plus immédiats de notre application pris en considération sont :

- ❖ Sur le plan ergonomique, l'application doit être compatible avec n'importe quel système d'exploitation toute en étant facile à manipuler. Ces interfaces doivent être compréhensibles, et bien organisées.
- ❖ **L'extensibilité:** Le système doit permettre l'intégration de nouvelles fonctionnalités.
- ❖ **La sécurité des données:** Sécuriser les données revient à appliquer une stratégie d'identification, d'authentification, d'autorisation et de contrôle de chaque tentative d'accès à ces données. Dans notre système l'accès aux informations personnelles n'est autorisé qu'aux personnes propriétaires et selon un privilège qui détermine les droits d'accès.

## II. Diagramme de cas d'utilisation global du système

Un cas d'utilisation est une manière spécifique d'utiliser un système. C'est l'image d'une fonctionnalité du système, déclenchée en réponse à la stimulation d'un acteur externe.

Les cas d'utilisation apportent une solution aux problèmes de la détermination et de la compréhension des besoins.

Ce diagramme donne à chaque employé la possibilité de vérifier et de superviser. Chaque employé doit s'authentifier avant d'accéder à son groupe.

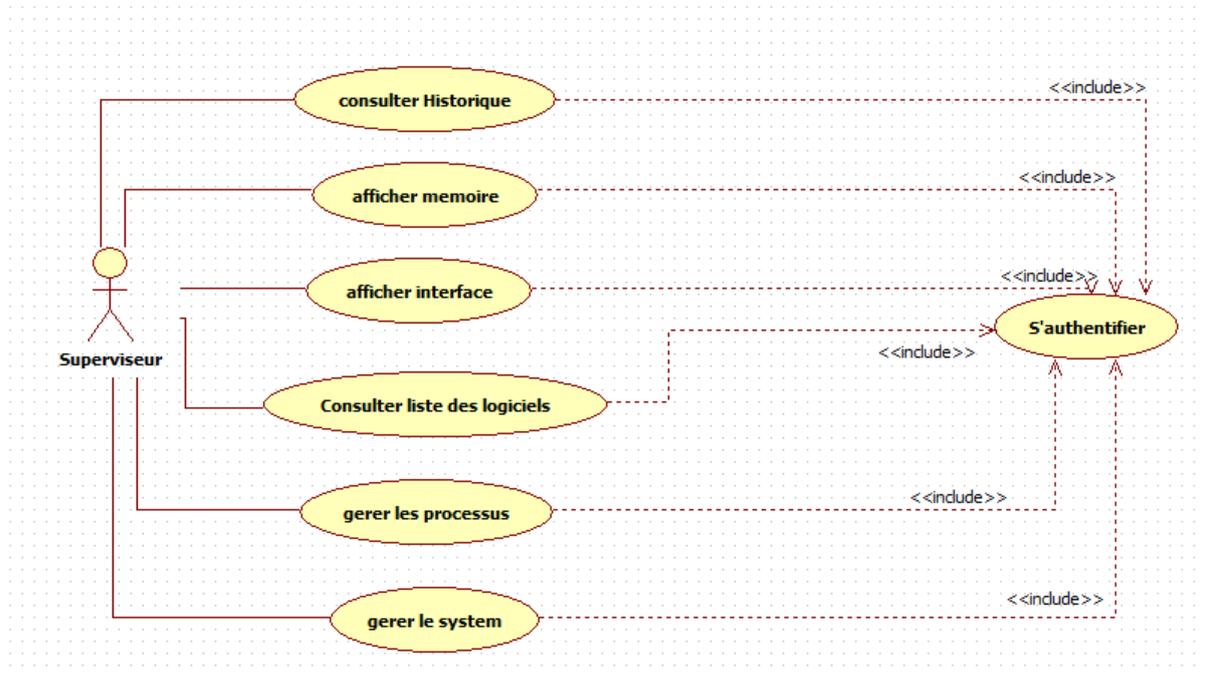


Figure 4 Diagramme de cas d'utilisation global du système

### III. Affectation des priorités

Cas d'utilisation	Acteur	Priorité
S'authentifier	Superviseur	1
Consulter l'historique	Superviseur	2
Afficher la mémoire	Superviseur	2
Consulter la liste des logiciels	Superviseur	2
Gérer les processus	Superviseur	2
Gérer les systèmes	Superviseur	2

### IV. Raffinement de cas d'utilisation de priorité 1

#### 1. Raffinement de cas « s'authentifier »

**Cas d'utilisation** : « s'authentifier ».

**Résumé** : s'identifier.

**Acteur** : Superviseurs.

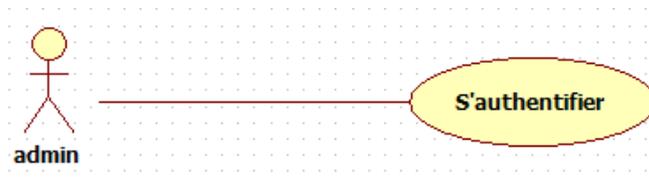
**Pré condition** : besoin d'accès.

**Post condition** : Authentification validée et succès d'accès.

**Scénario principal** :

1. Le système affiche la page de connexion ;
2. L'utilisateur saisit son identifiant et son mot de passe ;
3. Le système vérifie la validité des informations fournies ;
4. Le système donne l'accès à l'utilisateur, le menu principal s'affiche ;

**Exception** : Si les champs sont invalides, il y aura une demande de vérification des champs.



**Figure 5 : Description de cas d'utilisation « s'authentifie »**

## V. Modèle d'analyse des cas d'utilisation de priorité 1

### 1. Modèle d'analyse de cas d'utilisation « s'authentifier »

- Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « s'authentifier » :

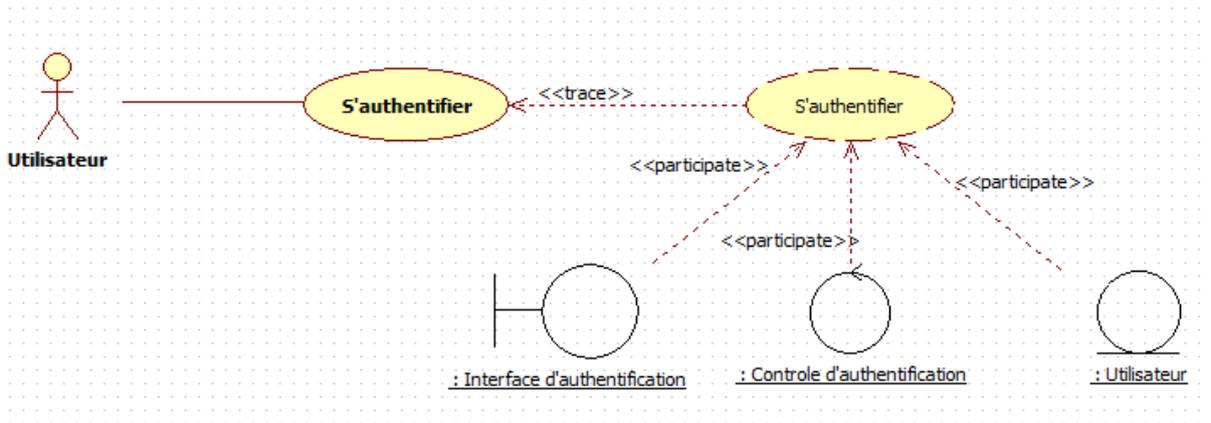


Figure 6: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse de cas « s'authentifier »

- Diagramme de classe d'analyse de cas d'utilisation « S'authentifier » :

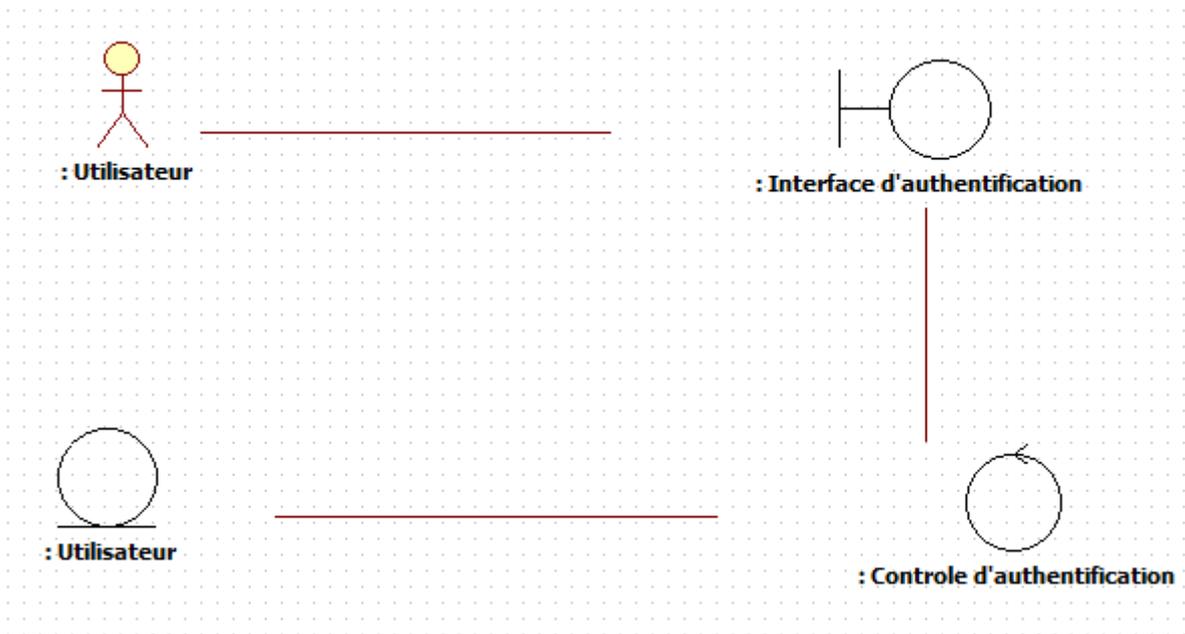


Figure 7: Diagramme de classe d'analyse de cas d'utilisation « S'authentifier »

- Diagramme de collaboration de cas d'utilisation « S'authentifier » :

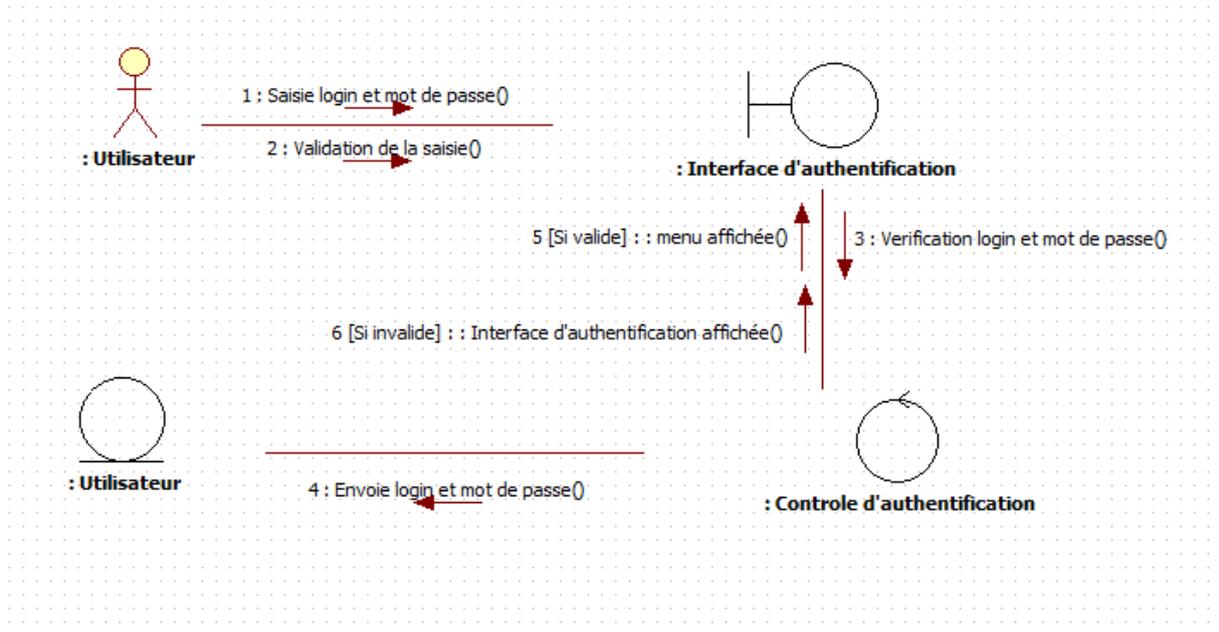
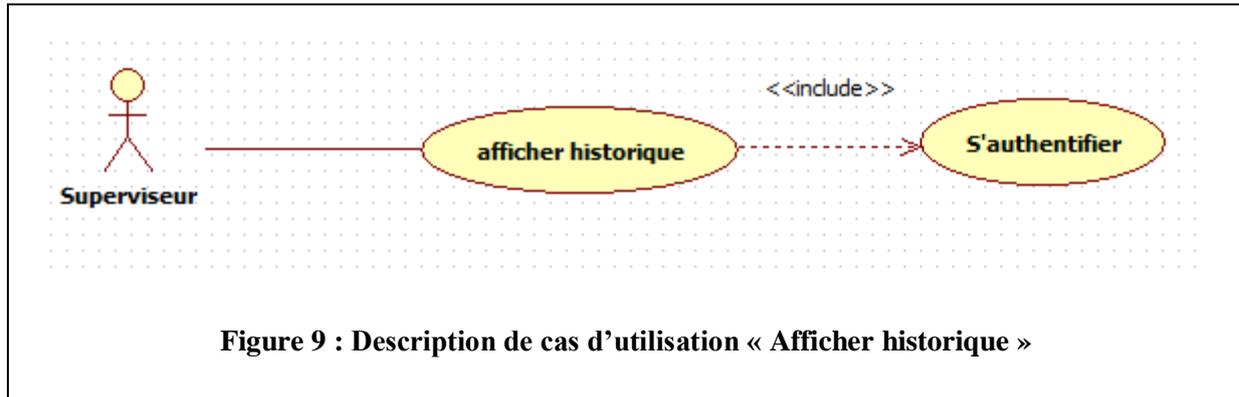


Figure 8: Diagramme de collaboration de cas d'utilisation « S'authentifier »

## VI. Raffinement de cas d'utilisation de priorité 2

### 1. Raffinement de cas « Consulter historique »

<b>Cas d'utilisation :</b> « Consulter Historique».
<b>Résumé :</b> Consulter l'historique de supervision.
<b>Acteur :</b> Superviseurs.
<b>Pré condition :</b> Visualisation de l'historique de consultation.
<b>Post condition :</b> Historique de consultation affiché.
<b>Scénario principal :</b> <ol style="list-style-type: none"> <li>1. Le superviseur choisit le menu « consulter historique » ;</li> <li>2. Le système affiche l'historique de consultation ;</li> </ol>



## 2. Raffinement de cas « Afficher mémoire »

<b>Cas d'utilisation :</b> « Afficher mémoire ».
<b>Résumé :</b> Afficher les détails mémoire.
<b>Acteur :</b> Superviseurs.
<b>Pré condition :</b> afficher les détails mémoire du disque de la machine à superviser.
<b>Post condition :</b> détails mémoire affichée.
<b>Scénario principal :</b> <ol style="list-style-type: none"> <li>1. Le superviseur choisit le menu « Afficher mémoire » ;</li> <li>2. Le superviseur choisit la machine à superviser ;</li> <li>3. Le système affiche les détails mémoire. ;</li> </ol>
<pre>             graph LR               S((Superviseur)) --- UC1(afficher memoire)               UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2(S'authentifier)           </pre> <p><b>Figure 10 : Description de cas d'utilisation « Afficher mémoire»</b></p>

## 3. Raffinement de cas « Afficher interfaces »

<b>Cas d'utilisation</b> : « Afficher interfaces».
<b>Résumé</b> : Afficher les ports et les caractéristiques de la carte réseau.
<b>Acteur</b> : Superviseurs.
<b>Pré condition</b> : Afficher les caractéristiques de la carte réseau et des ports.
<b>Post condition</b> : détails de la carte réseau affichés.
<p><b>Scénario principal</b> :</p> <ol style="list-style-type: none"> <li>1. Le superviseur choisit le menu « Afficher interfaces» ;</li> <li>2. Le superviseur choisit la machine à superviser ;</li> <li>3. Le système affiche les détails de la carte réseau ;</li> </ol>
<pre> graph LR     Actor[Superviseur] --- UC1(afficher interface)     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2(S'authentifier)         </pre>
<b>Figure 11 : Description de cas d'utilisation « Afficher interface »</b>

#### 4. Raffinement de cas « Consulter liste des logiciels »

<b>Cas d'utilisation</b> : « Consulter liste des logiciels ».
<b>Résumé</b> : Afficher la liste des logiciels qui sont installé dans la machine à superviser.
<b>Acteur</b> : Superviseurs.
<b>Pré condition</b> : Afficher la liste des logiciels qui sont installés.
<b>Post condition</b> : liste des logiciels affichée.
<b>Scénario principal</b> :

1. Le superviseur choisit le menu « Consulter liste des logiciels» ;
2. Le superviseur choisit la machine à superviser ;
3. Le système affiche la liste des logiciels installés ;

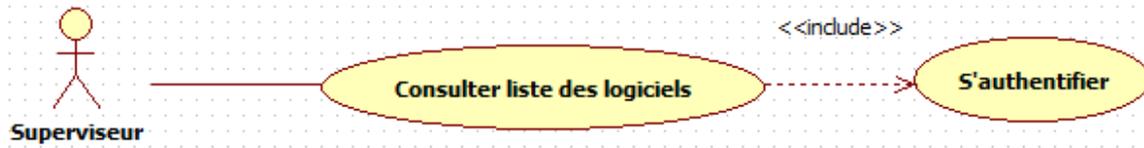


Figure 12 : Description de cas d'utilisation « Consulter liste des logiciels»

### 5. Raffinement de cas « Gérer les processus»

**Cas d'utilisation :** « Gérer les processus».

**Résumé :** Afficher les détails sur la gestion des processus de la machine à superviser.

**Acteur :** Superviseurs.

**Pré condition :** Afficher les détails de la gestion des processus.

**Post condition :** détails de la gestion des processus affichés.

**Scénario principal :**

1. Le superviseur choisit le menu «Gérer les processus » ;
2. Le superviseur choisit la machine à superviser :
3. Le système affiche les détails de la gestion des processus ;

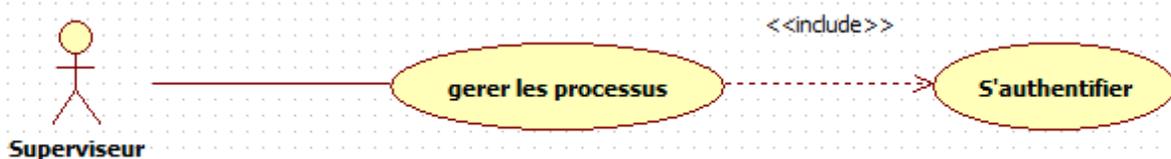


Figure 13 : Description de cas d'utilisation « Gérer les processus »

## 6. Raffinement de cas « Gérer les systèmes »

**Cas d'utilisation :** « Gérer les systèmes».

**Résumé :** Afficher les détails du système installé sur la machine à superviser.

**Acteur :** Superviseurs.

**Pré condition :** Afficher les détails du système installé.

**Post condition :** Détails système affichés.

**Scénario principal :**

1. Le superviseur choisit le menu «Gérer les systèmes» ;
2. Le superviseur choisit la machine à superviser ;
3. Le système affiche les détails du système de la machine à superviser ;

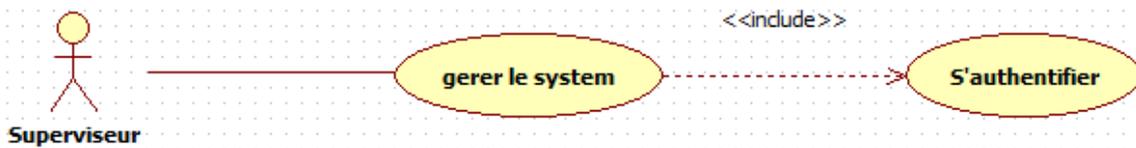
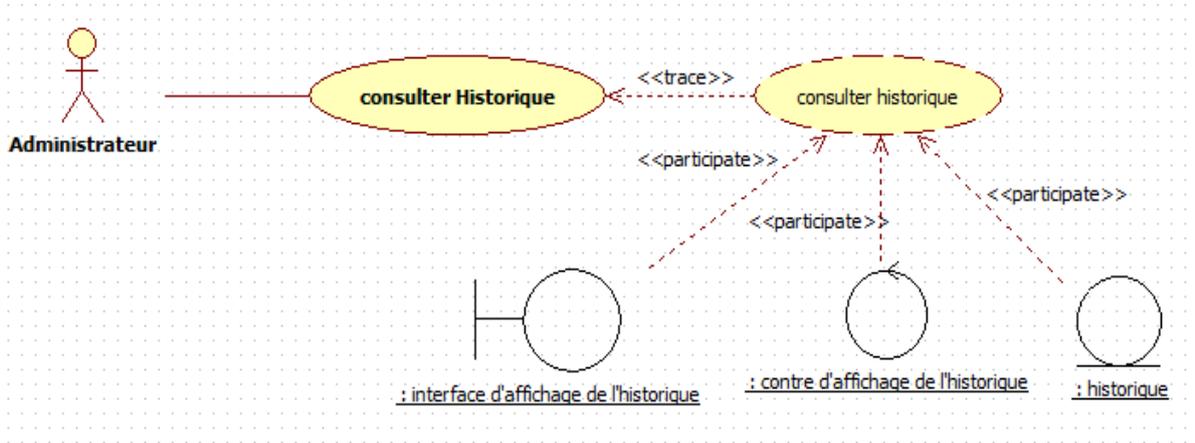


Figure 14 : Description de cas d'utilisation « Gérer le système»

## VII. Modèle d'analyse des cas d'utilisation de priorité 2

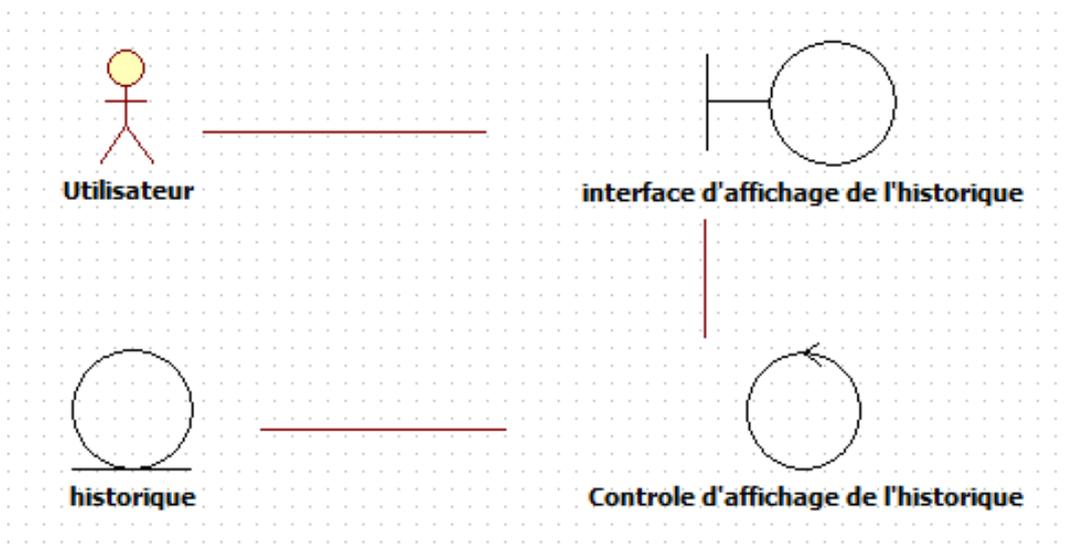
### 1. Modèle d'analyse de cas d'utilisation « Consulter historique »

- Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Consulter historique »



**Figure 15: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Consulter historique »**

- Diagramme de classe d'analyse de cas d'utilisation « Consulter historique»



**Figure 16: Diagramme de classe d'analyse de cas d'utilisation « Consulter historique»**

- Diagramme de collaboration de cas d'utilisation « Consulter historique»

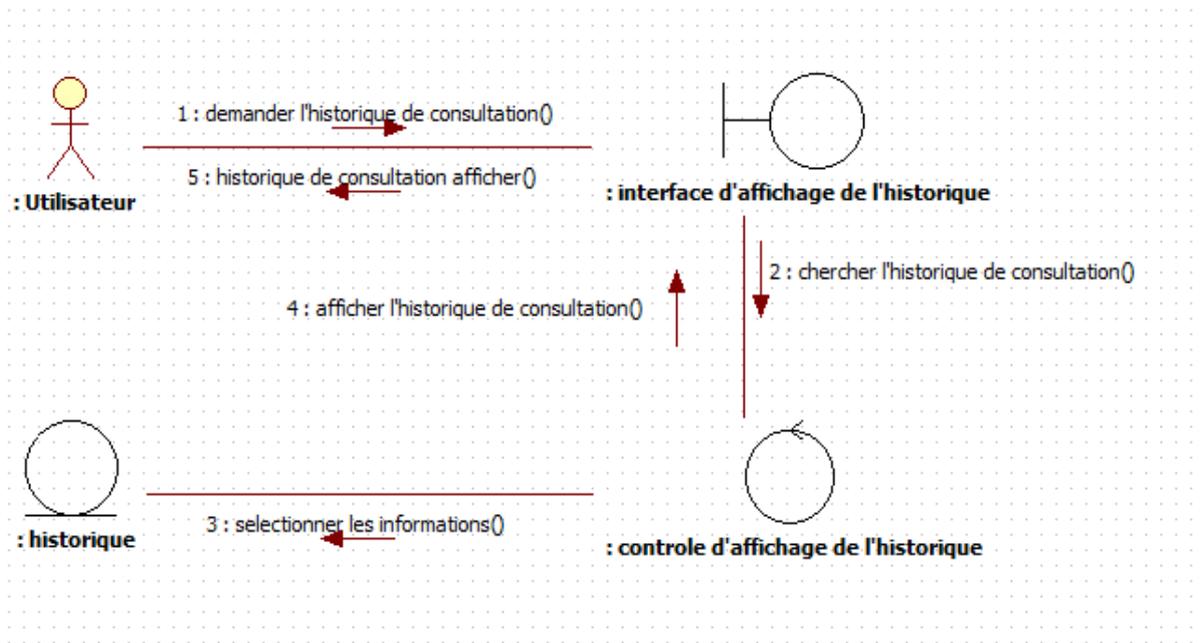


Figure 17: Diagramme de collaboration de cas d'utilisation « Consulter historique»

## 2. Modèle d'analyse de cas d'utilisation « Afficher mémoire»

- Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Afficher mémoire»

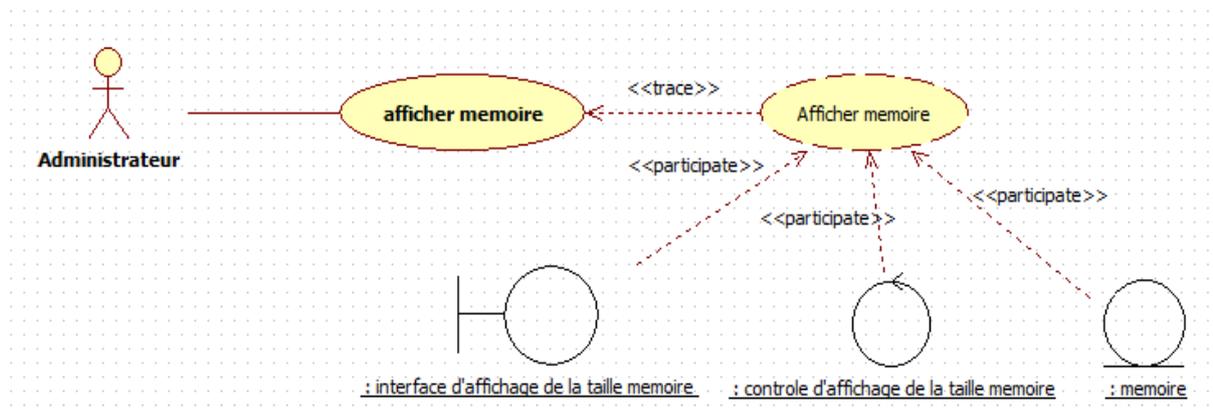


Figure 18: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Afficher mémoire »

- Diagramme de classe d'analyse de cas d'utilisation « Afficher mémoire»

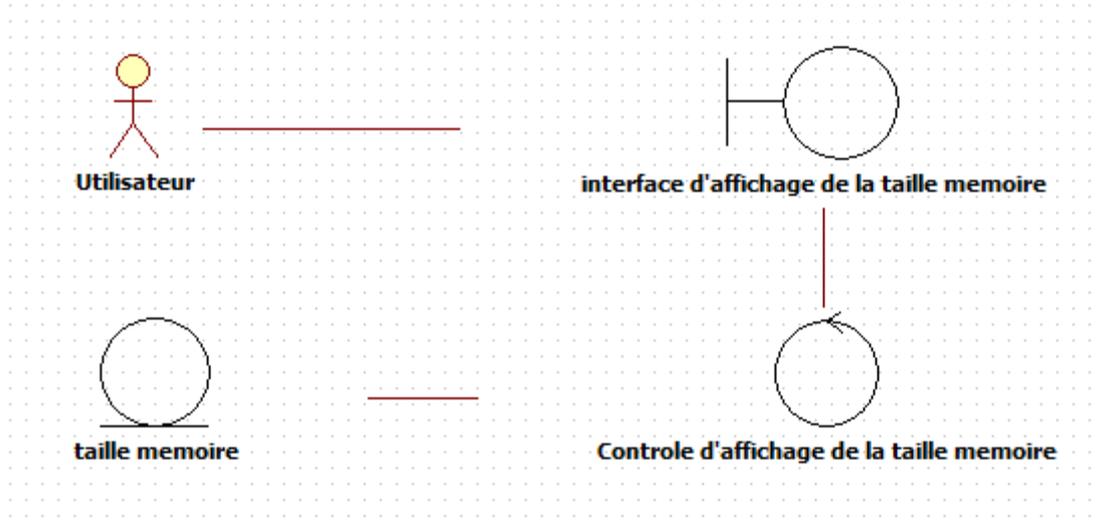


Figure 19: Diagramme de classe d'analyse de cas d'utilisation « Afficher mémoire»

- Diagramme de collaboration de cas d'utilisation « Afficher mémoire»

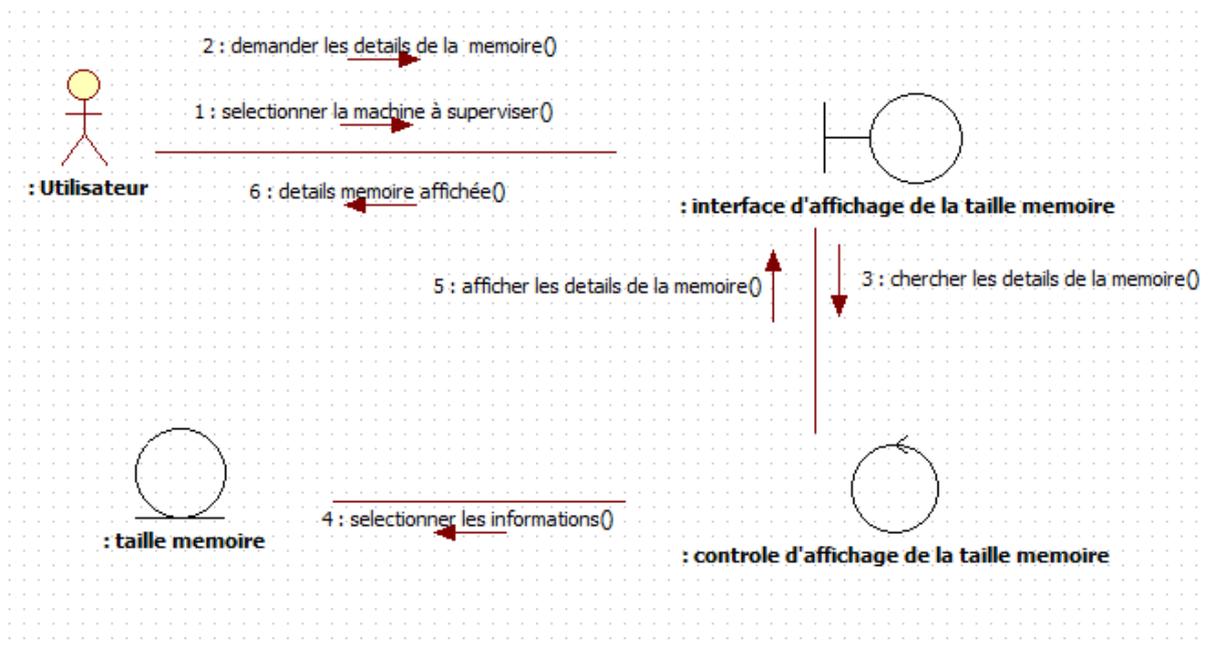
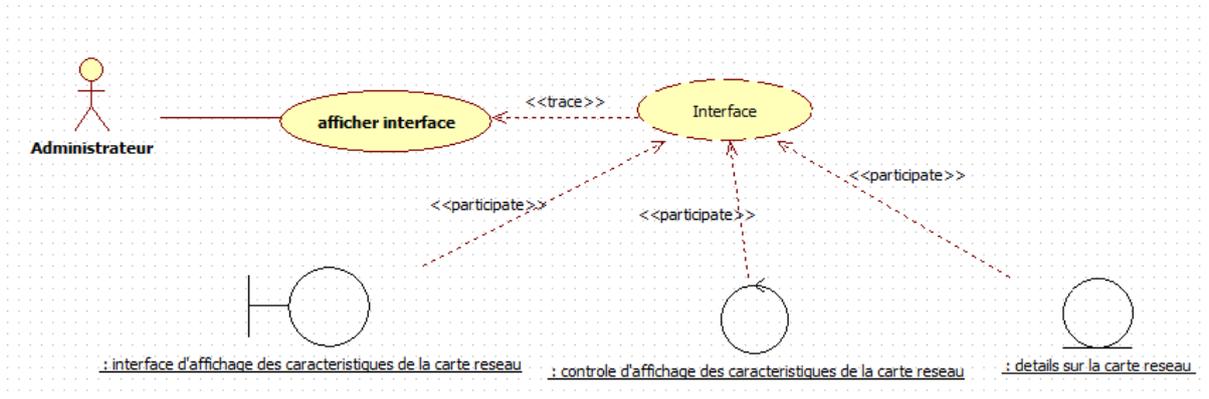


Figure 20: Diagramme de collaboration de cas d'utilisation « Afficher mémoire»

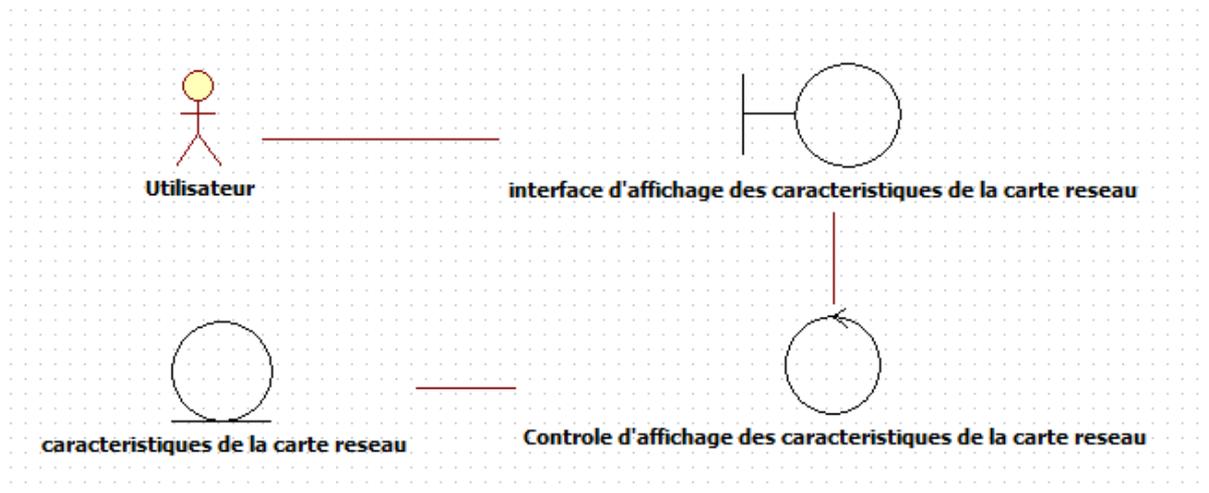
### 3. Modèle d'analyse de cas d'utilisation « Afficher interfaces»

- Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Afficher interfaces »



**Figure 21: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Afficher interfaces »**

- Diagramme de classe d'analyse de cas d'utilisation « Afficher interfaces»



**Figure 22: Diagramme de classe d'analyse de cas d'utilisation « Afficher interfaces»**

- Diagramme de collaboration de cas d'utilisation « Afficher interfaces»

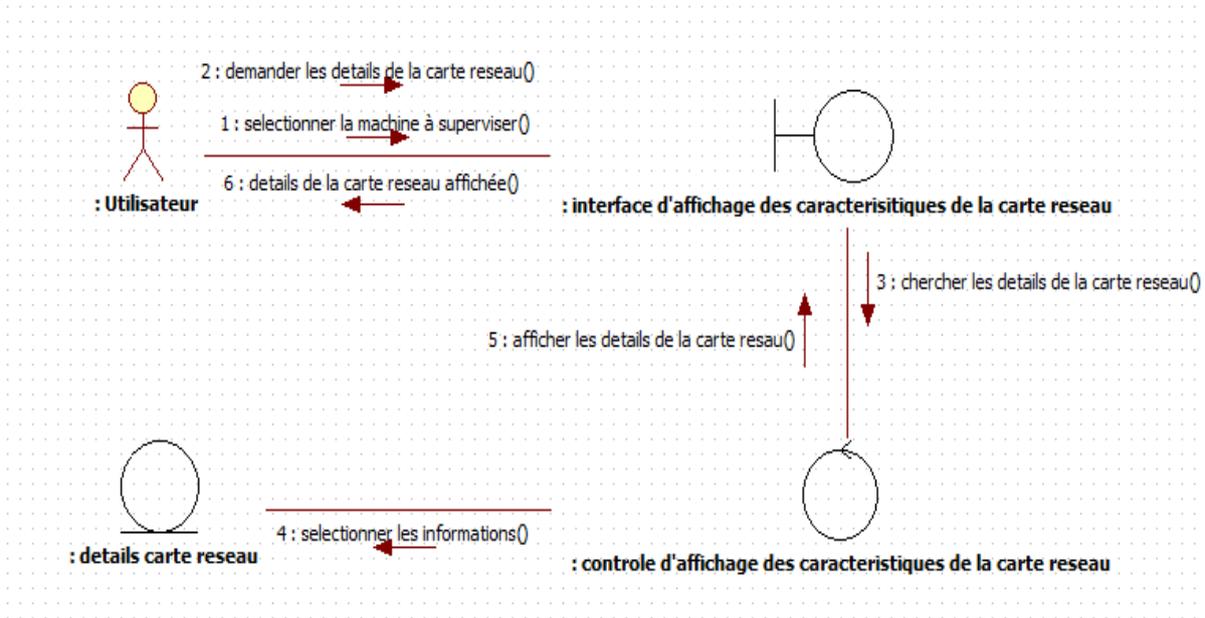


Figure 23: Diagramme de collaboration de cas d'utilisation « Afficher interfaces»

#### 4. Modèle d'analyse de cas d'utilisation « Afficher processus»

- Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Afficher processus »

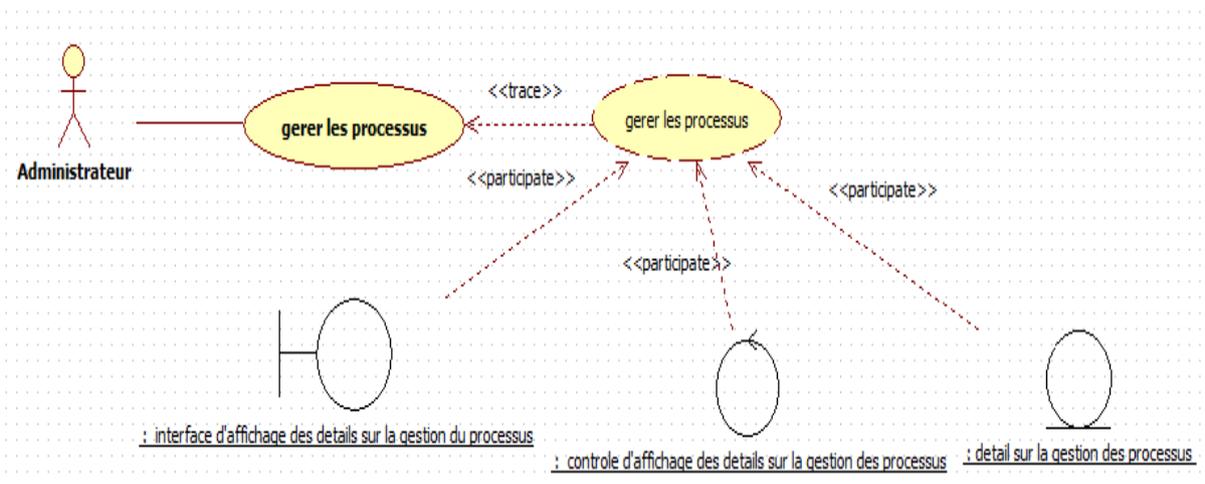


Figure 24: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Afficher processus »

- Diagramme de classe d'analyse de cas d'utilisation « Afficher processus»

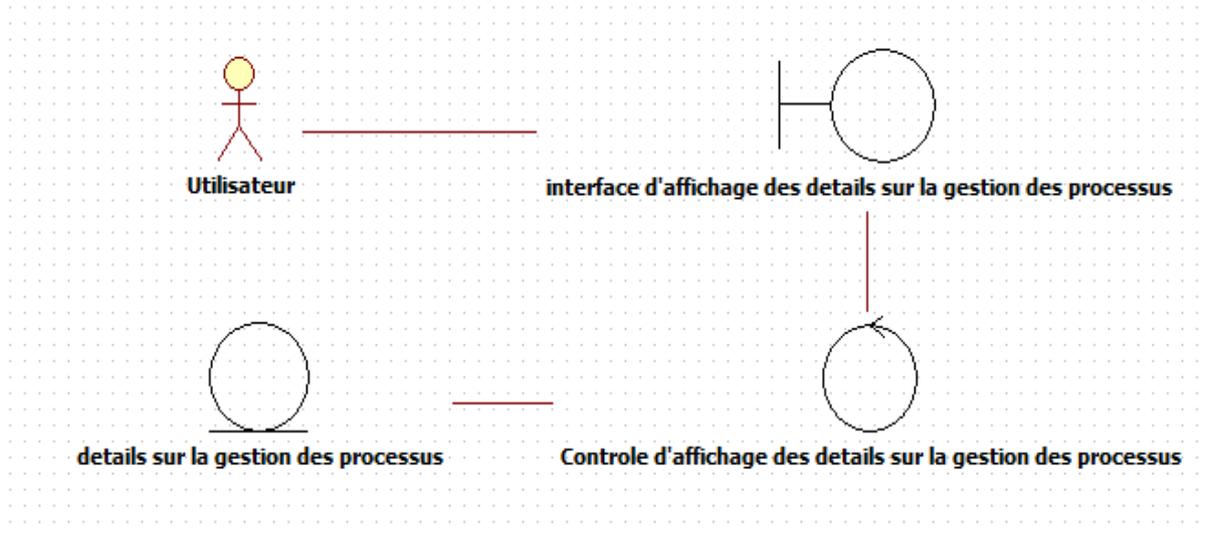


Figure 25: Diagramme de classe d'analyse de cas d'utilisation « Afficher processus»

- Diagramme de collaboration de cas d'utilisation « Afficher processus»

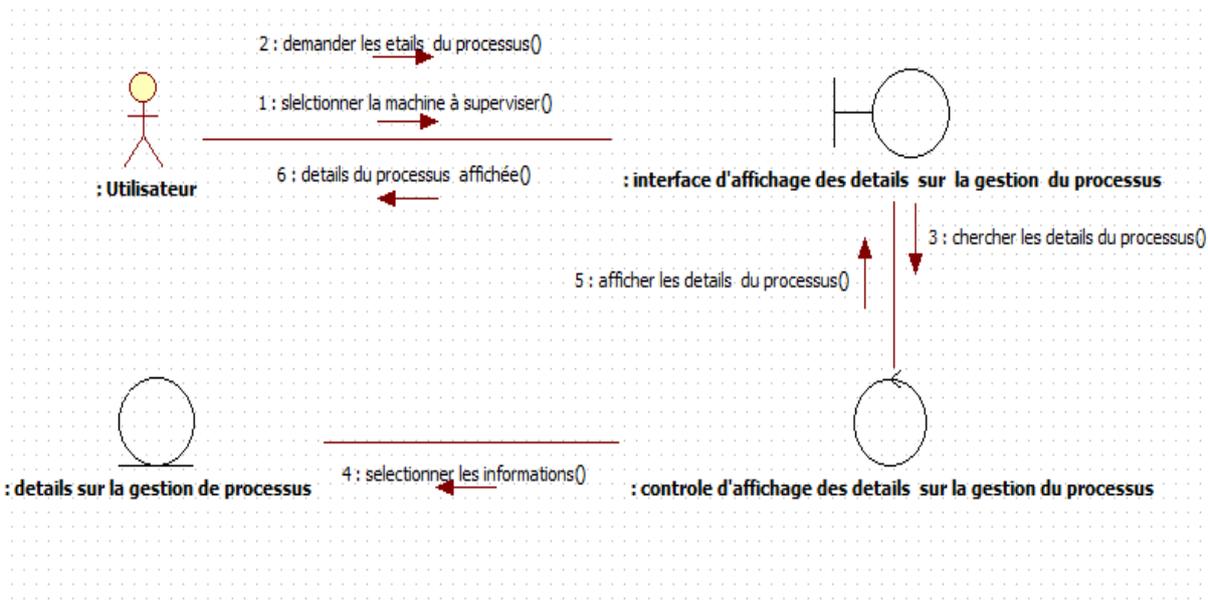
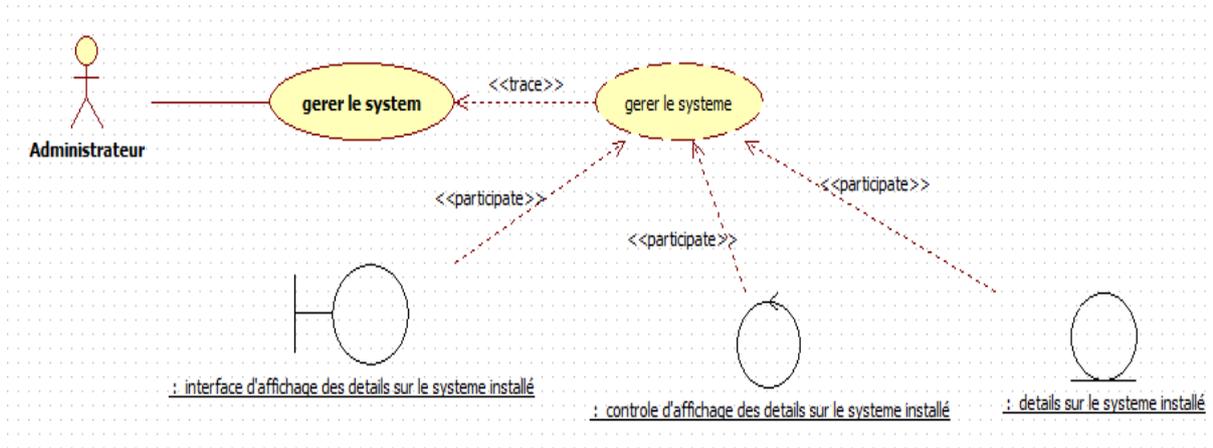


Figure 26: Diagramme de collaboration de cas d'utilisation « Afficher processus»

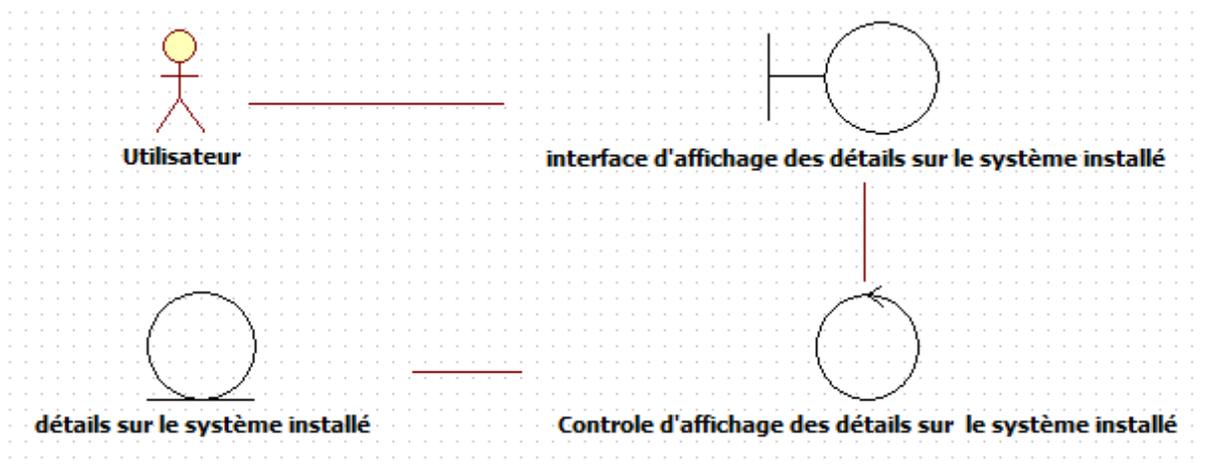
### 5. Modèle d'analyse de cas d'utilisation « Gérer le système»

- Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Gérer le système »



**Figure 27: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Gérer le système»**

- Diagramme de classe d'analyse de cas d'utilisation « Gérer le système»



**Figure 28: Diagramme de classe d'analyse de cas d'utilisation « Gérer le système»**

- Diagramme de collaboration de cas d'utilisation « Gérer le système»

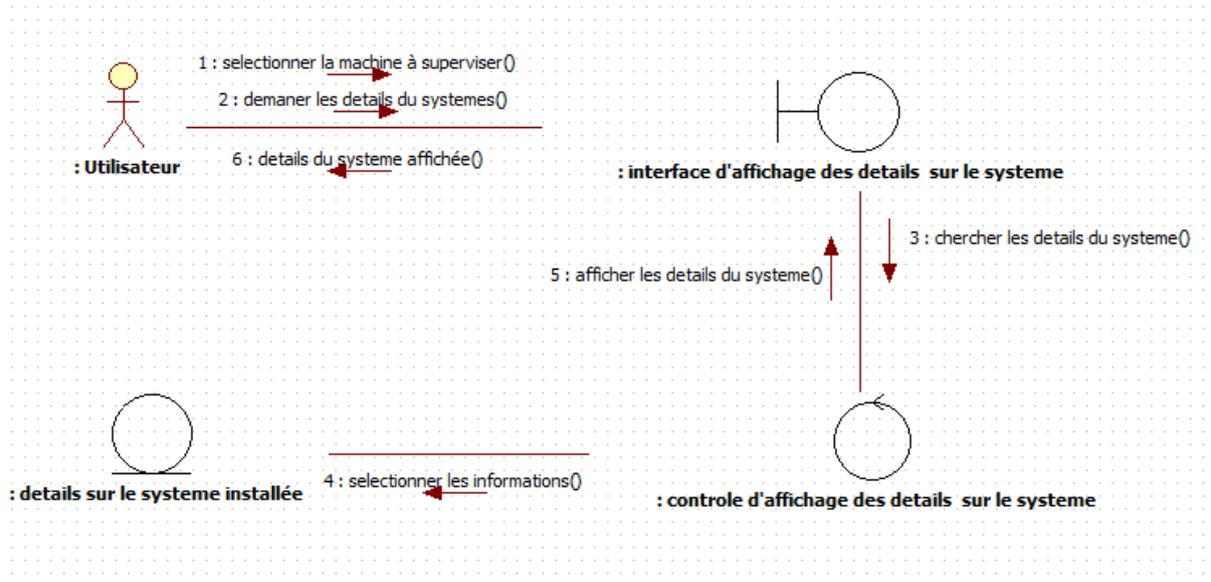


Figure 29: Diagramme de collaboration de cas d'utilisation « Gérer le système»

### 6. Modèle d'analyse de cas d'utilisation « Logiciels»

- Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Logiciels»

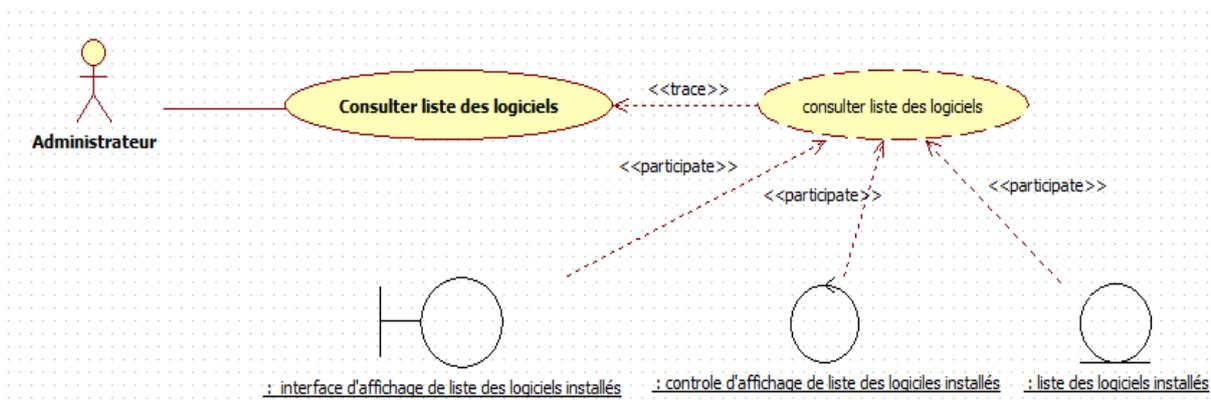


Figure 30: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Logiciels»

- Diagramme de classe d'analyse de cas d'utilisation « Logiciels»

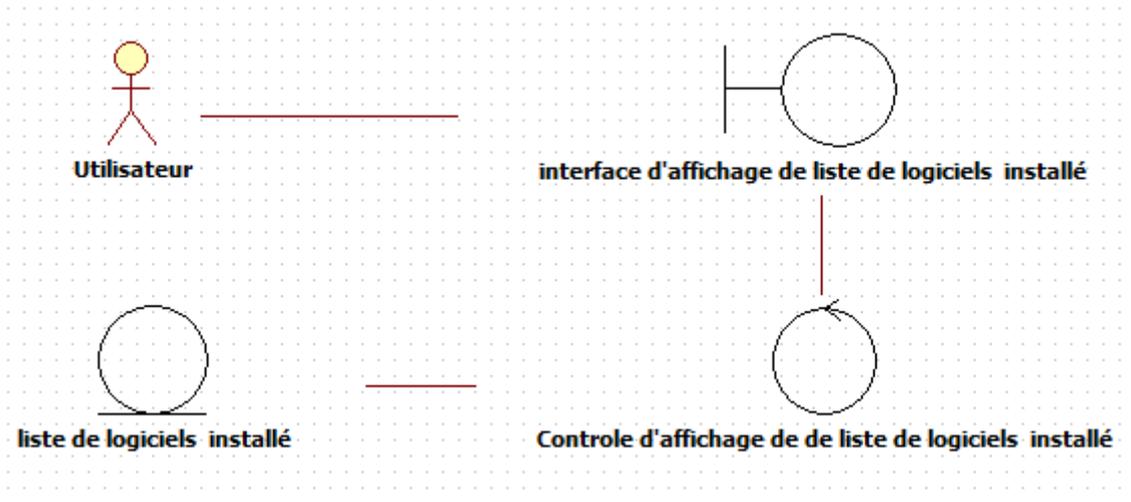


Figure 31: Diagramme de classe d'analyse de cas d'utilisation « Logiciels»

- Diagramme de collaboration de cas d'utilisation « Logiciels»

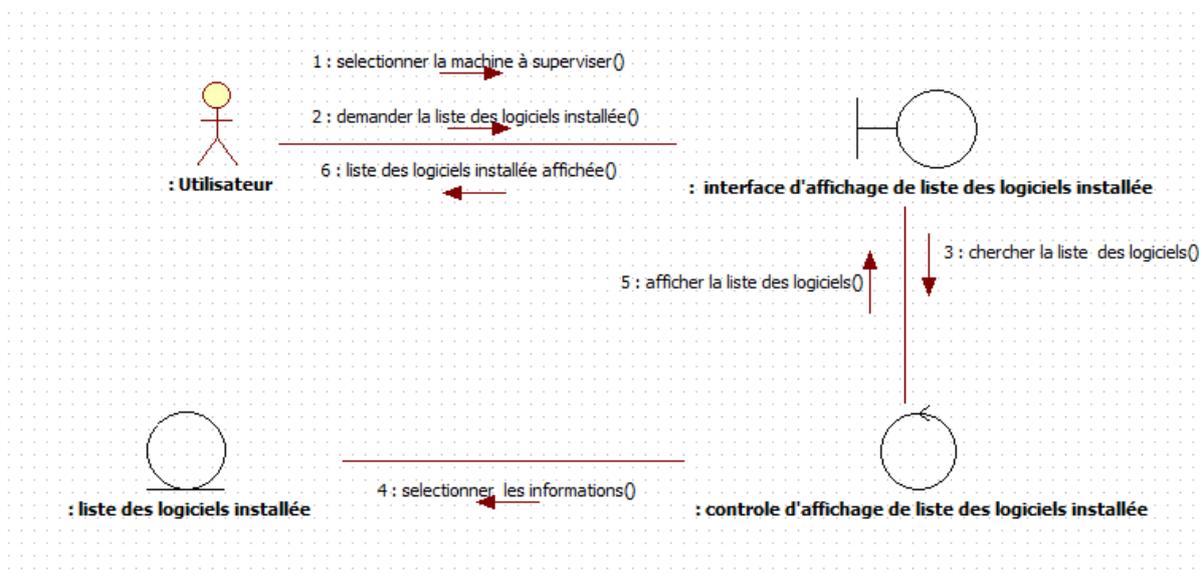


Figure 32: Diagramme de collaboration de cas d'utilisation « Logiciels»

### Conclusion

Dans ce chapitre nous avons pu identifier les différentes fonctionnalités de notre application. Nous avons construit le modèle des exigences du système et défini les cas d'utilisation et planifié la suite d'itération qui va permettre de les implanter.

Nous avons analysé, effectué une conception des différents cas d'utilisation, précisé la plupart des cas d'utilisation et nous avons pu concevoir l'architecture du système qui doit être exprimé sous forme de vues de tous les modèles du système.



# CHAPITRE

# 3

# CONCEPTION

## Introduction

De nombreuses applications fonctionnent selon un environnement client/serveur, parmi ces applications se situe notre application "Tableau de bord de supervision". Il s'agit donc, dans ce qui suit, d'en fixer l'architecture tout en se référant aux architectures et aux modèles précédemment cités. Ainsi ce chapitre sera dédié pour la conception architecturale de notre site et pour la conception détaillée.

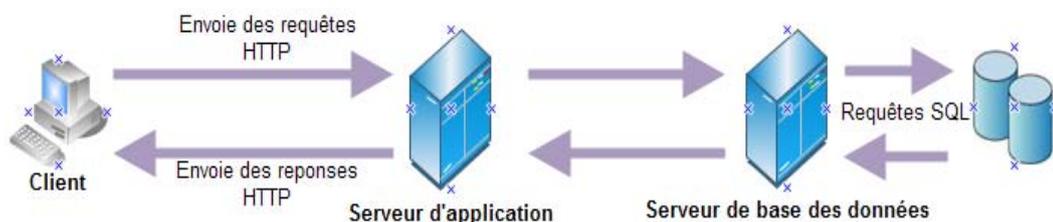
### I. Conception générale

#### 1. Architecture 3-tiers

Notre application, se base sur le principe client-serveur, nous adopterons donc l'architecture 3-tiers. Dans l'architecture à trois niveaux (appelées architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

- ❖ Le client : le demandeur de ressources.
- ❖ Le serveur d'application (appelé aussi middleware) : le serveur chargé de fournir les ressources.
- ❖ Le serveur de base de données : fournissant un service au premier serveur.

Dans l'architecture 3-tiers, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est destiné pour une tâche spécifiée, la figure (**Figure 33**) est un exemple explicatif :



**Figure 33: Architecture 3-tiers**

Les avantages de l'architecture 3-tiers sont principalement au nombre de quatre :



- ✓ Les requêtes clients vers le serveur sont d'une plus grande flexibilité que dans celles de l'architecture 2-tiers basées sur le langage SQL.
- ✓ Cette flexibilité permet à une entreprise d'envisager dans le cadre d'une architecture 3-tiers une grande souplesse pour l'introduction de toutes nouvelles technologies.
- ✓ D'un point de vue développement, la séparation qui existe entre le client, le serveur et le SGBD permet une spécialisation des développeurs sur chaque tiers de l'architecture.
- ✓ Plus de flexibilité dans l'allocation des ressources ; la portabilité du tiers serveur permet d'envisager une allocation et ou modification dynamique aux grés des besoins évolutifs au sein d'une entreprise.

## 2. Modèle en couches

Ce modèle logique d'architecture applicative est constitué de trois couches logicielles séparées. Ainsi l'application est composée de trois couches (étages ou niveaux) complètement indépendantes l'une de l'autre.

En effet le modèle en couches cherche à séparer les couches présentation, traitement et accès aux données. Le développeur a intérêt à maintenir une grande indépendance entre ces trois entités afin que si l'une d'elles change, les deux autres n'aient pas à changer.

- ❖ La représentation des données : contient les clients (navigateurs web ou une simple interface cliente).
- ❖ Le traitement métier des données : correspondant à la mise en œuvre de l'ensemble des règles de gestion de la logique applicative.
- ❖ L'accès aux données persistantes.

La figure suivante (**Figure 34**) représente une application qui respect ce modèle :

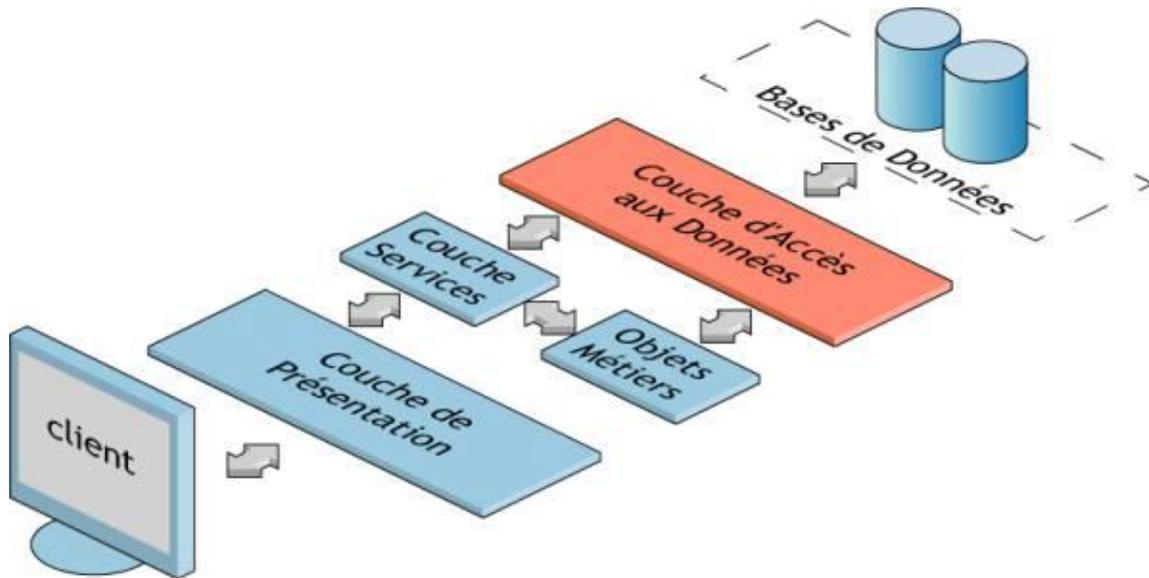


Figure 34: Architecture du modèle en couches

### 3. Modèle MVC (Model, View, Controller)

Notre application repose sur le modèle en couche. Cependant la connexion de ces couches entre-elles n'est pas une mince affaire. En effet, pour rester dans le principe de l'autonomie, il faut que les couches soient indépendantes. C'est donc à ce niveau que les problèmes apparaissent. Donc pour obtenir une telle architecture il est indispensable d'utiliser des modèles de conception qui facilite cette tâche fastidieuse.

Le modèle de conception que nous allons choisir et qui satisfait notre besoin est le MVC  
2 La figure suivante (**Figure 35**) illustre le principe du modèle MVC 2

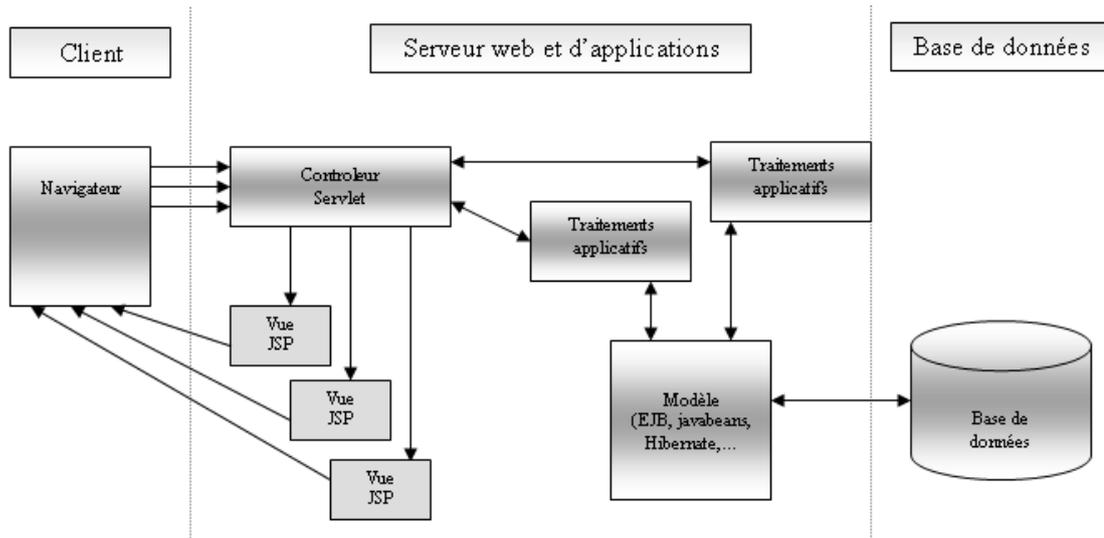


Figure 35: Modèle MVC 2

## II. Conception détaillée

### 1. Couche d'accès aux données

C'est la couche la plus basse de notre système. Elle est implémentée selon le principe de mapping objet-relationnel. Cette implémentation nous permet, par l'intermédiaire d'un ensemble de classes, de manipuler les données de la base de données relationnelle sous forme d'objet. Cette manipulation est beaucoup plus pratique ; en effet cela nous permet de nous défaire de toute la couche SQL lors de la manipulation de la base de données. De plus, cela permet de définir clairement la limite entre la persistance et la couche métier, ce qui se révèle être très utile dans le cas d'une application trois-tiers.

L'ensemble des classes de cette couche sont générées manuellement en se basant sur le principe mapping objet-relationnel. En effet, le mapping objet-relationnel (en anglais Object-Relational Mapping ou ORM) est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé. Ce concept consiste à mettre en correspondance chaque table de la base de données avec une classe dont le nom et les attributs seront identiques au nom et aux champs de la table considérée. Cependant, chaque classe dispose d'un ensemble de méthodes à concevoir et à implémenter selon les besoins de l'application.

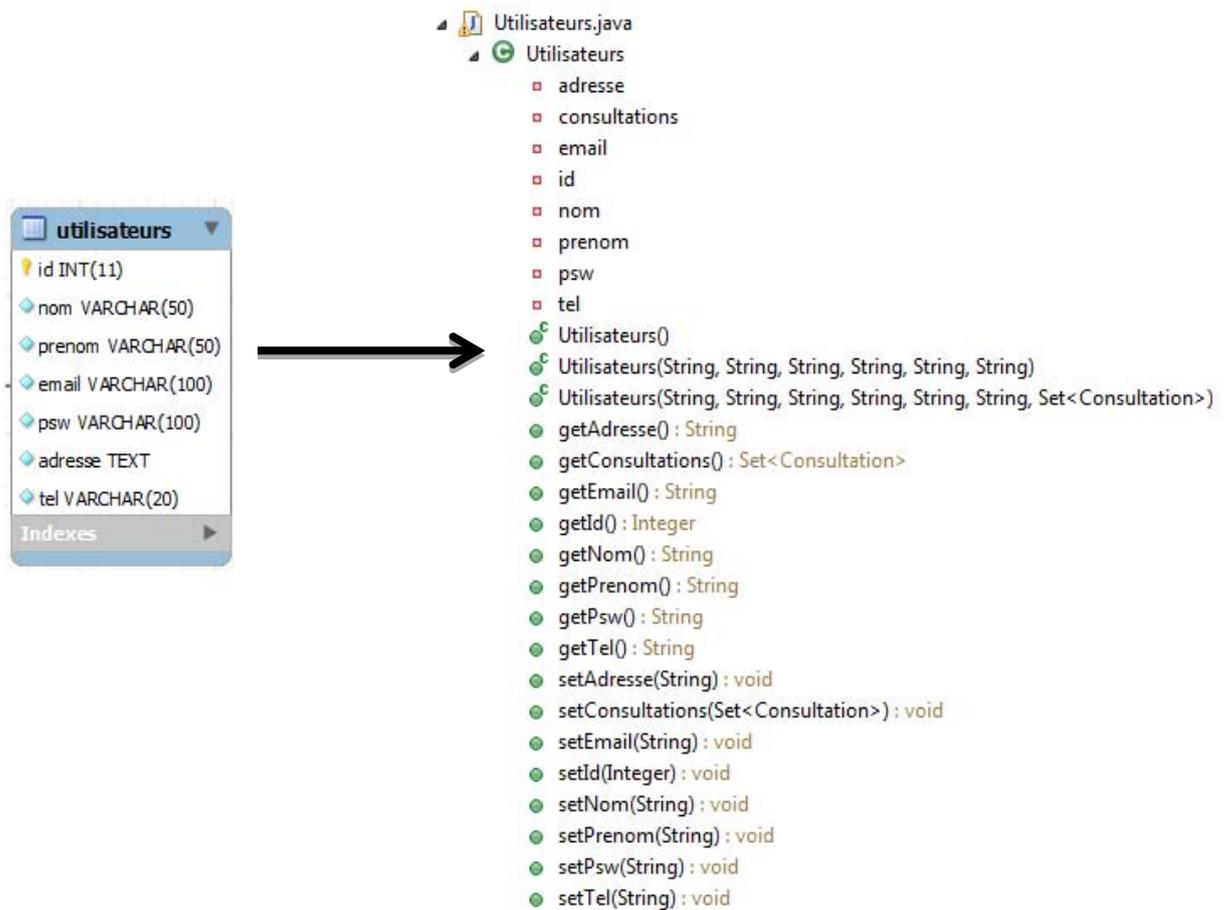


Figure 36: Migration du modèle relationnel en modèle Objet

## 2. Couche métier

Cette couche intègre les fonctionnalités et les traitements propres à l'application. Les composants métier sont généralement dissimulés derrière une sous-couche d'interface de service qui agit comme une façade afin de dissimuler la complexité de l'implémentation.

Pour faciliter la maintenance du système et augmenter la réutilisation de ces composantes nous avons décidé de séparer les traitements métier suivant les tables de la base de données c'est-à-dire que nous allons regrouper pour chaque table toute la logique qui peut s'y appliquer dans une seule classe de la couche métier. Chaque classe métier fournit toutes les méthodes gérant le traitement métier appliqué sur la table correspondante.

## 3. Couche présentation



La couche présentation correspond à la partie visible de notre application. En fait, c'est l'interface interactive avec les utilisateurs.

Cette interface est connue sous le nom d'Interface Homme Machine (IHM). Elle présente comme avantages une simplicité et une facilité d'utilisation, elle permet aux utilisateurs de manipuler de manière transparente les données complexes.

Cette couche assure le relais entre les requêtes de l'utilisateur et entre la couche métier qui, en retour, lui présente les informations renvoyées par le traitement des requêtes.

Il s'agit donc d'un ensemble de pages web, consultées par les utilisateurs de l'application via leur navigateur web. Cet ensemble d'interfaces fait l'assemblage des services métiers et applicatifs offerts par la couche métier.

### **III. Conception des classes**

Un diagramme de classe constitue l'un des pivots essentiels dans la modélisation avec le PU. Il permet de définir quelles seront les composantes du système final en mettant en évidence les différentes classes d'un système et en modélisant les relations qui les associent, les interactions et les hiérarchisations.

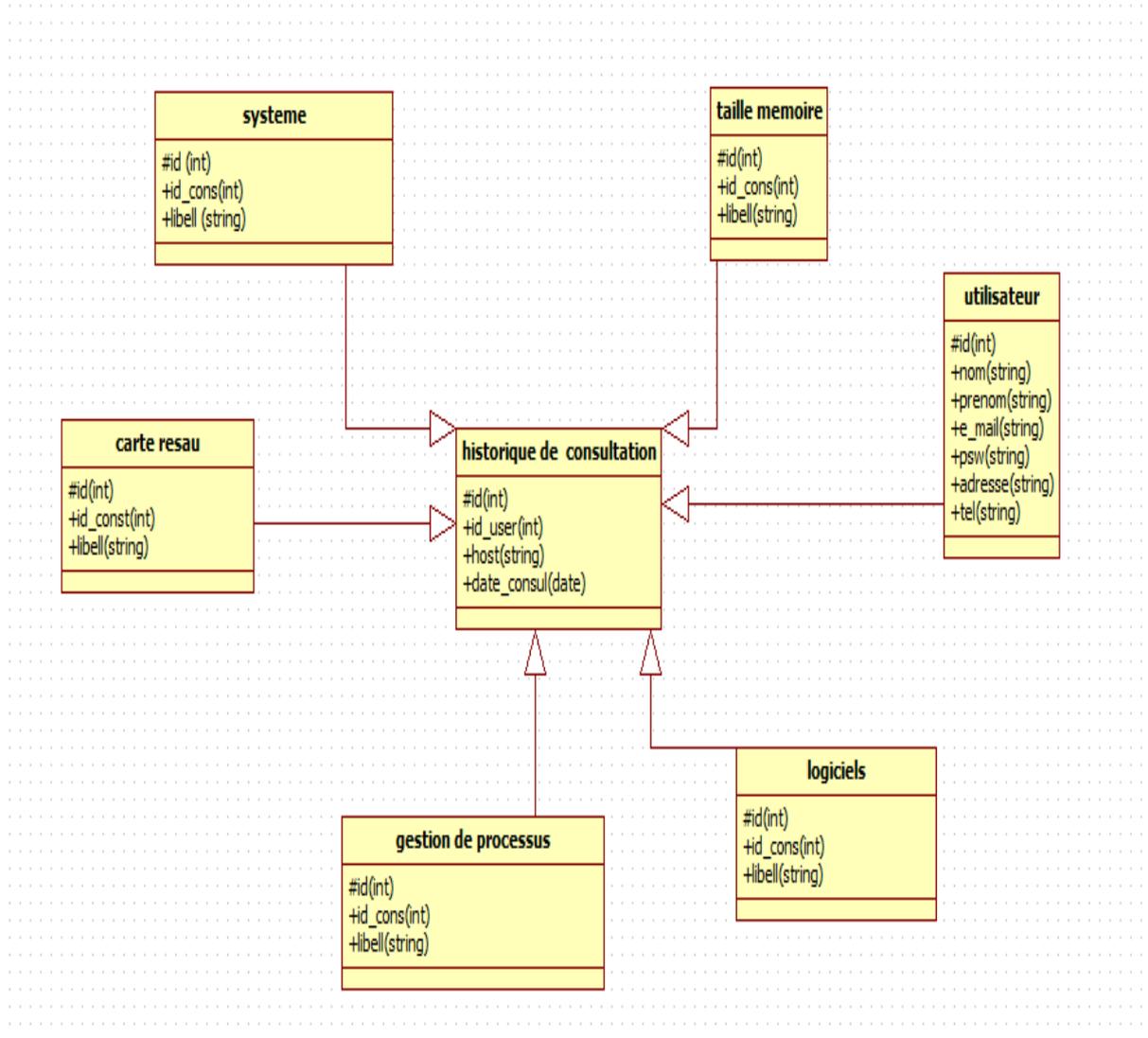


Figure 37: Diagramme de classe

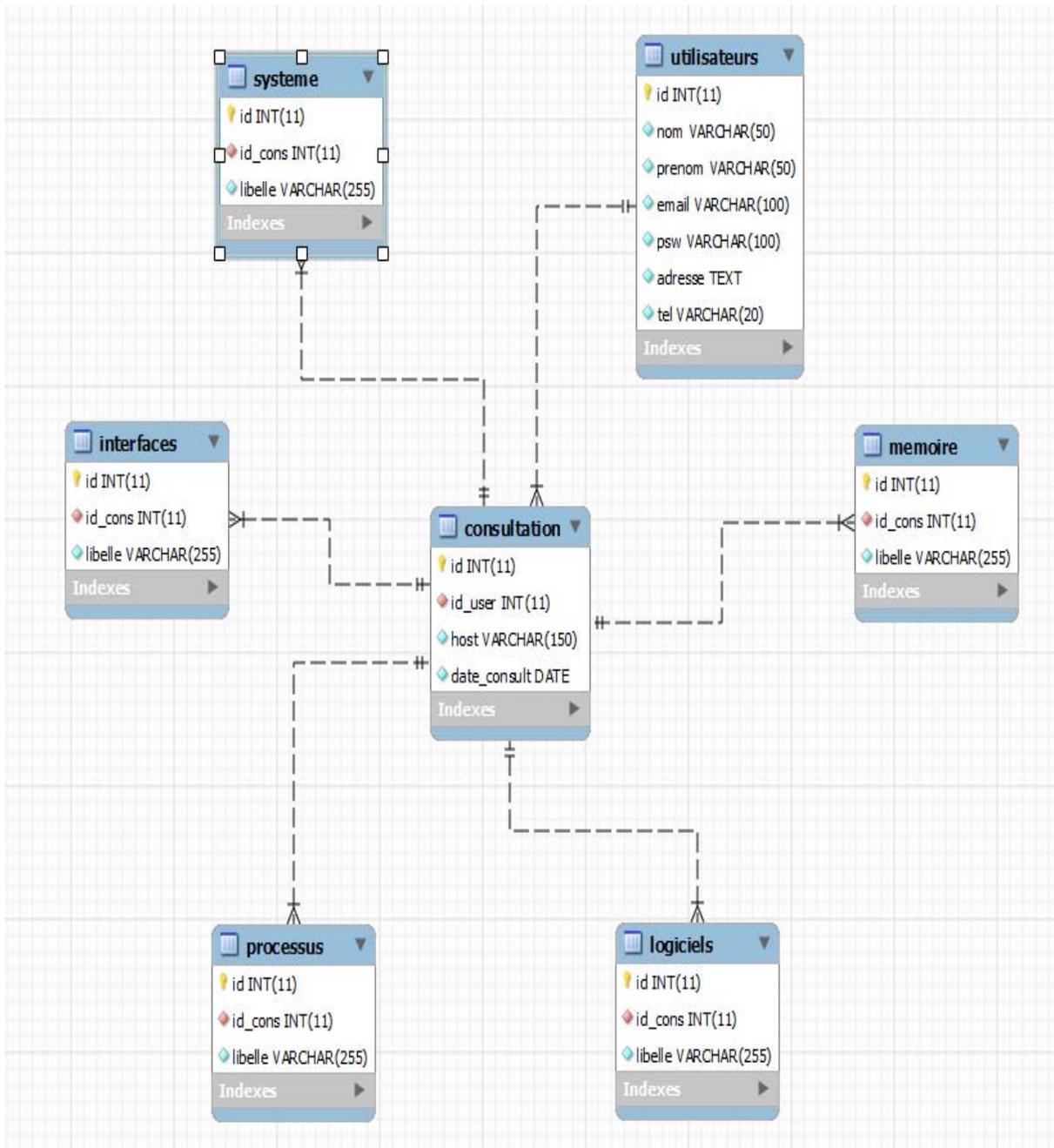


Figure 38: Schéma relationnel de la base de données

#### IV. Conception des cas d'utilisation de priorité 1

##### 1. Conception de cas d'utilisation « s'authentifier »

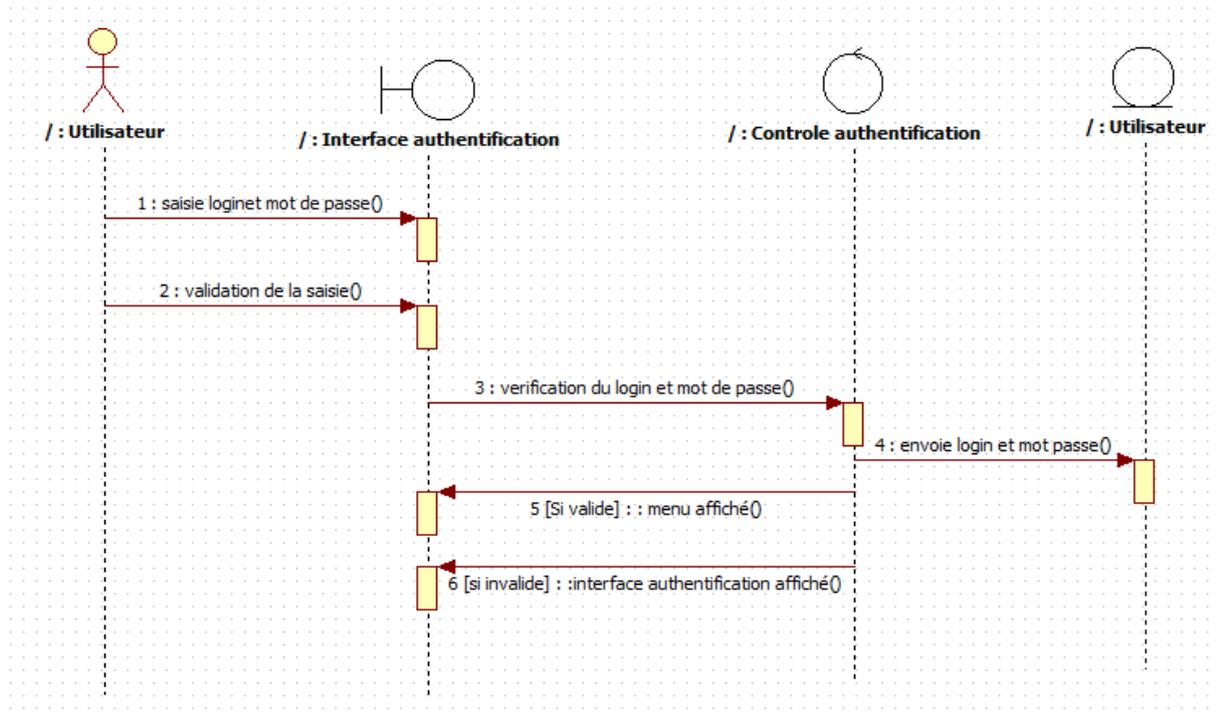


Figure 39: Diagramme de séquence de cas « s'authentifier »

## V. Conception de cas d'utilisation de priorité 2

### 1. Conception de cas d'utilisation « Consultation »

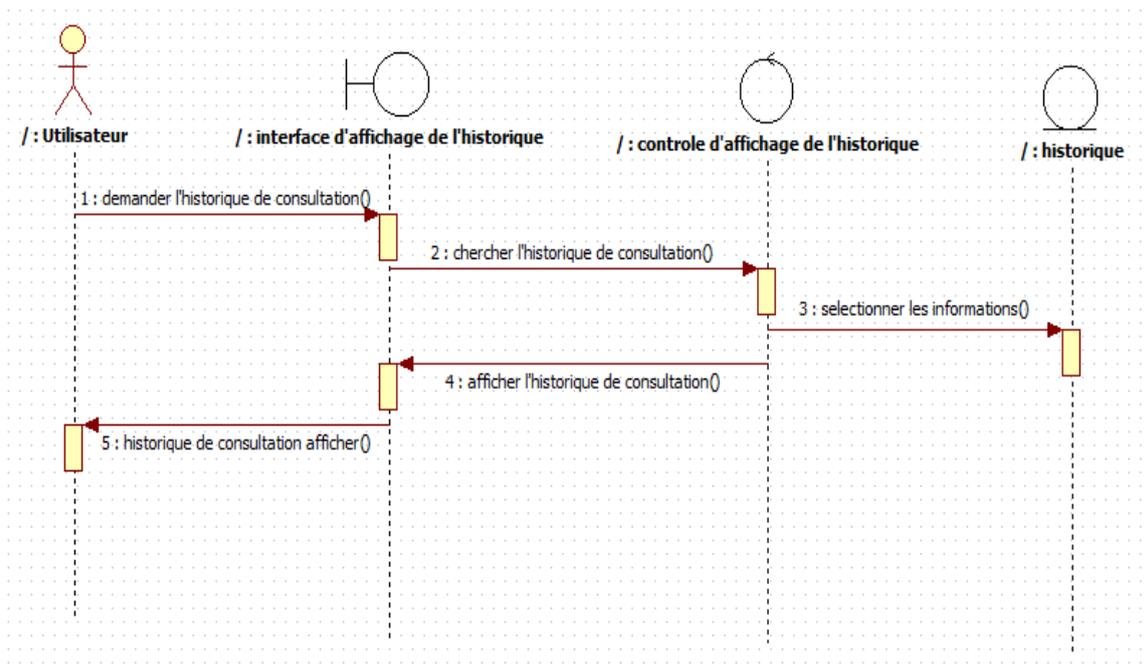


Figure 40: Diagramme de séquence de cas « Consultation »

## 2. Conception de cas d'utilisation « Afficher mémoire »

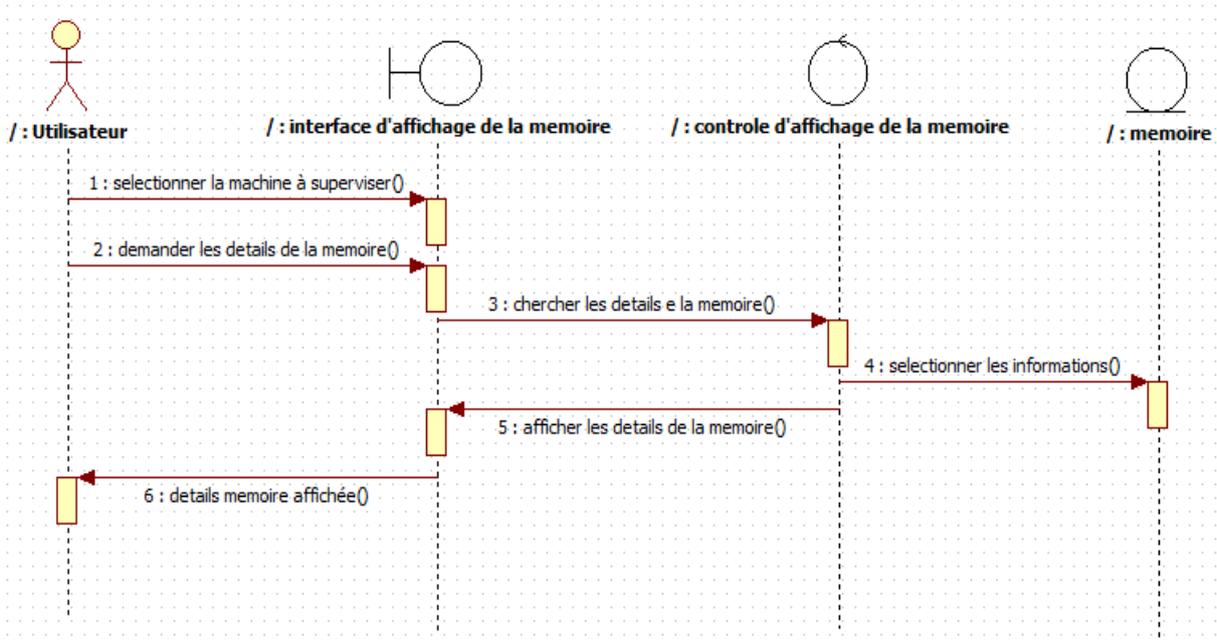


Figure 41: Diagramme de séquence de cas « Afficher mémoire »

## 3. Conception de cas d'utilisation « Afficher interface »

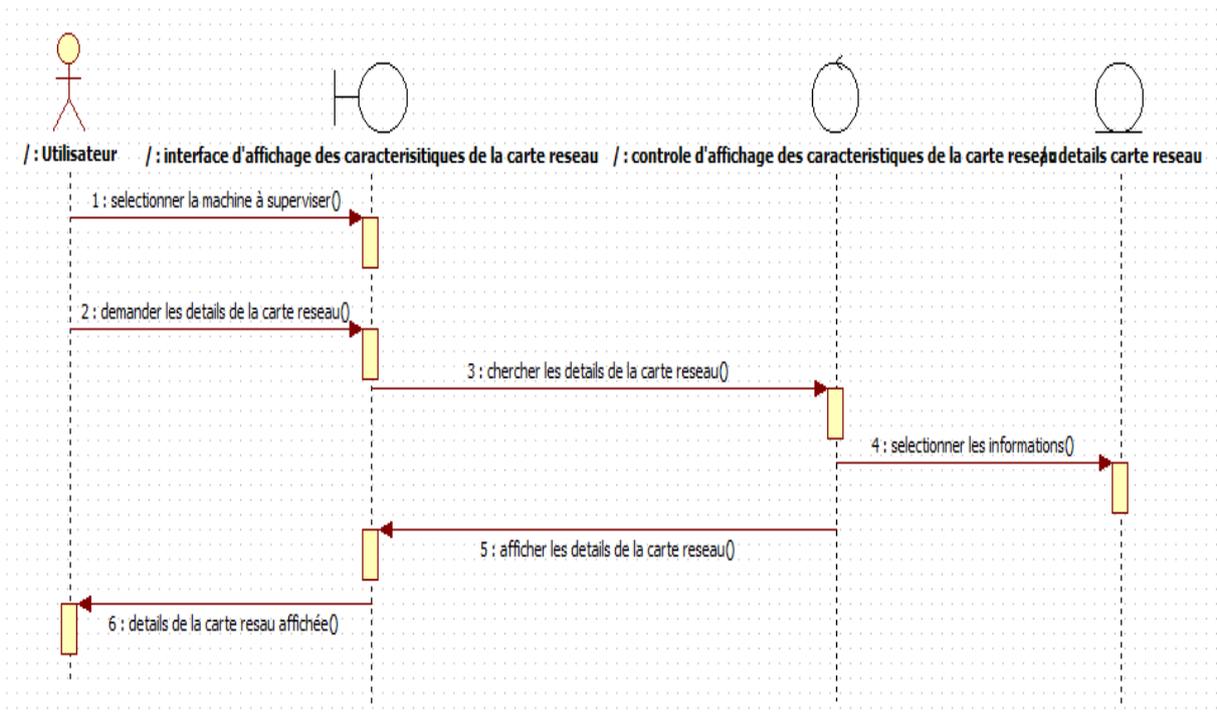


Figure 42: Diagramme de séquence de cas « Afficher interface réseau »

#### 4. Conception de cas d'utilisation « Afficher processus »

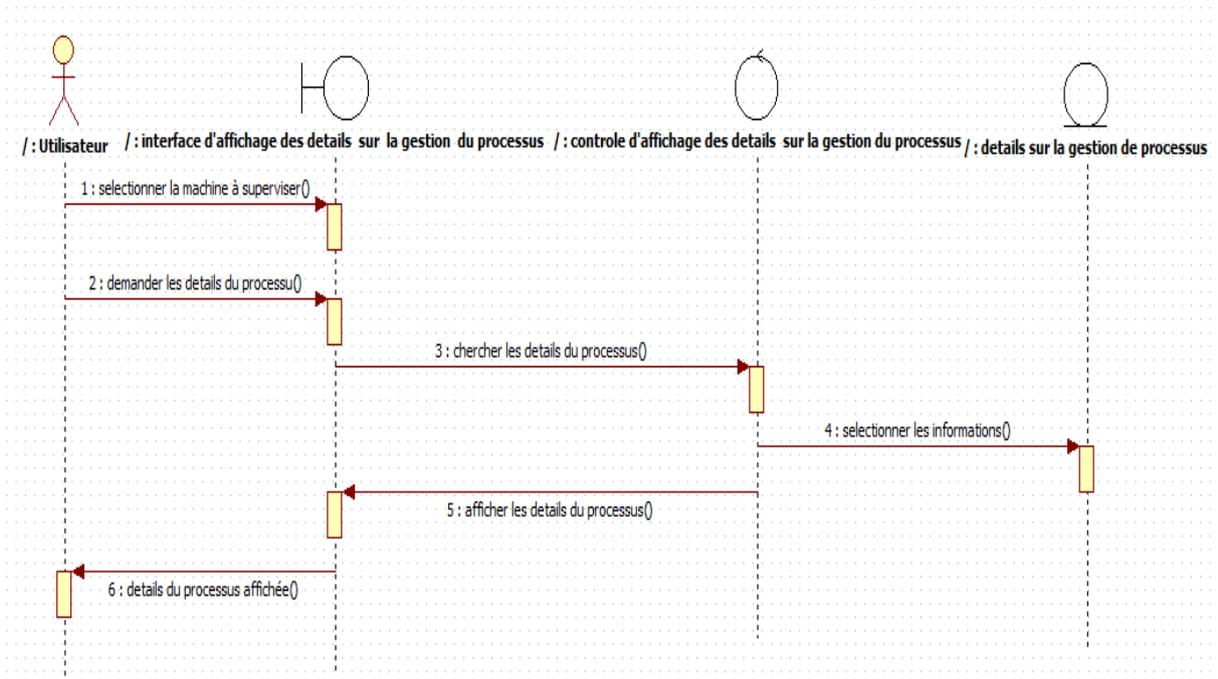


Figure 43: Diagramme de séquence de cas « Afficher processus »

#### 5. Conception de cas d'utilisation « Gérer les systèmes »

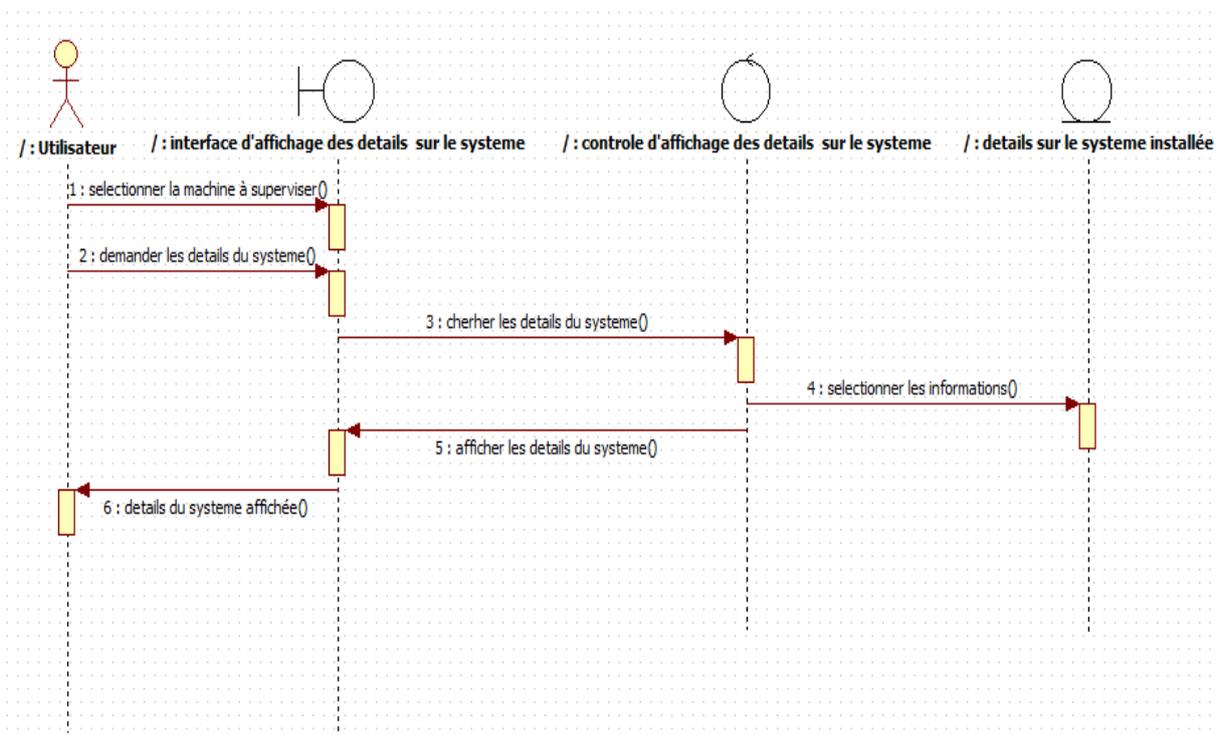


Figure 44: Diagramme de séquence de cas « Gérer les systèmes »

## 6. Conception de cas d'utilisation « Logiciels »

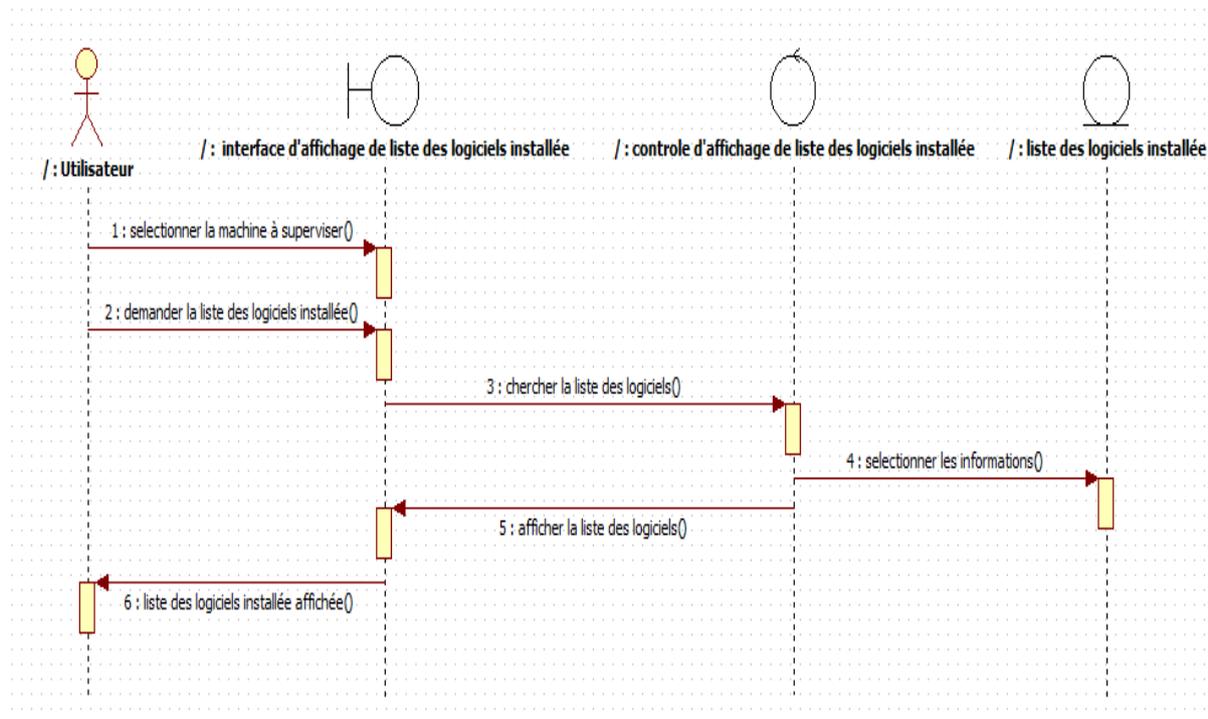


Figure 45: Diagramme de séquence de cas « Logiciels »

### Conclusion

A travers ce chapitre, nous avons présenté notre conception de l'application. Nous avons fourni, dans un premier lieu, une conception globale à travers un schéma général décrivant l'organisation de notre système. Ensuite, nous avons présenté la conception détaillée de l'application à travers les diagrammes de description de chaque couche et des scénarios. A présent, nous sommes capables d'entamer la partie réalisation.



# CHAPITRE

# 4

# RÉALISATION

## Introduction

Dans cette partie, il est question de décrire l'aspect d'implémentation. Nous commencerons alors par présenter les choix technologiques. Puis, nous spécifierons l'environnement matériel et logiciel supportant notre application. Ensuite, nous allons présenter notre application ainsi qu'une description générale.

### I. Environnement et outils du travail

#### 1. Configuration matérielle

Pour la réalisation de cette application nous avons disposé d'un PC portable ayant les caractéristiques suivantes :

- Processeur : Intel(R) Core(TM) 2 Duo CPU T5670 1.80 GHz
- Mémoire installée (RAM) : 3,00 Go
- Disque dur : SATA 160 Go
- Système d'exploitation : Windows 7 Edition Intégrale

#### 2. Configuration logicielle

Tout le long de la phase de développement, nous nous sommes servi de l'environnement logiciel suivant :

-  **Eclipse** [2] : Eclipse est projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libres qui soit extensible, universel et polyvalent, en s'appuyant principalement sur JAVA.  
Eclipse est devenu un standard du marché des logiciels de développement, intégré par de grands éditeurs logiciels et sociétés de services.
-  **Apache Tomcat 6.0** [3]: Apache Tomcat est un conteneur libre de servlet Java 2 Enterprise Edition (J2EE). Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache, il implémente les spécifications des servlets et des

JSP de Sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP.

-  **JDK Java 1.6** : Le conteneur de servlets Tomcat 6.x nécessite une machine virtuelle Java 1.6.
-  **WampServer** [4]: WampServer est une plate-forme de développement web sous Windows pour des applications web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.
-  **MySQL** : MySQL est la base de données open source la plus populaire au monde, il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données (SGBD) les plus utilisées au monde .MySQL peut vous aider à concevoir des applications de base de données évolutives et hautement performantes.
-  **StarUML** [5] : StartUML est un logiciel de modélisation UML/MDA rapide, flexible, disponible en open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU CPL.  
L'objectif du projet StartUML est de construire un outil de modélisation de logiciels et aussi la plate-forme qui est un remplacement convaincant d'outils UML commerciaux tels que Rational Rose, Together et ainsi de suite.

## II. Présentation générale de l'application

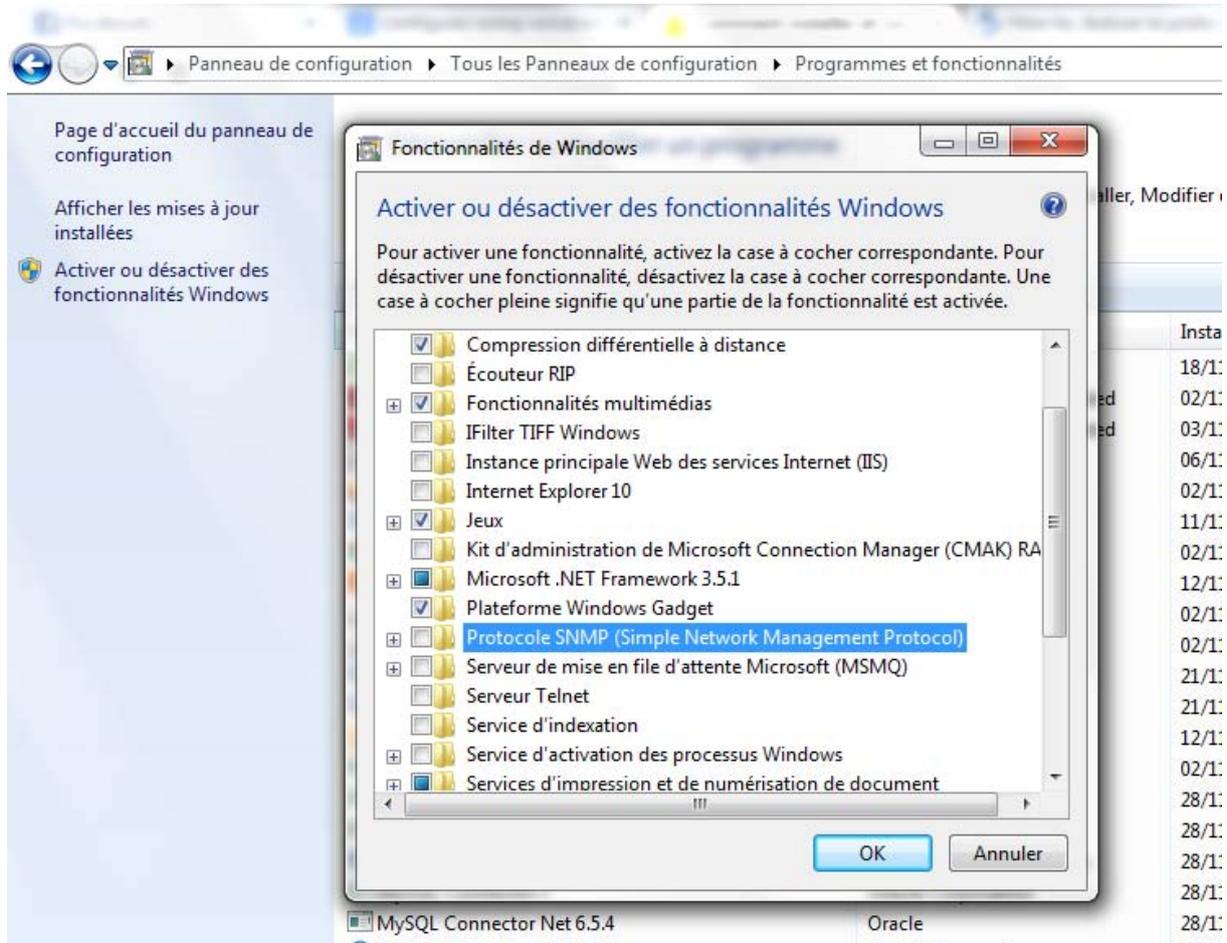
Notre application consiste à assurer le suivi et la supervision qui permet à une entreprise d'assurer la haute disponibilité des différentes machines qui sont du même réseau.

### 1. Activer et configurer le protocole SNMP [6]

Pour installer l'agent Microsoft SNMP, on doit entrer dans le « Panneau de configuration » puis on clique sur « Programmes » puis dans le menu on sélectionne « Activer ou désactiver les fonctionnalités Windows.

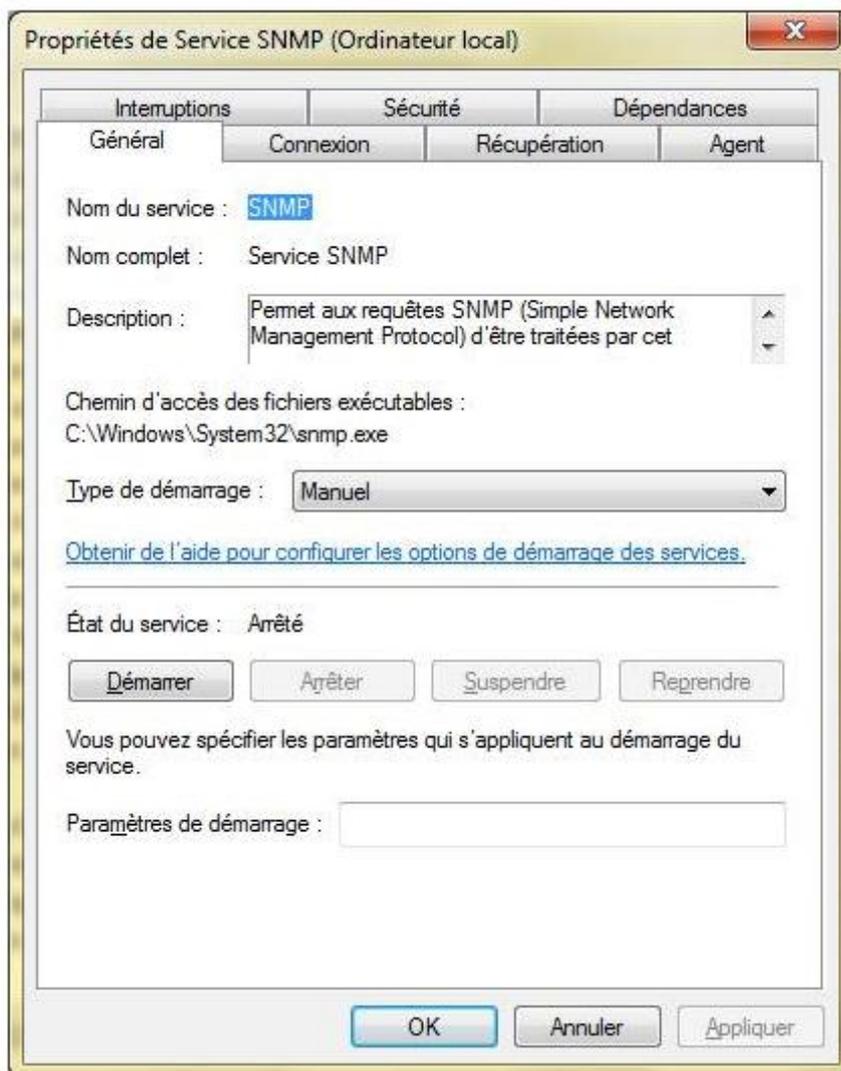
Pour installer l'agent SNMP et d'autres services SNMP, dans la liste des fonctionnalités, on doit cocher sur « Protocole SNMP Simple Network Management Protocol.

La configuration du service SNMP est effectuée par le biais de l'option de propriétés de service. Pour y accéder, on ouvre le panneau de configuration et on sélectionne « outils d'administration » ensuite on sélectionne l'icône des Services.



**Figure 46: Activation du protocole SNMP**

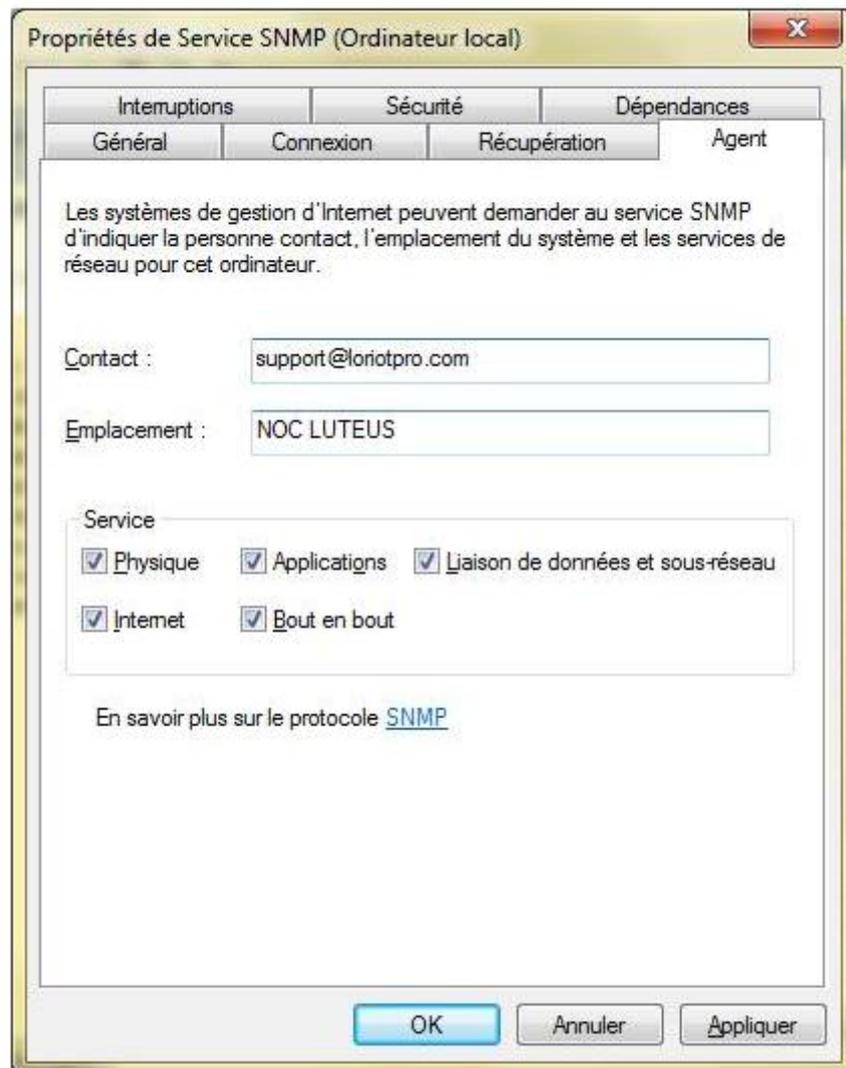
Le processus SNMP s'exécute sous le compte système local ou un compte spécifié. Dans l'onglet Agent, les variables SNMP de la Mib2 system peuvent être définies



**Figure 47: Propriétés de service SNMP « Général »**

- ❖ Contact : Nom et les coordonnées de l'administrateur (objet syscontact de la mib2)
- ❖ Emplacement : Emplacement du dispositif. Ici vous pouvez entrer l'adresse, le numéro de bâtiment, étage, salle, numéro de rack.(objet syslocation de la mib2)
- ❖ Services : les propriétés avancées de l'agent indiquant les fonctions fournies par cet équipement: (objet syservices de la mib2)
- ❖ Physique : Cet équipement propose des services physiques au réseau, hub, répéteur Ethernet Liaison de données et de sous-réseaux : Cet équipement propose des services de liaison, par exemple, pont, (Couche 2 du modèle OSI).

- ❖ Internet : Cet équipement propose des services de transport IP (Couche 3 du modèle OSI)
- ❖ End-to-end : Cet équipement propose des services de bout en bout (Protocole TCP). (Couche 4 du modèle OSI)
- ❖ Applications : Cet équipement propose des services d'application, serveur d'application (couche 7 du modèle OSI)



**Figure 48: Propriétés de service SNMP « Agent »**

- Nom de la communauté à accepter. Le service SNMP nécessite la configuration d'au moins un nom de communauté par défaut. Le nom **public** est généralement utilisé comme nom de la communauté parce que c'est le nom commun qui est universellement reconnu dans toutes les implémentations de SNMP. Vous pouvez supprimer ou modifier le nom



de la communauté par défaut ou ajouter plusieurs noms de communautés. Si l'agent SNMP reçoit une demande d'une communauté qui n'est pas sur cette liste, il peut générer un Trap d'authentification (option ci-après). Si aucun nom de la communauté n'est défini, l'agent SNMP refuse toutes les requêtes entrantes SNMP en provenance des managers SNMP (LoriotPro).

- Autorisations. Vous pouvez sélectionner les niveaux d'autorisation qui déterminent la façon dont un agent traite les demandes SNMP de diverses communautés. Par exemple, vous pouvez configurer le niveau d'autorisation pour bloquer l'agent SNMP de traiter toute demande d'une communauté spécifique.
- Accepter des paquets SNMP de n'importe quel hôte. Dans ce contexte, l'hôte de la source et la liste des hôtes acceptables consulter le système de gestion SNMP source et la liste des autres systèmes de gestion acceptable. Lorsque cette option est activée, aucun des paquets SNMP n'est rejeté. Fondée sur le nom, sur l'adresse de l'hôte source ou sur la base de la liste des hôtes acceptables, cette option est activée par défaut.
- Accepter uniquement les paquets SNMP provenant de ces hôtes. Cette option offre une sécurité limitée. Lorsque l'option est activée, seuls les paquets SNMP reçoivent des hôtes sur une liste d'hôtes acceptables qui sont déjà acceptés. L'agent SNMP rejette les messages des autres hôtes et envoie un piège d'authentification.
- Envoyer des interruptions d'authentification. Lorsqu'un agent SNMP reçoit une demande qui ne contient pas un nom valide de communauté ou l'hôte qui envoie le message n'est pas sur la liste des hôtes acceptés, l'agent peut envoyer un message Trap d'erreur d'authentification à un ou plusieurs destinations du Trap (manager SNMP (LoriotPro)).

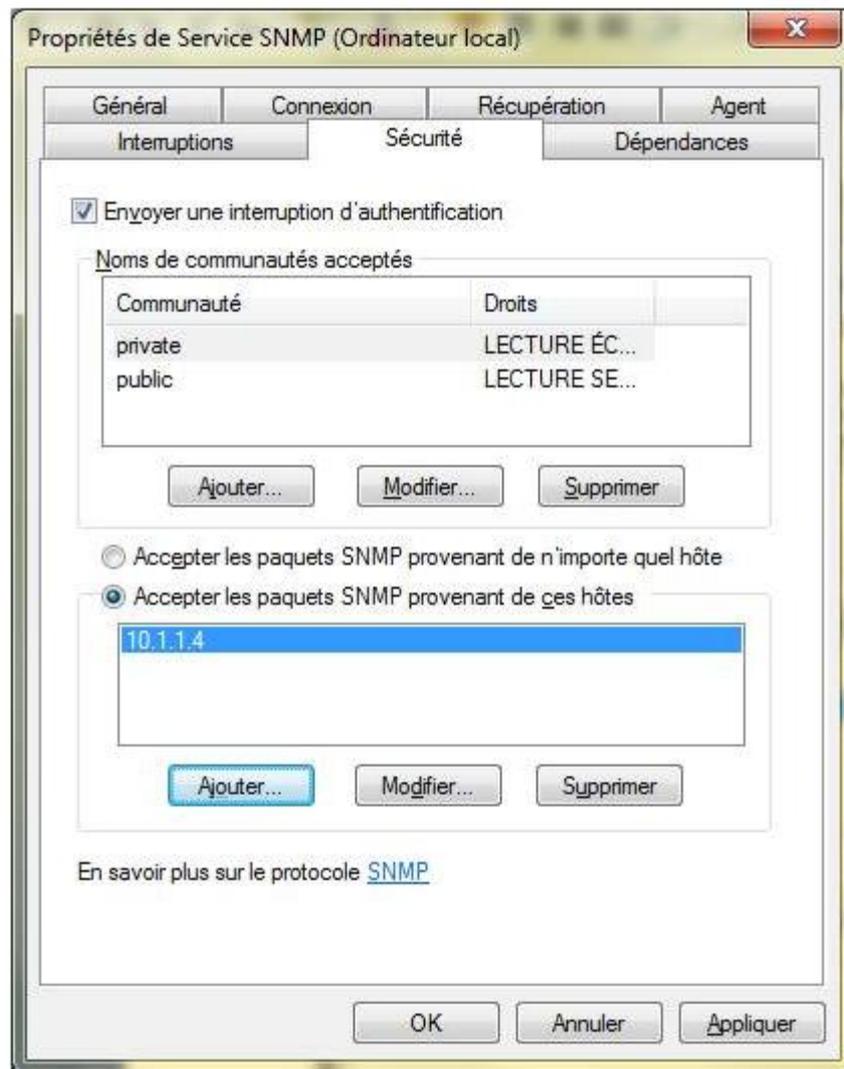


Figure 49: Propriétés de service SNMP « Sécurité »

## 2. Principe de fonctionnement de l'application

Le schéma suivant montre le principe de fonctionnement de notre application en mettant en valeur les principales connexions et les relations entre les différents composants après avoir effectué une configuration SNMP sur un serveur ou une machine A dont on souhaite récupérer les informations de fonctionnement pour les exploiter sur une autre machine B.

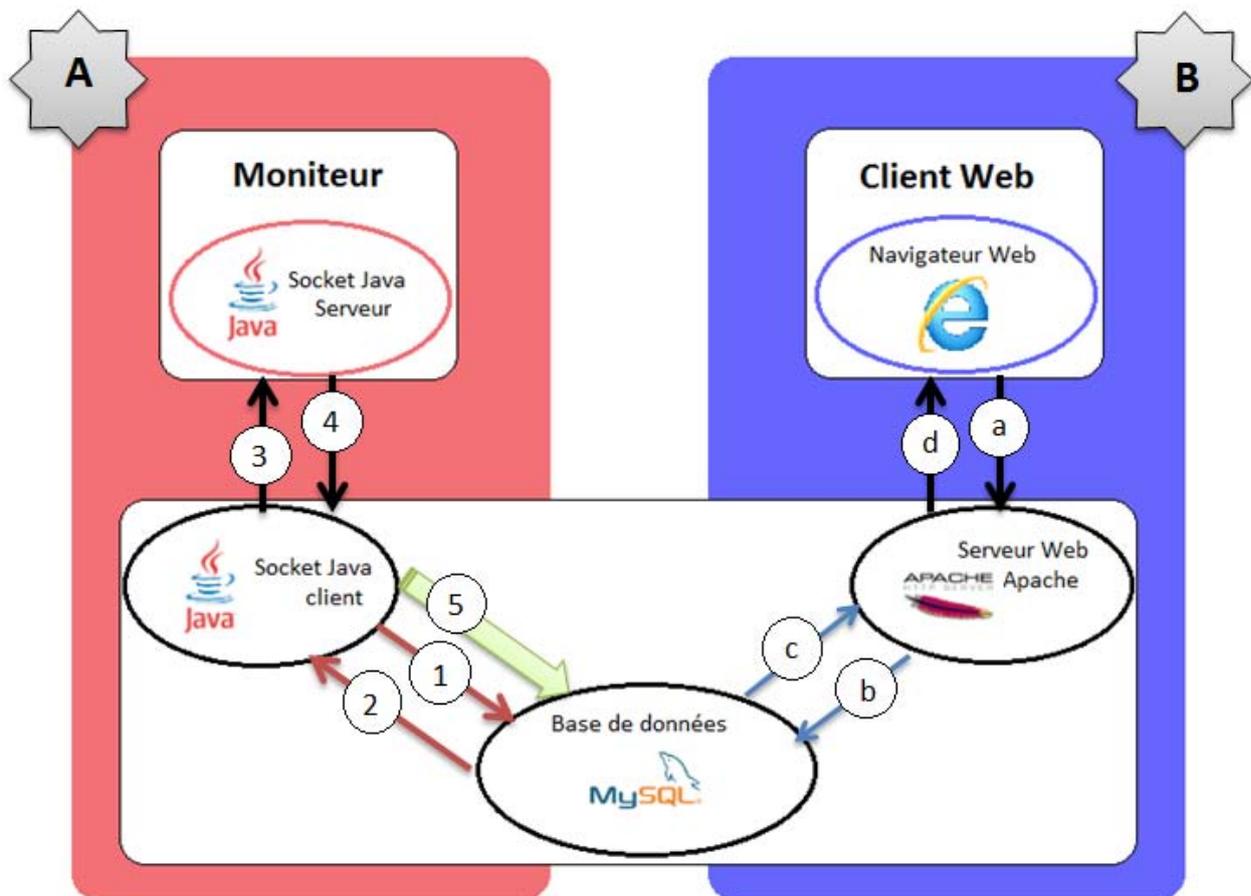


Figure 50: schéma du principe de fonctionnement de l'application de la supervision

### 2.1 Description de la partie A :

1. Le client Socket-java demande qui est le client d'où le serveur censé doit le collecter.
2. La réponse de la base de données à la requête (1).
3. Le client Socket-java contacte le serveur Socket –java du client en question pour la collecte d'informations.
4. Envoie des résultats des informations collectés du serveur Socket au client Socket.
5. Insertion des informations collectées dans la base de données MySQL.

### 2.2 Description de la partie B :

- a) Le client web se connecte au serveur web et demande l'information collectée au niveau de la base.
- b) Recherche de l'information en question dans la base de données MySQL.



c) Envoi du résultat de la recherche

d) Affichage et traitement des résultats de la recherche des informations

### **III. Description de l'application de la supervision :**

Notre application est composée de trois parties :

1. la première partie consiste à créer des programmes en langage java en utilisant les sockets-java permettant d'exécuter des scripts pour l'exécution des commandes bien spécifique et aussi pour la collecte d'information au niveau des machines clientes.
2. la deuxième partie consiste à créer des tables au niveau de la base de données MySQL. Ces tables permettent l'enregistrement des données collectées au niveau des clients.
3. La troisième partie consiste à créer l'interface web de notre application de la supervision et d'interpréter graphiquement les données collectées.

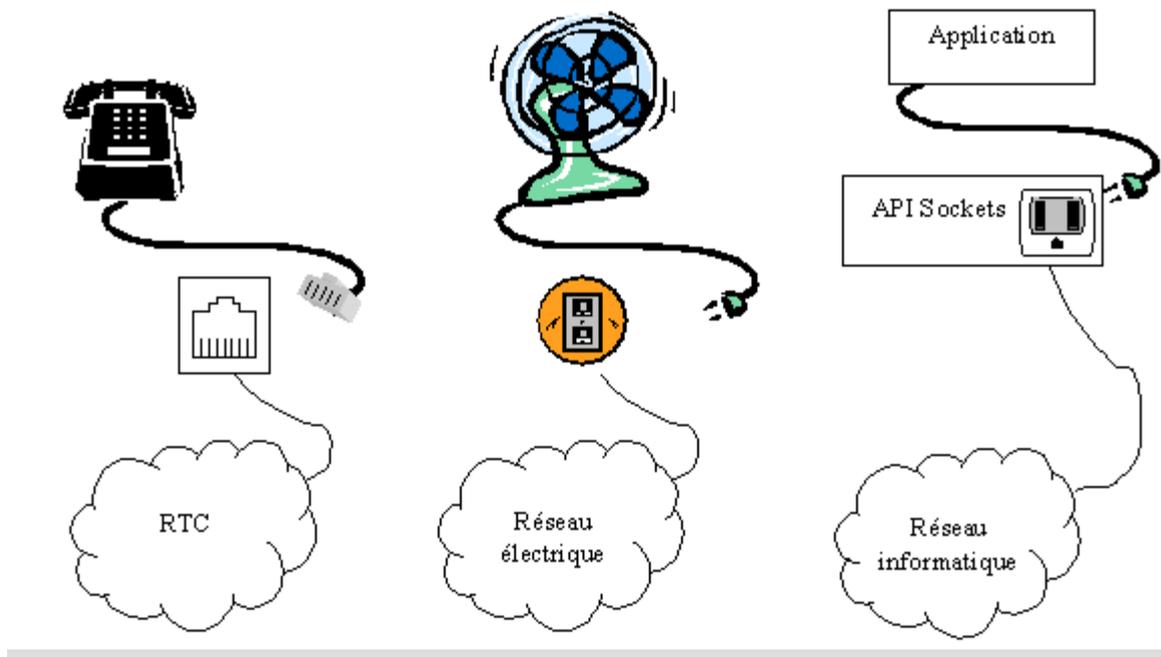
#### **1. Concept de sockets [7]**

Les Sockets forment une API (Application Program Interface): ils offrent aux programmeurs une interface entre le programme d'application et les protocoles de communication. En aucun cas, les sockets ne forment une norme de communication ou une couche de protocole à l'instar de TCP/IP.

Les sockets (prises de raccordement) forment un mécanisme de communication bidirectionnel interprocessus dans un environnement distribué ce qui n'est pas le cas des autres outils tels que les pipes. Ils permettent évidemment la communication interprocessus à l'intérieur d'un même système.

L'interface des "sockets" n'est pas liée à une pile de protocoles spécifiques. Dans cette partie, nous nous intéresserons, à l'utilisation des sockets dans le monde TCP/IP.

La notion de socket en tant que prise de raccordement vient d'une analogie avec le réseau électrique et le réseau téléphonique :



**Figure 51 : La métaphore des prises**

Les sockets représentent donc d'une part une API c'est à dire un ensemble de primitives de programmation et d'autre part les extrémités de la communication (notion de prise). Les extrémités de communication sont identifiées dans le monde TCP/IP par trois informations : une adresse IP, le protocole utilisé (TCP ou UDP) et un numéro de port (entier sur 16 bits donc de 0 à 65535). En fonction du protocole transport utilisé, les sockets vont fonctionner différemment. Nous présenterons dans notre rapport seulement les sockets utilisant le protocole TCP qui fonctionnent en utilisant le modèle client/serveur et offrent une communication par flux.

## 2. Adresses IP

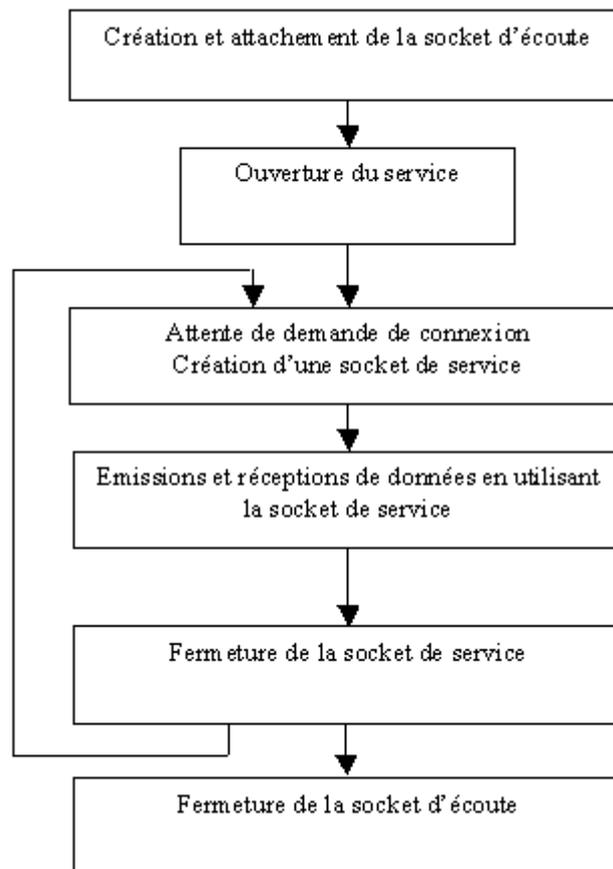
Une machine (appelée aussi hôte ou host) est identifiée dans l'Internet par son adresse. L'adresse IP d'une machine correspond à un numéro qui est unique dans le monde. Il existe actuellement cinq classes d'adresses IP. Les trois premières permettent de gérer des réseaux de tailles diverses. La classe D permet de gérer une communication multipoint (un message est envoyé à plusieurs machines à la fois). La classe E est réservée et ne sera probablement jamais utilisée puisqu'on devrait bientôt migrer vers la nouvelle version d'IP IPv6 qui stockera les adresses IP dans 16 octets.

### 3. Sockets TCP

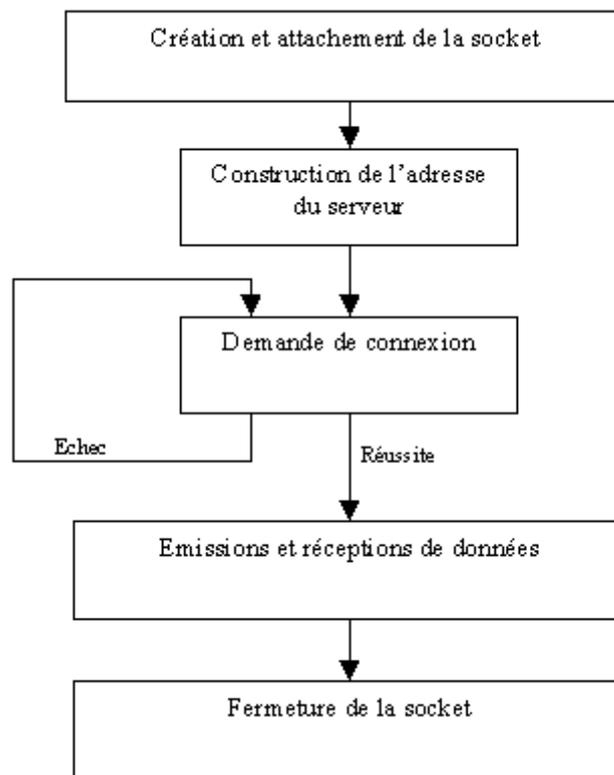
#### 3.1 Le modèle client/serveur

Le protocole TCP offre un service en mode connecté et fiable. Les données sont délivrées dans l'ordre de leur émission. La procédure d'établissement de connexion est dissymétrique. Un processus, appelé serveur, attend des demandes de connexion d'un processus, appelé client, qu'il lui envoie. Une fois l'étape d'établissement de connexion effectuée le fonctionnement redevient symétrique.

Les deux schémas suivants présentent les algorithmes de fonctionnement des clients et serveurs. Il est à noter que côté serveur on utilise deux sockets : l'un, appelé socket d'écoute, reçoit les demandes de connexion et l'autre appelé socket de service, sert pour la communication. En effet, un serveur peut être connecté simultanément avec plusieurs clients et dans ce cas on utilisera autant de sockets de services que de clients.



**Figure 52 : Fonctionnement du serveur**

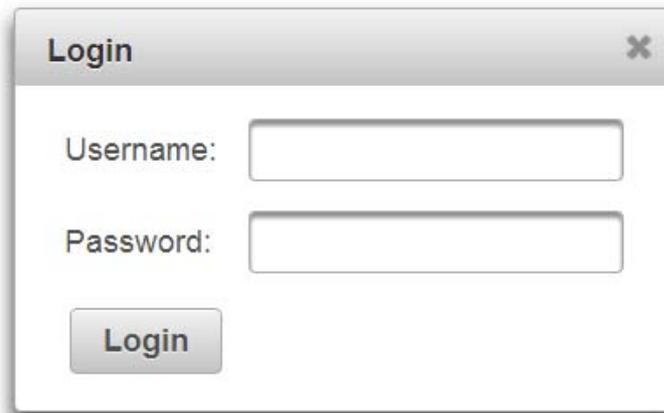


**Figure 53 : Fonctionnement du client**

#### **IV. Interface Homme-Machine**

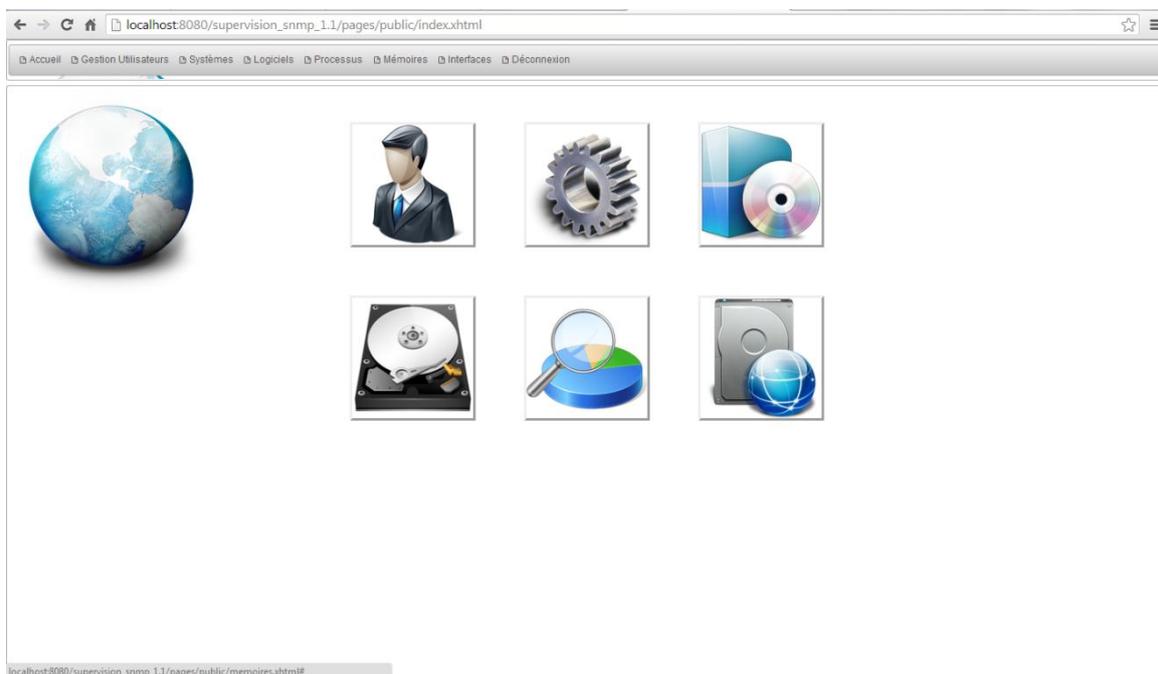
Les interfaces conçues ne sont que les fruits d'une longue chaîne de travail ; ce n'est qu'une simple image dont l'utilisateur ignore les arrières plans.

La page d'authentification : c'est la première page vue par l'utilisateur. L'authentification sert à restreindre l'accès à l'application et à délimiter ce mode d'accès selon les privilèges accordés à l'utilisateur. L'interface ci-dessous présente notre page d'authentification. Il s'agit de saisir l'identifiant et le mot de passe pour pouvoir accéder aux fonctionnalités accordées selon le droit d'accès.



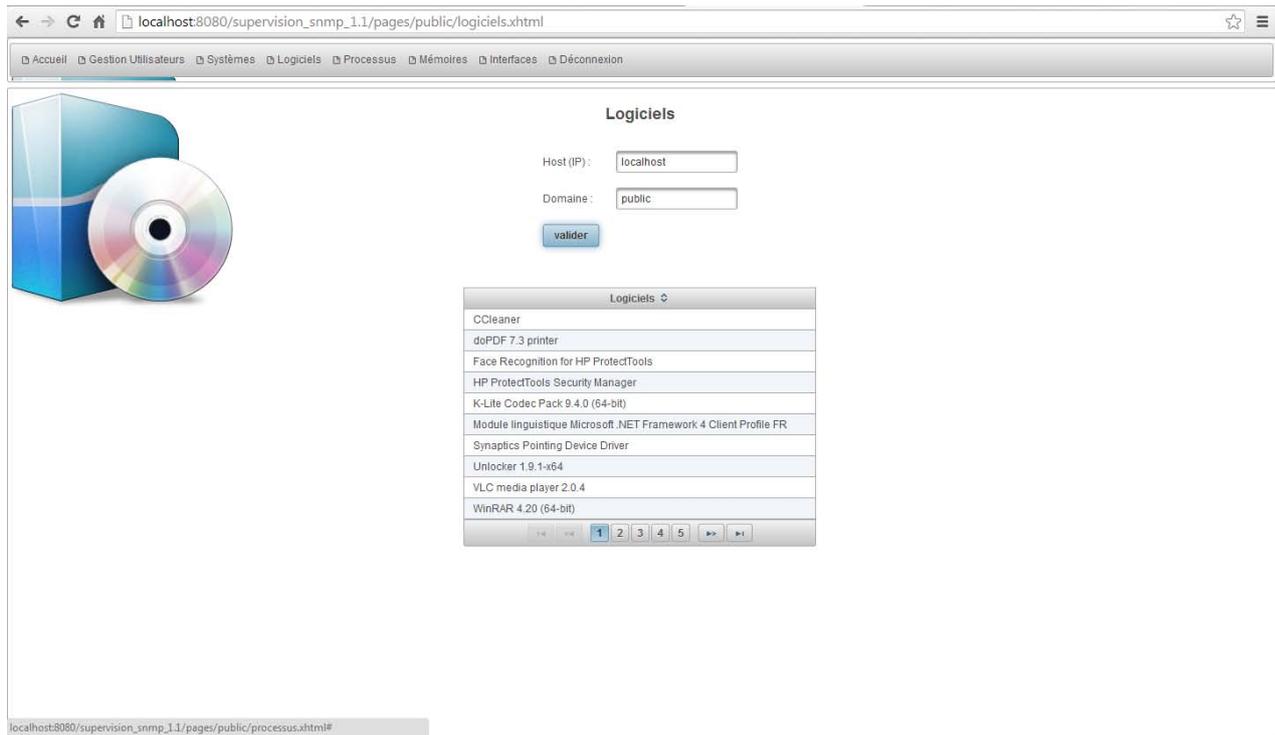
**Figure 54: Interface d'authentification**

Après l'authentification, on accède directement à la page principale : c'est le tableau de bord.



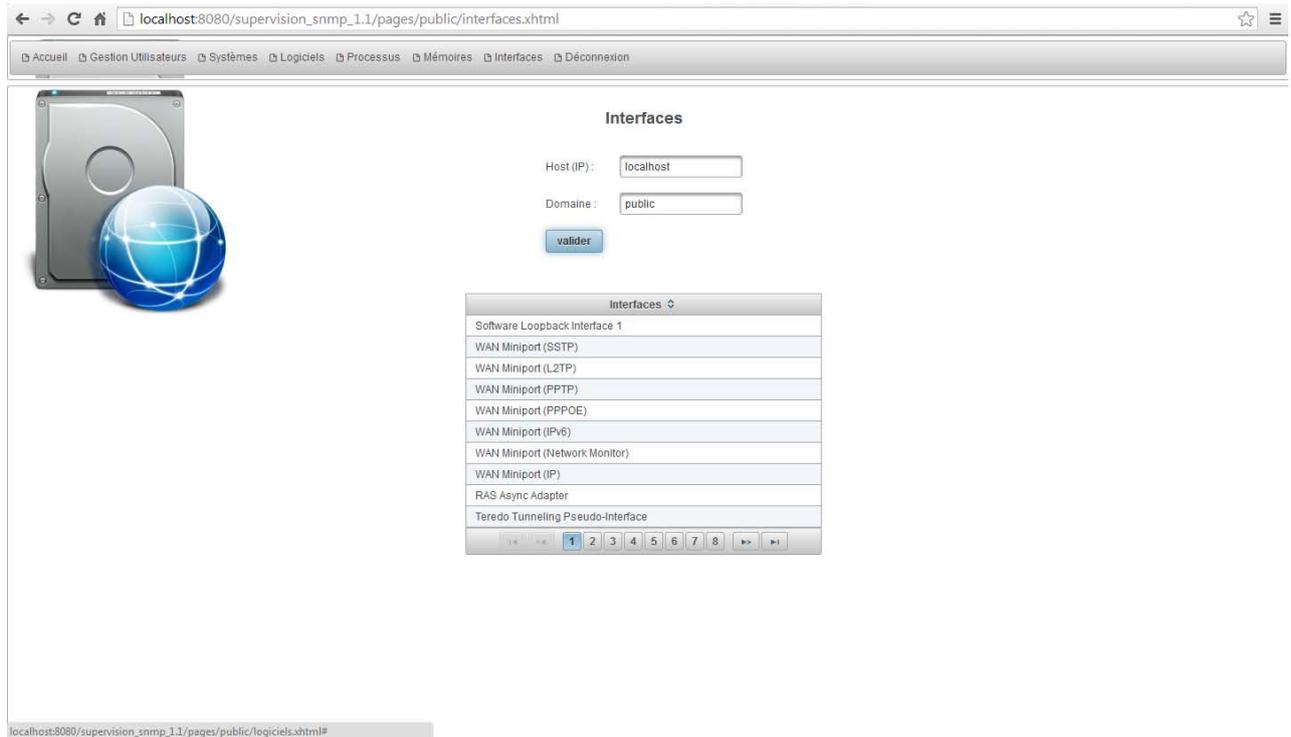
**Figure 55: Page principale du tableau de bord**

Depuis le tableau de bord, l'utilisateur choisit une action parmi les différentes fonctionnalités de notre application comme dans l'exemple de la figure (**figure56**) on peut afficher les listes des logiciels installés dans la machine qu'on veut superviser



**Figure 56: Page logiciels**

L'utilisateur peut aussi effectuer une supervision sur la carte réseau de la machine souhaitée la figure suivante nous montre les détails de la carte réseau et des interfaces réseaux.



**Figure 57 : Page interfaces**

## Conclusion

Au cours de ce chapitre, nous avons présenté en détails la projection de la conception du plan théorique sur le plan pratique. Nous avons décrit les plates-formes matérielles et logicielles sur et avec lesquelles nous avons construit notre application. Nous avons, ensuite, présenté les interfaces les plus significatives de notre application.



## CONCLUSION GÉNÉRALE

Notre stage a été une expérience enrichissante qui nous a permis d'une part d'appliquer et d'approfondir nos connaissances acquises tout au long de notre formation à l'UVT et d'autre part, de nous familiariser avec la conduite des projets informatique et d'explorer le milieu professionnel.

Dans ce stage, nous étions chargés de développer une application web performante et facile à utiliser qui permet la supervision à l'aide d'un outil open source "Nagios".

Ce projet de fin d'études nous a offert l'occasion d'approfondir nos connaissances en matière de modélisation à travers l'utilisation de la méthodologie UML qui est devenue quasiment incontournable. De plus, nous nous sommes familiarisés avec le SGBD et l'environnement Eclipse ainsi que la technologie J2EE, qui constituait une contrainte au départ.

Le développement de notre application, nous a également permis d'approfondir nos connaissances en matière de conduite de projet et de les mettre en pratique. Certes, le travail effectué était tout au long de sa réalisation réponse à ce qui était proposé par l'INM. Mais, il peut être amélioré de différentes manières et évoluer vers une solution plus optimisée.

Enfin, l'application que nous avons réalisée présente un aspect modulaire qui facilite la tâche pour les modifications futures. Nous souhaitons, ainsi, qu'elle apporte l'aide nécessaire aux différents agents de l'INM et restera un noyau riche pour préparer un avenir prospère. Notre application reste toujours extensible par d'autres développeurs pour d'autres améliorations et ajout de nouvelles fonctionnalités ou aussi pour l'entretenir et l'adapter aux besoins émergents.



# REFERENCES

## WEBOGRAPHIE

- [1]: Monitoring d'une infrastructure informatique sur base d'outils libres  
www.memoireonline.com
- [2] Eclipse (projet) fr.wikipedia.org
- [3] Commentçamarche. High-Tech. [en ligne].www.commentçamarche.net
- [4] www.wampserver.com
- [5] StarUML fr.wikipedia.org
- [6] InstallWXPAgent www.loriotpro.com
- [7] Support Sockets www.ouvea.edu.ups-tlse.fr
- <http://staruml.sourceforge.net/en/>
- <http://www.mysql.fr/>
- <http://fr.openclassrooms.com/>
- <http://primefaces.org/>



# ANNEXE

# A



## 1. Le Modele MVC (Model, View, Controler)

Le modèle MVC (Model View Controler) a été initialement développé pour le langage Smalltalk dans le but de mieux structurer une application avec une interface graphique.

Ce modèle est un concept d'architecture qui propose une séparation en trois entités des données, des traitements et de l'interface :

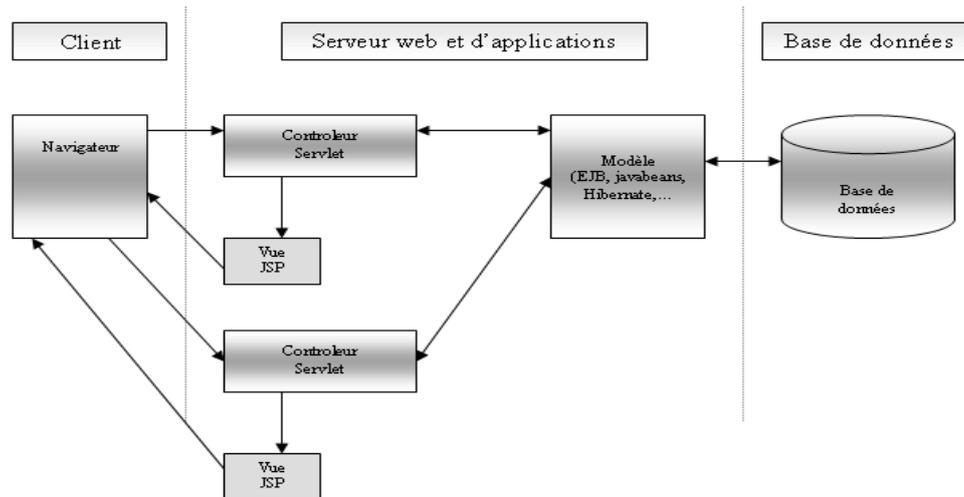
- le Modèle représente les données de l'application généralement stockées dans une base de données.
- la Vue correspond à l'IHM (Interface Homme Machine).
- le Contrôleur assure les échanges entre la vue et le modèle notamment grâce à des composants métiers.

Initialement utilisé pour le développement des interfaces graphiques, ce modèle peut se transposer pour les applications web sous la forme d'une architecture dite 3-tiers : la vue est mise en œuvre par des JSP, le contrôleur est mis en œuvre par des servlets et des Javabeans. Différents mécanismes peuvent être utilisés pour accéder aux données.

L'utilisation du modèle MVC rend un peu plus compliqué le développement de l'application qui le met en œuvre mais il permet une meilleure structuration de l'application.

### 1.1. Modèle MVC 1

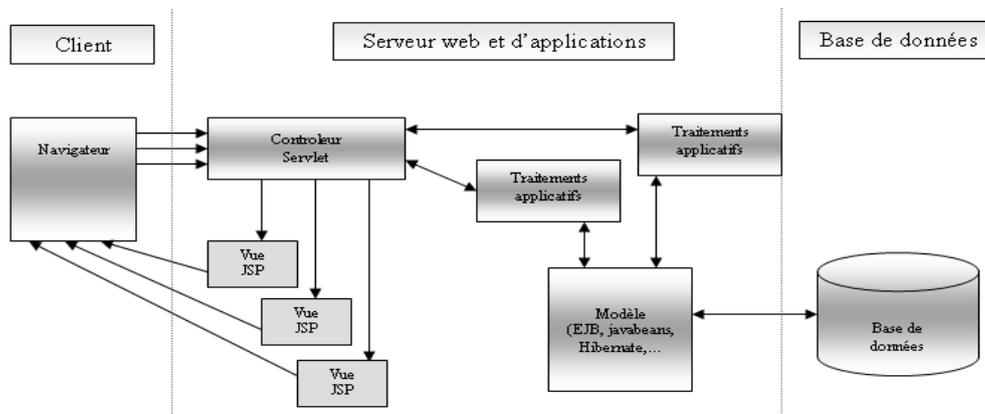
Dans ce modèle, chaque requête est traitée par un contrôleur sous la forme d'une servlet. Celle-ci traite la requête, fait appel aux éléments du modèle si nécessaire et redirige la requête vers une JSP qui se charge de créer la réponse à l'utilisateur.



L'inconvénient est donc une multiplication du nombre de servlets nécessaire à l'application : l'implémentation de plusieurs servlets nécessite beaucoup de code à produire d'autant que chaque servlet doit être déclarée dans le fichier web.xml.

### 1.1. Modèle MVC 2

Le principal défaut du modèle MVC est le nombre de servlets à développer pour une application. Pour simplifier les choses, le modèle MVC modèle 2 ou MVC2 de Sun propose de n'utiliser qu'une seule et unique servlet comme contrôleur. Cette servlet se charge d'assurer le workflow des traitements en fonction des requêtes http reçues. Le modèle MVC 2 est donc une évolution du modèle 1 : une unique servlet fait office de contrôleur et gère toutes les requêtes à traiter en fonction d'un paramétrage généralement sous la forme d'un fichier au format XML.





## 2. Framework



**Spring** : Le framework Spring fournit un modèle de programmation et la configuration globale pour les applications d'entreprise modernes basées sur Java - sur n'importe quel type de plate-forme de déploiement. Un élément clé du printemps est un soutien infrastructurel au niveau de l'application: Printemps met l'accent sur la «plomberie» des applications d'entreprises afin que les équipes puissent se concentrer sur le logique métier du niveau de l'application, sans liens inutiles aux environnements de déploiement spécifiques. Spring Framework inclut un certain nombre de modules différents, ici nous montrons `spring-context` qui fournit les fonctionnalités de base. Reportez-vous aux guides de démarrage sur le droit pour d'autres options.

Une fois que vous avez configuré votre construction avec la dépendance `spring-context` , vous serez en mesure de faire ce qui suit:

```
hello/MessageService.java
```



**Hibernate** : Historiquement, Hibernate a facilité le stockage et la récupération d'objets de domaine Java via mapping Objet / Relationnel. Aujourd'hui, Hibernate est une collection de projets connexes permettant aux développeurs d'utiliser des modèles de domaine POJO style dans leurs applications de manière s'étendant bien au-delà mapping objet / relationnel.



# Table des matières

INTRODUCTION GÉNÉRALE	6
ETAT DE L'ART	8
I. Présentation de Nagios	8
II. Fonctionnement de Nagios	8
III. Fonctionnalités [1]	9
IV. Architecture	10
CHAPITRE 1 Etude Préalable	12
Introduction	13
I. Présentation du projet :	13
II. Présentation de l'organisme d'accueil	14
III. Cahier de charge	15
IV. Objectifs	15
V. Etude de l'existence	16
1. Critique de l'existence	16
2. Solution	16
VI. Cycle de vie	16
VII. Méthodologie de conception	18
1. Processus unifié	18
2. UML	19
2.1 Définition	19
2.2 Les avantages de l'UML	19
3. Logiciel de modélisation	20
Conclusion :	20
CHAPITRE 2 phase d'incubation	21
Introduction	22
I. Capture des besoins	22
1. Spécification des besoins fonctionnels	22
2. Présentation des acteurs	22
3. Spécification des besoins non fonctionnels	23



II. Diagramme de cas d'utilisation global du système	23
III. Affectation des priorités	24
IV. Raffinement de cas d'utilisation de priorité1	24
1. Raffinement de cas « s'authentifier »	24
V. Modèle d'analyse des cas d'utilisation de priorité 1	25
1. Modèle d'analyse de cas d'utilisation « s'authentifier »	25
VI. Raffinement de cas d'utilisation de priorité 2	27
1. Raffinement de cas « Consulter historique »	27
2. Raffinement de cas « Afficher mémoire »	28
3. Raffinement de cas « Afficher interfaces »	28
4. Raffinement de cas « Consulter liste des logiciels »	29
5. Raffinement de cas « Gérer les processus»	30
6. Raffinement de cas « Gérer les systèmes »	31
VII. Modèle d'analyse des cas d'utilisation de priorité 2	31
1. Modèle d'analyse de cas d'utilisation « Consulter historique »	31
2. Modèle d'analyse de cas d'utilisation « Afficher mémoire»	33
3. Modèle d'analyse de cas d'utilisation « Afficher interfaces»	34
4. Modèle d'analyse de cas d'utilisation « Afficher processus»	36
5. Modèle d'analyse de cas d'utilisation « Gérer le système»	37
6. Modèle d'analyse de cas d'utilisation « Logiciels»	39
Conclusion	40
CHAPITRE 3 Conception	41
Introduction	42
I. Conception générale	42
1. Architecture 3-tiers	42
2. Modèle en couches	43
3. Modèle MVC (Model, View, Controler)	44
II. Conception détaillée	45
1. Couche d'accès aux données	45
2. Couche métier	46
3. Couche présentation	46
III. Conception des classes	47
IV. Conception des cas d'utilisation de priorité 1	49



1. Conception de cas d'utilisation « s'authentifier »	49
V. Conception de cas d'utilisation de priorité 2	50
1. Conception de cas d'utilisation « Consultation»	50
2. Conception de cas d'utilisation « Afficher mémoire»	51
3. Conception de cas d'utilisation « Afficher interface»	51
4. Conception de cas d'utilisation « Afficher processus»	52
5. Conception de cas d'utilisation « Gérer les systèmes»	52
6. Conception de cas d'utilisation « Logiciels»	53
Conclusion	53
CHAPITRE 4 Réalisation	54
Introduction	55
I. Environnement et outils du travail	55
1. Configuration matérielle	55
2. Configuration logicielle	55
II. Présentation générale de l'application	56
1. Activer et configurer le protocole SNMP [6]	56
2. Principe de fonctionnement de l'application	61
2.1 Description de la partie A :	62
2.2 Description de la partie B :	62
d) Affichage et traitement des résultats de la recherche des informations	63
III. Description de l'application de la supervision :	63
1. Concept de sockets [7]	63
2. Adresses IP	64
3. Sockets TCP	65
3.1 Le modèle client/serveur	65
IV. Interface Homme-Machine	66
Conclusion	69
CONCLUSION GÉNÉRALE	70
REFERENCES	71
ANNEXE A	72



## Table des figures

Figure1 : Fonctionnalité de Nagios -----	10
Figure2 : Architecture de Nagios-----	11
Figure 3: Cycle de vie-----	17
Figure 4 Diagramme de cas d'utilisation global du système -----	24
Figure 5 : Description de cas d'utilisation « s'authentifie » -----	25
Figure 6: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse de cas « s'authentifier » -----	26
Figure 7: Diagramme de classe d'analyse de cas d'utilisation « S'authentifier »-----	26
Figure 8: Diagramme de collaboration de cas d'utilisation « S'authentifier »-----	27
Figure 9 : Description de cas d'utilisation « Afficher historique » -----	28
Figure 10 : Description de cas d'utilisation « Afficher mémoire» -----	28
Figure 11 : Description de cas d'utilisation « Afficher interface » -----	29
Figure 12 : Description de cas d'utilisation « Consulter liste des logiciels» -----	30
Figure 13 : Description de cas d'utilisation « Gérer les processus » -----	31
Figure 14 : Description de cas d'utilisation « Gérer le système» -----	31
Figure 15: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Consulter historique » -----	32
Figure 16: Diagramme de classe d'analyse de cas d'utilisation « Consulter historique»-----	32
Figure 17: Diagramme de collaboration de cas d'utilisation « Consulter historique» -----	33
Figure 18: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Afficher mémoire »-----	33
Figure 19: Diagramme de classe d'analyse de cas d'utilisation « Afficher mémoire»-----	34
Figure 20: Diagramme de collaboration de cas d'utilisation « Afficher mémoire» -----	34
Figure 21: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Afficher interfaces »-----	35
Figure 22: Diagramme de classe d'analyse de cas d'utilisation « Afficher interfaces» -----	35
Figure 23: Diagramme de collaboration de cas d'utilisation « Afficher interfaces» -----	36
Figure 24: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Afficher processus »-----	36
Figure 25: Diagramme de classe d'analyse de cas d'utilisation « Afficher processus» -----	37
Figure 26: Diagramme de collaboration de cas d'utilisation « Afficher processus» -----	37
Figure 27: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Gérer le système» -----	38



Figure 28: Diagramme de classe d'analyse de cas d'utilisation « Gérer le système» -----	38
Figure 29: Diagramme de collaboration de cas d'utilisation « Gérer le système»-----	39
Figure 30: Traçabilité entre le modèle de cas d'utilisation et le modèle d'analyse « Logiciels»-----	39
Figure 31: Diagramme de classe d'analyse de cas d'utilisation « Logiciels» -----	40
Figure 32: Diagramme de collaboration de cas d'utilisation « Logiciels»-----	40
Figure 33: Architecture 3-tiers-----	42
Figure 34: Architecture du modèle en couches -----	44
Figure 35: Modèle MVC 2 -----	45
Figure 36: Migration du modèle relationnel en modèle Objet -----	46
Figure 37: Diagramme de classe -----	48
Figure 38: Schéma relationnel de la base de données -----	49
Figure 39: Diagramme de séquence de cas « s'authentifier » -----	50
Figure 40: Diagramme de séquence de cas « Consultation» -----	50
Figure 41: Diagramme de séquence de cas « Afficher mémoire»-----	51
Figure 42: Diagramme de séquence de cas « Afficher interface réseau» -----	51
Figure 43: Diagramme de séquence de cas « Afficher processus»-----	52
Figure 44: Diagramme de séquence de cas « Gérer les systèmes»-----	52
Figure 45: Diagramme de séquence de cas « Logiciels» -----	53
Figure 46: Activation du protocole SNMP -----	57
Figure 47: Propriétés de service SNMP « Général »-----	58
Figure 48: Propriétés de service SNMP « Agent » -----	59
Figure 49: Propriétés de service SNMP « Sécurité »-----	61
Figure 50: schéma du principe de fonctionnement de l'application de la supervision -----	62
Figure 51 : La métaphore des prises -----	64
Figure 52 : Fonctionnement du serveur -----	65
Figure 53 : Fonctionnement du client -----	66
Figure 54: Interface d'authentification -----	67
Figure 55: Page principale du tableau de bord -----	67
Figure 56: Page logiciels -----	68
Figure 57 : Page interfaces -----	69