**UNIVERSITE SIDI MOHAMED BEN ABDELLAH**

**FACULTE DES SCIENCES ET TECHNIQUES FES**

**DEPARTEMENT D'INFORMATIQUE**

*Projet de Fin d'Etudes*

**Licence Sciences et Techniques Génie Informatique**

# *Développement d'une Application Web JEE Pour la supervision réseau*

*Lieu de stage :* **La Régie Autonome de Distribution d'Eau et d'Electricité de Fès.**

*Réalisé par :*

Khaoula LAHMAMSI

*Encadré par :*

Pr. Khalid ZENKOUAR

Mr. Zouhair BELAHBIB

**Soutenu le 11 /06/2016 devant le jury composé de :**

**Pr. K. Zenkouar**

**Pr. A. Boushaba**

**Pr. M. Ouzarf**

**Année Universitaire 2015-2016**

# Remerciements

Il est souvent difficile de remercier les gens qui vous aident à accomplir les tâches qui vous sont données, et Pourtant nous devons exprimer l'entière gratitude que nous ressentons envers eux.

Avant toute chose, je tiens à témoigner toute ma reconnaissance en premier à mon **DIEU**, mon créateur pour m'avoir donnée la force pour accomplir ce travail.

Je tiens à adresser mes remerciements à mon encadrant **Pr Khalid Zenkouar** d'avoir accepté de m'encadrer et de m'orienter durant l'élaboration de ce modeste travail. Vous m'avez apportée un grand soutien tout au long de mon stage.

Je tiens également à exprimer toute ma reconnaissance à Monsieur le Directeur de la R.A.D.E.E.F Mr. **LAKLALECH YOUSSEF** qui m'a accordée le privilège de passer mon stage au sein de la régie pour une durée de deux mois.

Je tiens à remercier vivement mon maitre de stage, **Mr BELAHBIB Zouhair,** analyste programmeur au sein du département système d'information, pour m'avoir encadrée tout au long de ce stage, pour le partage de son expertise au quotidien et sans hésiter à aucun moment de me consacrer une part de son temps précieux afin de m'aider considérablement dans la réalisation de ce travail, aussi d'être source d'information, de communication, d'encadrement et d'orientation technique et grâce à sa confiance j'ai pu m'accomplir dans mes missions.

Mes profondes gratitudes et mon profond respect aux membres du jury Pr K. Zenkouar, Pr A. Boushaba, et Pr M. Ouzarf, pour m'avoir honorée en acceptant d'évaluer et de juger ce travail.

Enfin, Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidée lors de la rédaction de ce rapport.

Merci
·BEAUCOUP·

# Dédicaces

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

*À mes très chers parents,*

*En témoignage de ma profonde gratitude, 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.*

*Aux Membres du Club Espoir*

*Je vous dédie mon travail en témoignage de mon sincère attachement.*

*Que ce modeste travail soit le gage de mon estime profonde et de ma vive reconnaissance.*

*Aussi à toute les personnes qui m'ont aidée de près ou de loin à la réalisation de ce travail*

# Sommaire

INTRODUCTION GENERALE ..... 9

Chapitre 1 : Contexte général du projet ..... 10

I. Structure d'accueil ..... 10

1. Présentation générale de la RADEEF ..... 10
2. Organisation de la RADEEF ..... 11
3. Département système d'information ..... 12

II. Présentation du projet ..... 12

1. Problématique ..... 13
2. Résultats attendus ..... 13
3. Cahier de charge ..... 13
4. Taches du projet ..... 14
5. Diagramme de GANTT ..... 15
6. Cycle de vie ..... 15
7. Objectifs ..... 16

Chapitre 2 : Modélisation et Conception ..... 17

I. Méthodologie de conception ..... 18

1. Processus unifié ..... 18
2. UML ..... 19

II. Conception générale ..... 19

1. Architecture 3-tiers ..... 19
2. Modèle en couches ..... 20
3. Modèle MVC (Model, View, Controler) ..... 21

III. Spécifications fonctionnelles ..... 22

1. Spécification des besoins fonctionnels ..... 22
2. Spécification des besoins non fonctionnels ..... 22

IV. Identification des acteurs ..... 22

1. Diagramme de package ..... 23
2. Diagramme de cas d'utilisation ..... 23
   a. L'administrateur ..... 24
   b. L'employé ..... 25
3. Description détaillée des cas d'utilisation ..... 25

V. Conception des classes ..... 27
1. Diagramme de classe ..... 27
2. Conception des cas d'utilisation ..... 28

Chapitre 3 : Réalisation ..... 32

I. Choix techniques ..... 33
1. Choix de langage de travail ..... 33
   a. HTML5 ..... 33
   b. CSS3 ..... 33
   c. JavaScript ..... 33
   d. XML ..... 33
   e. SVG ..... 34
   f. Java ..... 34
2. Outils de développement ..... 34
3. Plateforme de développement ..... 36
4. SGBD ..... 39
5. Serveur d'application ..... 40

II. Configuration ..... 40
1. Configuration matérielle ..... 40
2. Installation et configuration de GNS3 ..... 40

III. Interface-Homme-Machine ..... 46

CONCLUSION GENERALE ..... 51

REFERENCES ..... 52

ANNEXE ..... 53
1. Client-serveur ..... 53
2. Client léger ..... 54

# *Liste des figures*

*Figure 1 : Organigramme de Direction*

*Figure 2 : répartition des taches*

*Figure 3 : Diagramme de GANTT*

*Figure 4 : Cycle de vie*

*Figure 5 : Architecture 3-tiers.*

*Figure 6 : Architecture du modèle en couches.*

*Figure 7 : Modèle MVC.*

*Figure 8 : Diagramme de package des acteurs*

*Figure 9 : Diagramme des cas d'utilisations de l'administrateur*

*Figure 10 : Diagramme des cas d'utilisations d'observateur*

*Figure 11 : Diagramme de classe.*

*Figure 12 : Diagramme de séquence de cas « s'authentifier »*

*Figure 13 : Diagramme de séquence de cas « Consulter Archive de scan »*

*Figure 14 : Diagramme de séquence de cas « Consulter Interfaces »*

*Figure 15 : Diagramme de séquence de cas « Ajouter Equipement »*

*Figure 16 : caractéristiques du PC*

*Figure 17 : La page d'authentification*

*Figure 18 : L'accueil de l'administrateur*

*Figure 19 : La page de création d'un nouvel équipement*

*Figure 20 : La page de la liste des utilisateurs*

*Figure 21 : La page « Supprimer un Utilisateur »*

*Figure 22 : La page de l'archive de scan*

*Figure 23 : L'accueil de l'observateur*

*Figure 24 : la page « Consulter Interfaces »*

# *Liste des tableaux*

***Table 1 :*** ***Description détaillée du cas « Authentification »***

***Table 2 :*** ***Description détaillée du cas « Liste Utilisateurs »***

***Table 3 :*** ***Description détaillée du cas « Ajouter Equipement »***

***Table 4 :*** ***Description détaillée du cas « Consulter Archive de scan »***

***Table 5 :*** ***Tableau des méthodes de la classe InetAddress.***

# INTRODUCTION GENERALE

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.

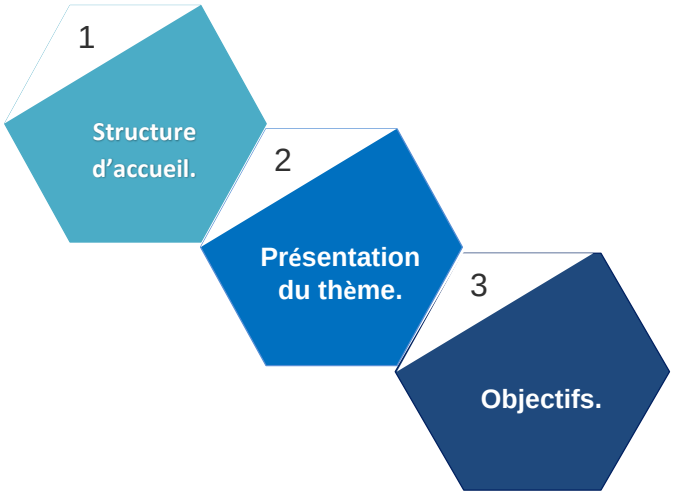
C'est pourquoi, les administrateurs réseaux font appel à des solutions de surveillance et de supervision de réseaux qui sont l'une des disciplines modernes qui connaissent actuellement le plus fort essor. La supervision de réseaux peut être définie comme l'utilisation de ressources réseaux adaptées dans 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 mon projet de fin d'études qui consiste à développer une application JEE performante et facile à utiliser qui permet la supervision des différentes interfaces des équipements réseaux au sein de la société RADEEF. Sans l'utilisation d'aucun outil de supervision (Nagios, Zabbix, Groundwork, ….).

Ce rapport va s'articuler autour de trois chapitres. Il fait la synthèse de mon étude durant la période de stage. Dans le premier chapitre, je vais mettre le point sur une présentation générale de l'entreprise et du projet. Le second chapitre est consacré à la conception et la modélisation de l'application le dernier chapitre sera dédié à la réalisation de mon application.

# Chapitre 1 :

# Contexte générale du projet



## Introduction

Dans ce chapitre, je vais présenter le cadre général de mon projet. En effet, je vais commencer par la présentation de l'organisme d'accueil, ensuite la présentation du projet, l'étude de l'existant de mon projet et son cycle de vie.

# I. Structure d'accueil

## 1. Présentation générale de la RADEEF

La Régie Autonome intercommunale de Distribution d'Eau et d'Electricité de la wilaya de Fès (RADEEF) est un établissement public à caractère industriel et commercial, doté de la personnalité morale et de l'autonomie financière, placé sous la tutelle du Ministère de l'Intérieur.

La RADEEF a été créée par délibération du conseil municipal de la ville de Fès en date du 30 avril et 29 août 1969 en vertu du Dahir n° 1.59.315 du 23 Juin 1960 relatif à l'Organisation communale, et ce après l'expiration du contrat de concession dont bénéficiait la Compagnie Fassie d'Electricité (CFE) au titre de la distribution de l'énergie électrique.

Par arrêté du 25 Décembre 1969, le Ministre de l'Intérieur a approuvé la délibération du conseil communal de la ville de Fès en date du 29 Août 1969 concernant la création de la RADEEF, fixant la dotation initiale établissant son règlement intérieur ainsi que son cahier des charges.

En Janvier 1970, la RADEEF s'est substituée, d'une part à la « Compagnie Fassie d'Electricité » pour la gestion du réseau électrique, et d'autre part à la ville de Fès pour la gestion du réseau d'eau potable.

La dotation en capital de la Régie, à sa création, fut constituée par l'apport initial auquel se sont ajoutés la valeur des installations, du matériel et du stock remis par la ville ainsi que les fonds détenus pour le compte de celle-ci par l'ancien concessionnaire.

Par la suite, la RADEEF a été transformée en Régie Intercommunale suite à l'arrêté du Ministre de l'Intérieur n°3211 du 02-10-1985 portant autorisation de créer le nouveau syndicat des communes pour la gestion du Service de l'Eau potable dans 19 communes.

A compter du 1er Janvier 1996, la RADEEF a été chargée de la gestion du réseau d'assainissement liquide de la ville de Fès en vertu de l'arrêté du Ministre de l'Intérieur n° 2806-95 du 3 Juin 1996 approuvant les délibérations du conseil de la Communauté Urbaine de Fès et des conseils communaux relevant de cette communauté, lesquelles délibérations ont chargé la RADEEF de la gestion du réseau d'assainissement liquide de la ville de Fès.

Actuellement, la **RADEEF** assure la distribution de l'eau et de l'électricité ainsi que la gestion du réseau d'assainissement liquide l'intérieur de la ville de Fès et de la commune Ain Chkef. Elle est en outre chargée de la distribution de l'eau potable dans les communes urbaines de Sefrou et Bhalil ainsi que dans les communes rurales suivantes : Bir Tam-Tam, Ras Tabouda, Sidi Harazem, Ain Timgnai, Ouled Tayeb, Douar Ait Taleb et Douar Ait El Kadi.

## 2. Organisation de la RADEEF

Il existe à la RADEEF plusieurs divisions et services représentés sous forme d'organigramme :

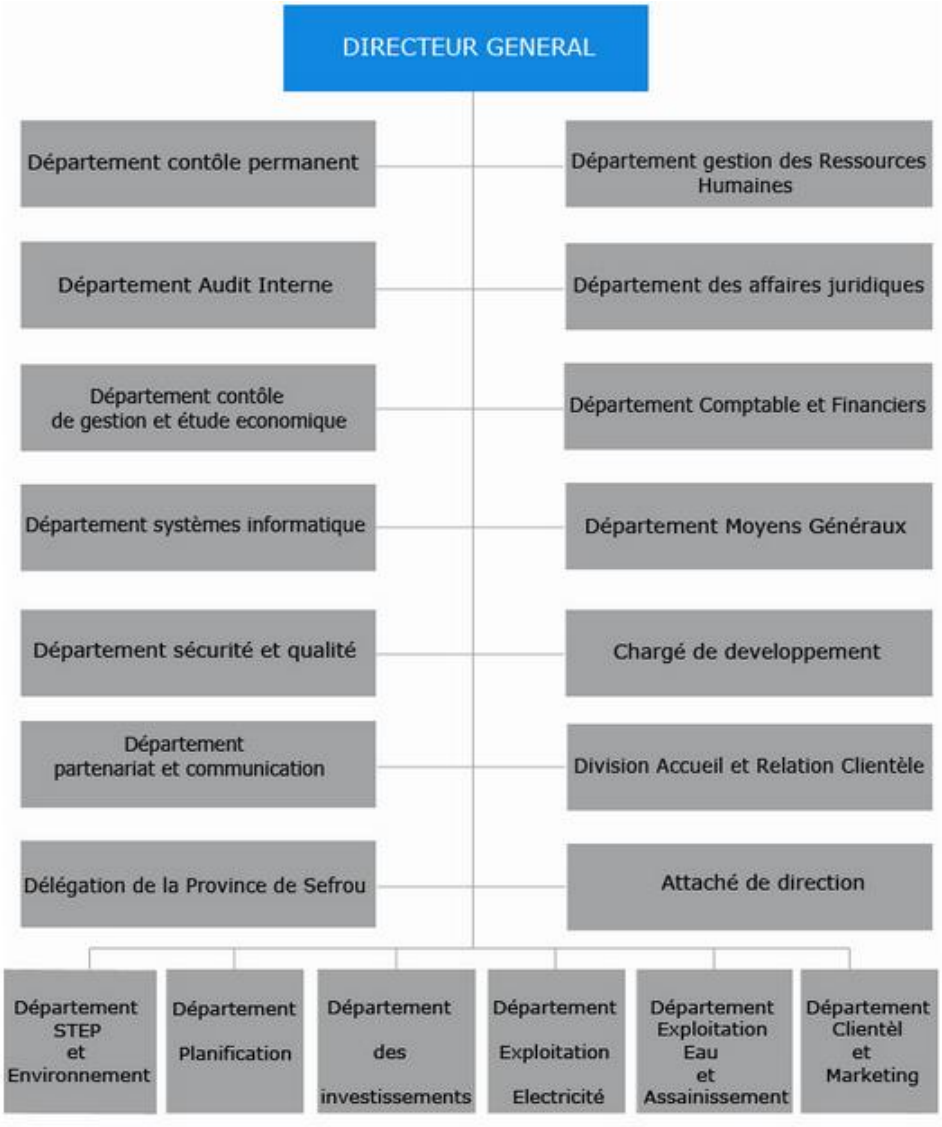



***Figure 1 :*** **Organigramme de Direction**

## 3. Département système d'information

Le département système d'information se compose d'une :

- **Division ATRS** : Architecture Technique des Réseaux et Systèmes

A pour mission de maintenir une connectivité permanente entre le site central et les sites distants et ce en assurant l'installation et la configuration des équipements réseau tels que les routeurs, les commutateurs, les pare-feux … d'une part et le suivi de la performance des contrôleurs principaux de domaine et des supports de communication d'autre part.

Chaque agence de la RADEEF se compose d'une salle système qui dispose de :

-un **système de climatisation** qui palie aux risques réchauffement des serveurs de production.

- **Coffrets réseau /baie de brassage** : comprend généralement des dispositifs tels que : des commutateurs Ethernet a terminaisons en fibre optique (port :LC, SC), des panneaux de brassage (connecteurs : RJ45), un routeur, un serveur.

A l'intérieur de l'agence, tous ces équipements sont reliés à travers des câbles FTP cat.6.
La communication avec les autres sites distants est assurée par une liaison principal de type radio fréquence, en cas de panne de cette dernière une ligne ADSL dite de secours prend la relève.

Pour les équipements et les connexions **de la RADEEF**,

- **Equipements** réseaux :
  - Routeurs CISCO
  - Switches Ethernet CISCO
  - Serveurs IBM
- **Connexions** réseau :
  - La RADEEF a pour type de connexions un VPN monté sur ;
    1. Liaison Onde Radio
    2. Liaison ADSL

# II. Présentation du projet

Les systèmes d'information sont tous différents 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, cette charge mène à penser à la création des applications pour leur faciliter la tâche.

## 1. Problématique

Au sein de la RADEEF précisément dans le département système d'information, qui a pour rôle principal la supervision du réseau de 24 agences situées dans la région Fès Séfrou. Le but de ce projet est la surveillance du bon fonctionnement des équipements réseau (les routeurs, les switches, les serveurs ...) des différentes agences.

L'administrateur chargé du bon fonctionnement du réseau, devra scanner un par un tous les postes, ainsi la durée de la réclamation de dysfonctionnement demandera plus de temps, sans oublier le risque que l'un des postes précédemment vérifié tombe en panne. Le responsable de la maintenance sera amené à examiner chaque interface afin de déterminer l'interface déficiente et cela conduira certainement à une perte de temps et nécessitera un grand effort et des coûts supplémentaires.

Ainsi, une application conviviale que j'ai nommée GOZETIM, servira comme un outil de la supervision et déclenchera une intervention immédiate des administrateurs et pourra remédier aux problèmes mentionnés auparavant.

## 2. Résultats attendus

Au regard des difficultés ci-dessus mentionnées, l'intérêt de cette application est de garantir un traitement automatisé du processus de la supervision et de gagner un temps très important et rendre le travail plus précis.

L'idée est de rendre la méthode de travail classique et statique qui consiste à exécuter la commande Ping pour vérifier le fonctionnement de chaque interface par une application qui donnera l'état de chaque interface (up, down) de tous les équipements réseau et d'une manière automatique en une autre dynamique.

## 3. Cahier de charge

Mon application GOZETIM va superviser une partie de l'infrastructure réseau de la RADEEF représentée par deux sites :

- RADEEF SIEGE
- RADEEF SIDI BRAHIM

Les acteurs nécessitent une authentification avant d'accéder aux services de l'application.

1) Les taches dédiées à l 'administrateur / l'acteur principal :

- La gestion de tous les utilisateurs (mise à jour des comptes).
- La consultation de la liste des équipements, aussi l'ajout d'un nouvel équipement.
- La consultation de la liste des interfaces de chaque équipement, en pouvant effectuer l'ajout et le changement de celle-ci.
- La consultation de la liste des sites, avec la possibilité d'ajouter un nouveau site.
- La gestion des alertes.
- Consulter l'archive de scan.
- La gestion du système de l'application.

1) Les taches dédiées à l'employé / l'acteur secondaire :

- La consultation de la liste des équipements.
- La consultation de la liste des interfaces de chaque équipement.
- La consultation de la liste des sites.
- Alerter dans le cas des pannes (par : SMS, Email, …).

## 4. Taches du projet

GANTT project

| Nom | Date de début | Date de fin |
|---|---|---|
| Etude préliminaire | 06/04/16 | 15/04/16 |
| Analyse et conception | 16/04/16 | 29/04/16 |
| Recherche et documentation | 11/04/16 | 10/05/16 |
| Développement et Réalisation | 30/04/16 | 28/05/16 |
| Déploiement et Maintenance | 29/05/16 | 02/06/16 |
| Cloture | 03/06/16 | 03/06/16 |

***Figure 2 :*** **répartition des taches**

## 5. Diagramme de GANTT

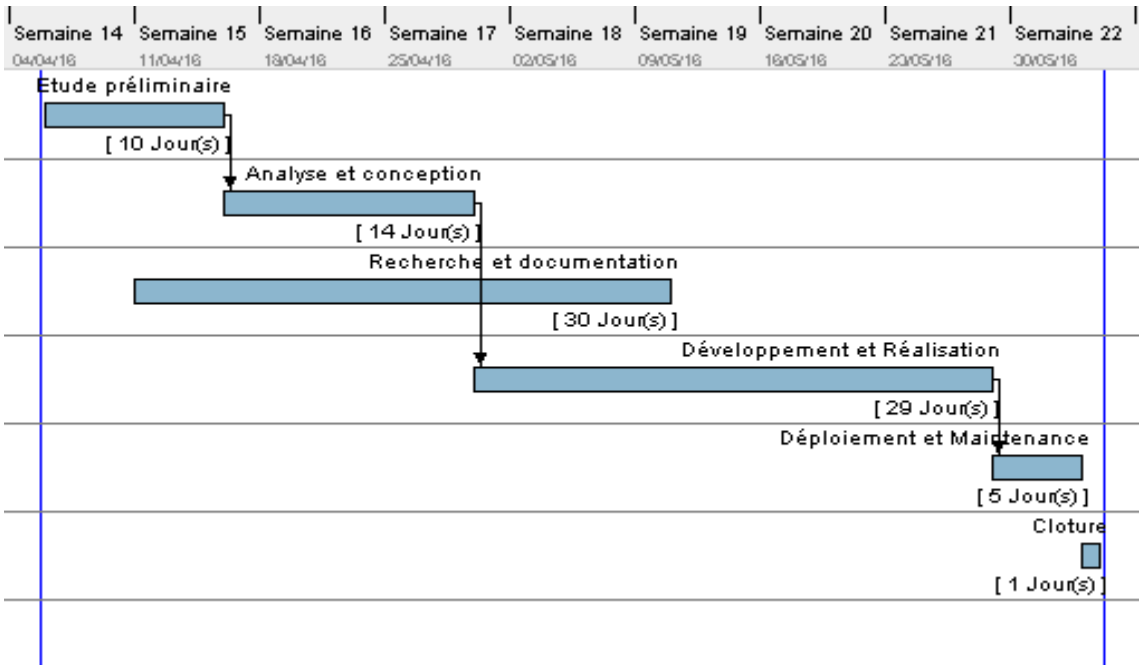



***Figure 3 :*** **Diagramme de GANTT**

## 6. 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 logiciels 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.

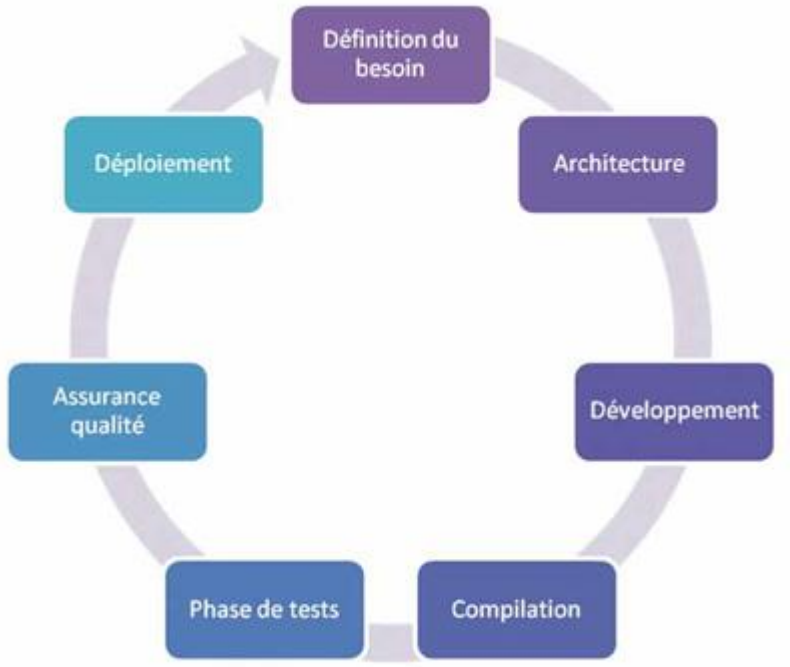



*Figure 4 :* **Cycle de vie**

## 7. Objectifs

Mon projet consiste à développer une application web « client léger » qui permet la supervision réseau par laquelle on obtiendra un point de centralisation pour la gestion de l'ensemble du System Informatique.

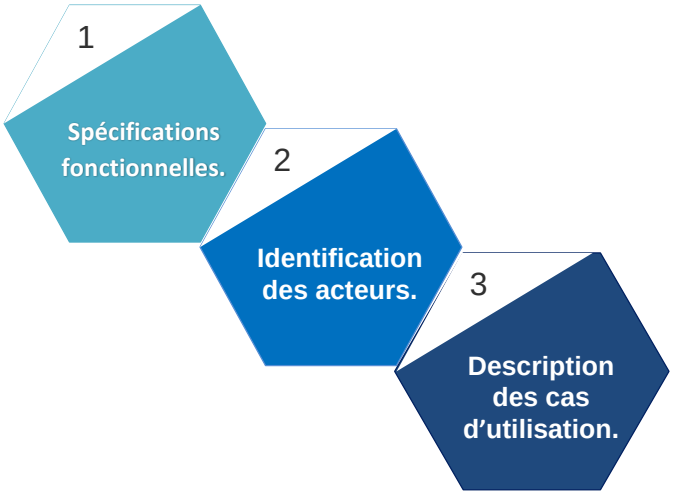
Cette application doit être une solution modulaire simple, réactive, proactive, pertinente et intuitive de façon à ne nécessiter aucun apprentissage particulier, sa maintenance et sa mise à jour doivent être faciles.

**Conclusion**

A travers ce 1er chapitre, j'ai pu me placer dans le contexte général de mon projet en introduisant le domaine de l'application ainsi que le cadre du travail.

# Chapitre 2 :

## MODELISATION ET CONCEPTION



**Introduction**

Ce chapitre sera dédié pour l'identification des acteurs, la conception architecturale de mon application et pour la conception détaillée des cas d'utilisation.

# I. 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

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.

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 d'A à Z
- Gérer le risque
- Obtenir de manière répétitive des produits de qualité constante.

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

# II. Conception générale

## 1. Architecture 3-tiers

Mon 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 5**) est un exemple explicatif :

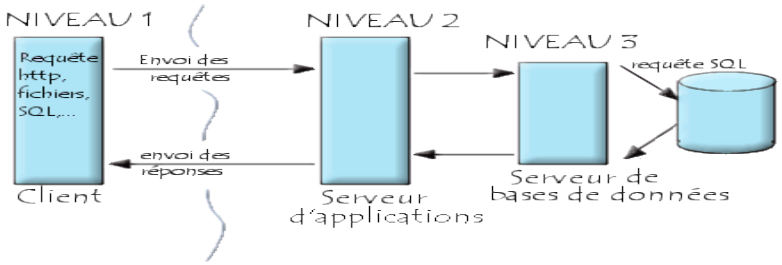



***Figure 5 :*** **Architecture 3-tiers.**

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 6**) représente une application qui respect ce modèle :

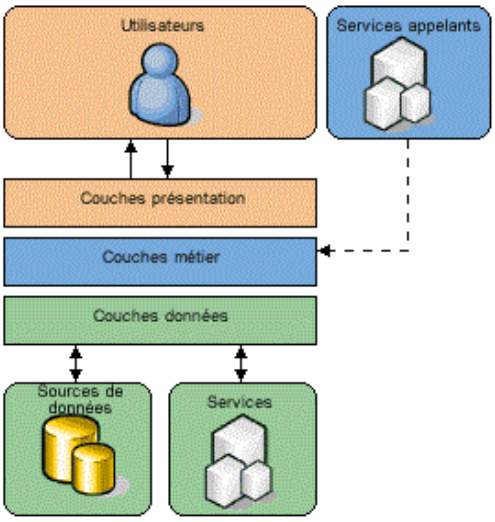



*Figure 6 :* **Architecture du modèle en couches.**

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

Mon 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.

Pour cela j'ai choisi le modèle de conception qui satisfait mon besoin et qui est le MVC

La figure suivante (**Figure7**) illustre le principe du modèle MVC :

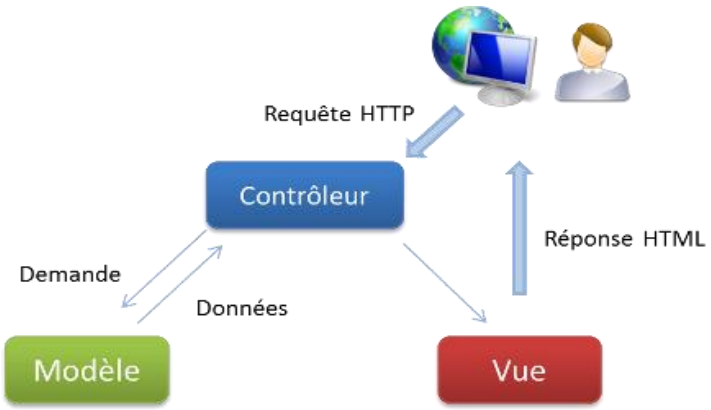



*Figure 7 :* **Modèle MVC.**

# III. Spécifications fonctionnelles

## 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.
- ✓ Supervision / Mise à jour de l'architecture réseau.
- ✓ Gestion des profils des différents acteurs.

## 2. 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 mon 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 mon application l'accès aux informations personnelles n'est autorisé qu'aux personnes propriétaires et selon un privilège qui détermine les droits d'accès.

# IV. Identification des acteurs

Un acteur est un humain ou une machine ne faisant pas partie de la solution à réaliser mais qui participe au fonctionnement général de la solution par une interaction.

Les acteurs du futur système sont :

./ L'administrateur : Il est responsable de la gestion des utilisateurs ainsi que de tous les paramétrages de l'application.

./ L'Observateur : parmi ces taches principales consulter l'architecture réseau avec possibilité de réclamer les pannes.

Types d'observateur :

- Le directeur général
- Les chefs d'agences

- L'équipe du département Système d'Information

## 1. Diagramme de package

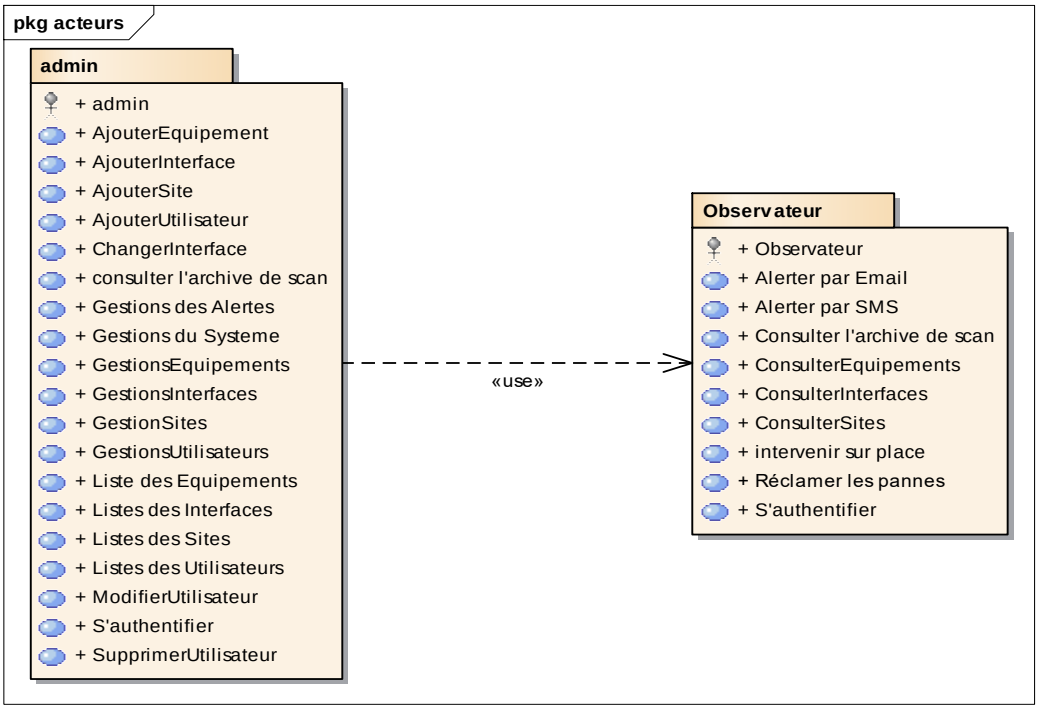



***Figure 8 :*** **Diagramme de package des acteurs**

## 2. Diagramme de cas d'utilisation

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.

a. L'administrateur

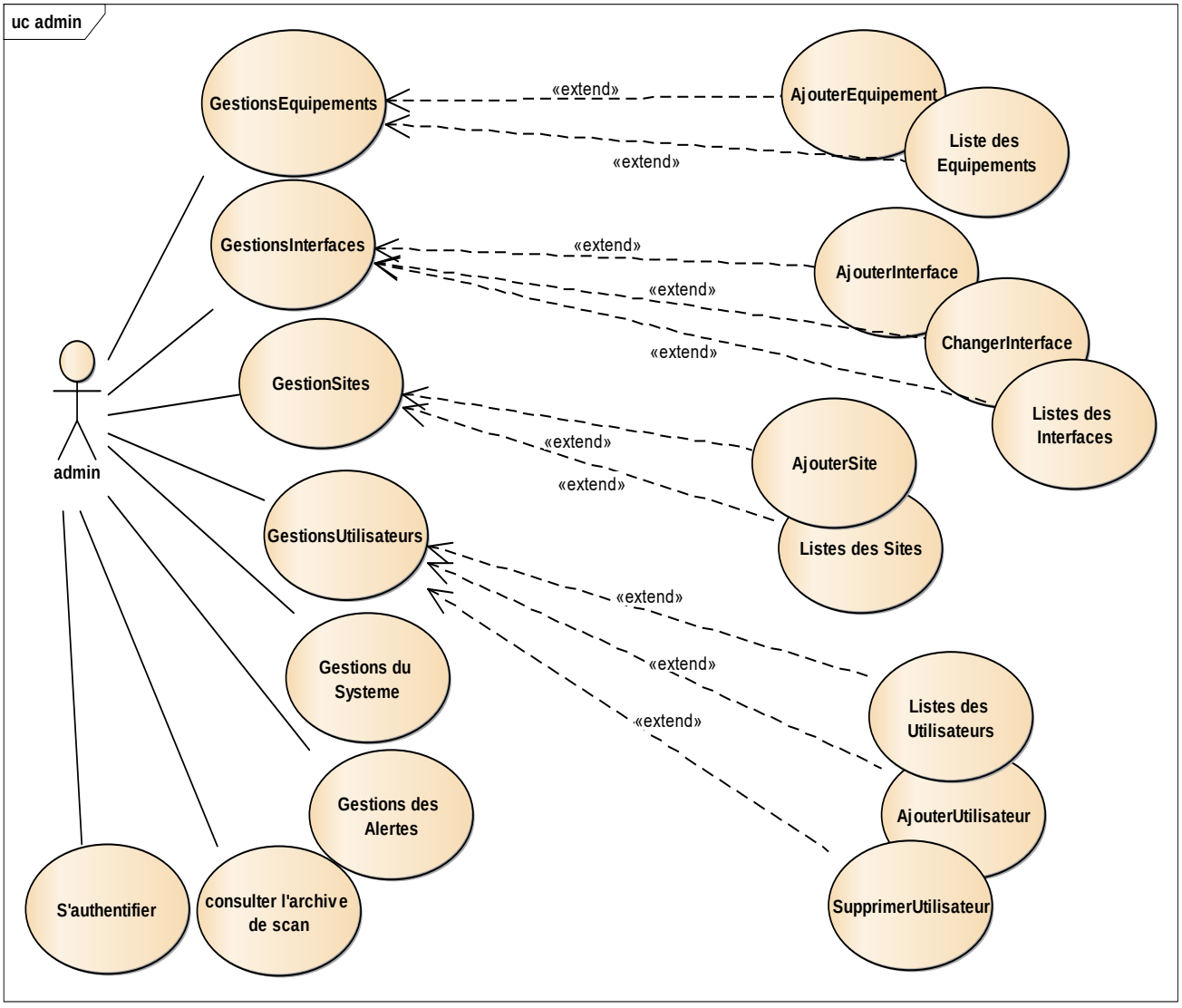



***Figure 9 :*** **Diagramme des cas d'utilisations de l'administrateur**

b. L'employé

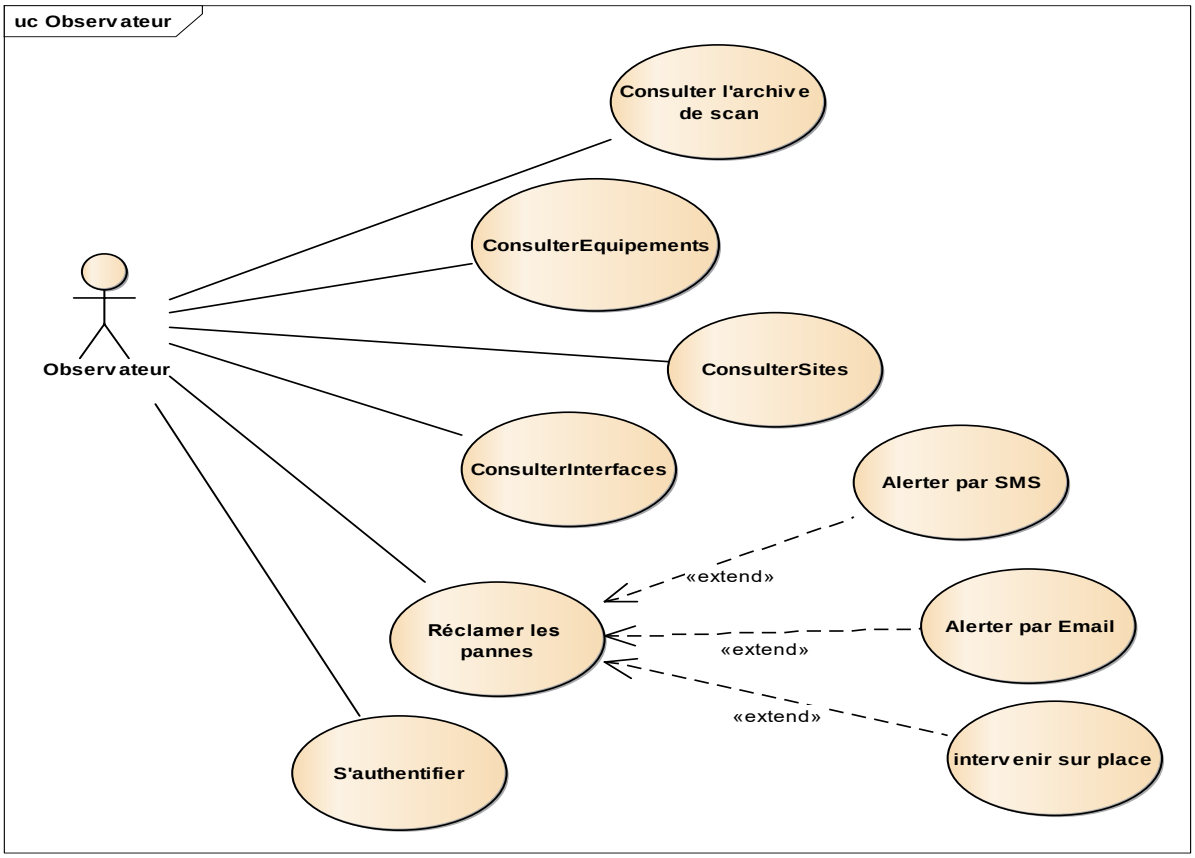



*Figure 10 :* **Diagramme des cas d'utilisations d'observateur**

## 3. Description détaillée des cas d'utilisation

- **Authentification :**

| **Nom du cas** | **« S'authentifier »** |
|---|---|
| **Acteur principal** | L'administrateur & l'employé |
| **Objectifs** | Accéder aux services de l'application |
| **Pré condition** | Aucune |
| **Scénario normal** | 1. affichage de la page de connexion.<br>2.L'utilisateur saisit son identifiant et son mot de passe.<br>3. vérification de la validité des informations fournies<br>4.Le système donne l'accès aux services. |
| **Scénario d'échec** | -si les champs vides : le système affiche des messages de renseignement |

| | |
|---|---|
| | -si les champs sont invalides : affichage de la page de connexion. |
| **Post-condition** | Authentification validée et succès d'accès. |

***Table 1 :*** **Description détaillée du cas « Authentification »**

- **Gestions Utilisateurs :**

| **Nom du cas** | **« Liste Utilisateurs »** |
|---|---|
| **Acteur principal** | L'administrateur |
| **Objectifs** | Consulter la liste des utilisateurs existants, afin d'ajouter ou supprimer un utilisateur. |
| **Pré condition** | Authentification |
| **Scénario** | 1.L'administrateur choisit le menu « liste utilisateurs ».<br>2.Le Système affiche la liste. |
| **Post-condition** | Liste des Utilisateurs affiché. |

***Table 2 :*** **Description détaillée du cas « Liste Utilisateurs »**

- **Gestion Equipements :**

| **Nom du cas** | **« Ajouter Equipement »** |
|---|---|
| **Acteur principal** | L'administrateur |
| **Objectifs** | Créer un nouvel équipement. |
| **Pré condition** | Gestion Equipements |
| **Scénario normal** | 1. Le système affiche le formulaire de création d'équipement.<br>2. L'administrateur saisit les informations nécessaires (type, marque, libelle)<br>3. L'administrateur valide<br>4. Le système vérifie les champs<br>5. Le système enregistre l'équipement |
| **Scénario d'échec** | -saisir les informations d'équipement<br>-cas d'erreur : champs non remplis, erreur de validation<br>→Affichage d'un message de renseignement. |
| **Post-condition** | Nouvel équipement enregistré. |

***Table 3 :*** **Description détaillée du cas « Ajouter Equipement »**

- **Archive de Scan :**

| **Nom du cas** | **« Consulter l'Archive de Scan »** |
|---|---|
| **Acteur principal** | L'administrateur et l'observateur |
| **Objectifs** | Consulter l'historique du scan de chaque interface. |
| **Pré condition** | Authentification |
| **Scénario** | 1.L'administrateur choisit le menu « Archive de Scan ».<br>2.Le Système affiche l'archive. |
| **Post-condition** | Archive de scan affiché. |

***Table 4 :*** **Description détaillée du cas « Consulter Archive de scan »**

# V. 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.

## 1. Diagramme de classe

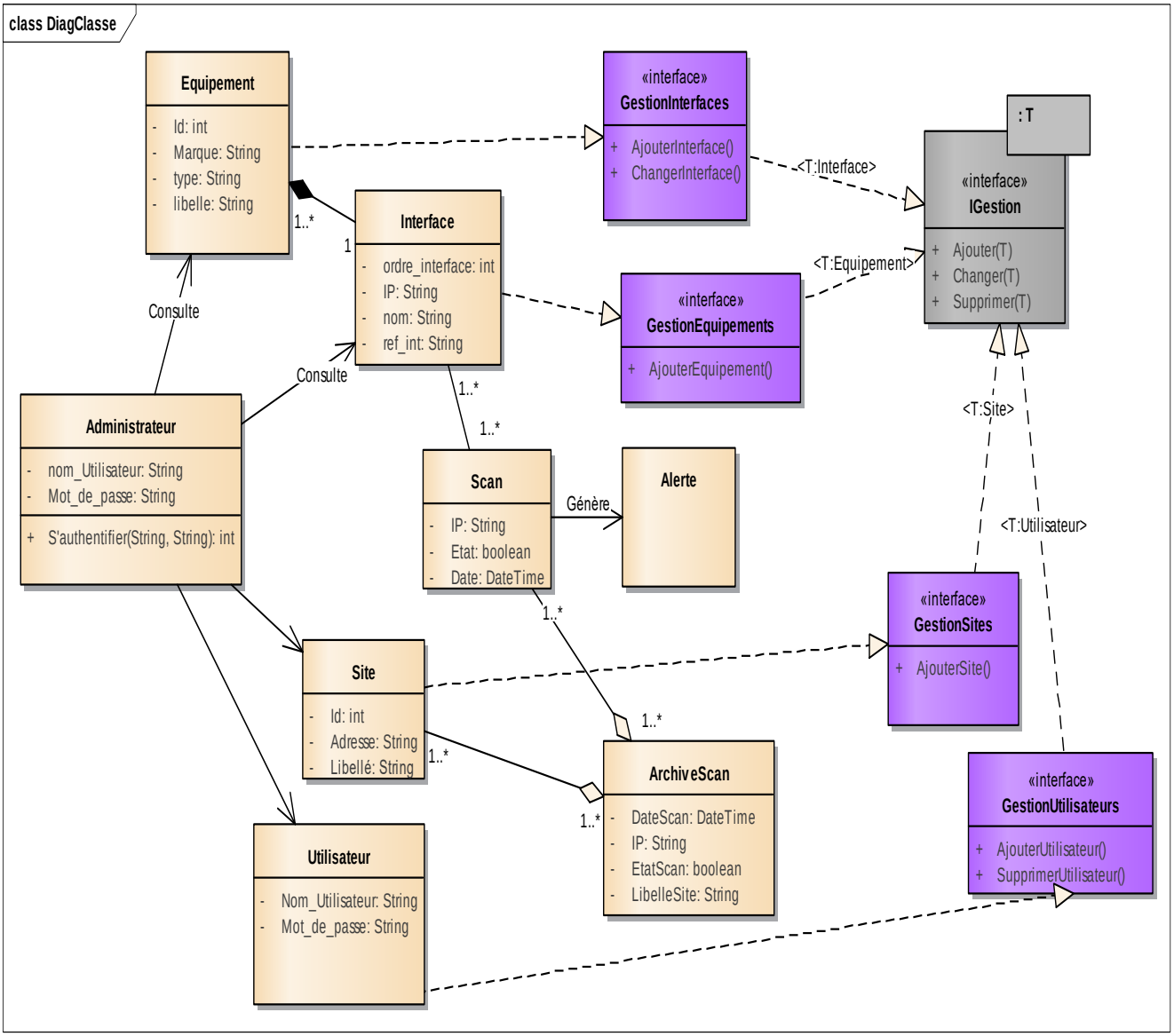



*Figure 11 :* **Diagramme de classe.**

## 2. Conception des cas d'utilisation

- **Conception de cas d'utilisation « s'authentifier » :**

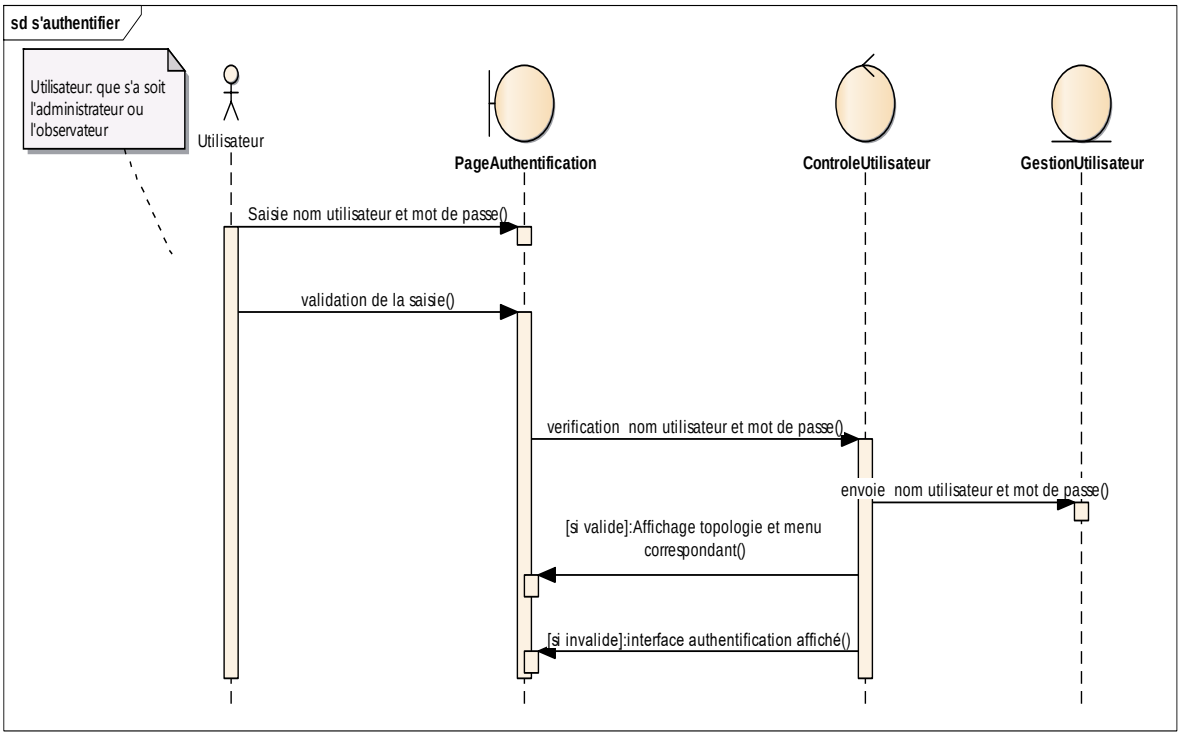



*Figure 12 :* **Diagramme de séquence de cas « s'authentifier »**

### Conception de cas d'utilisation « Consulter Archive de scan » :

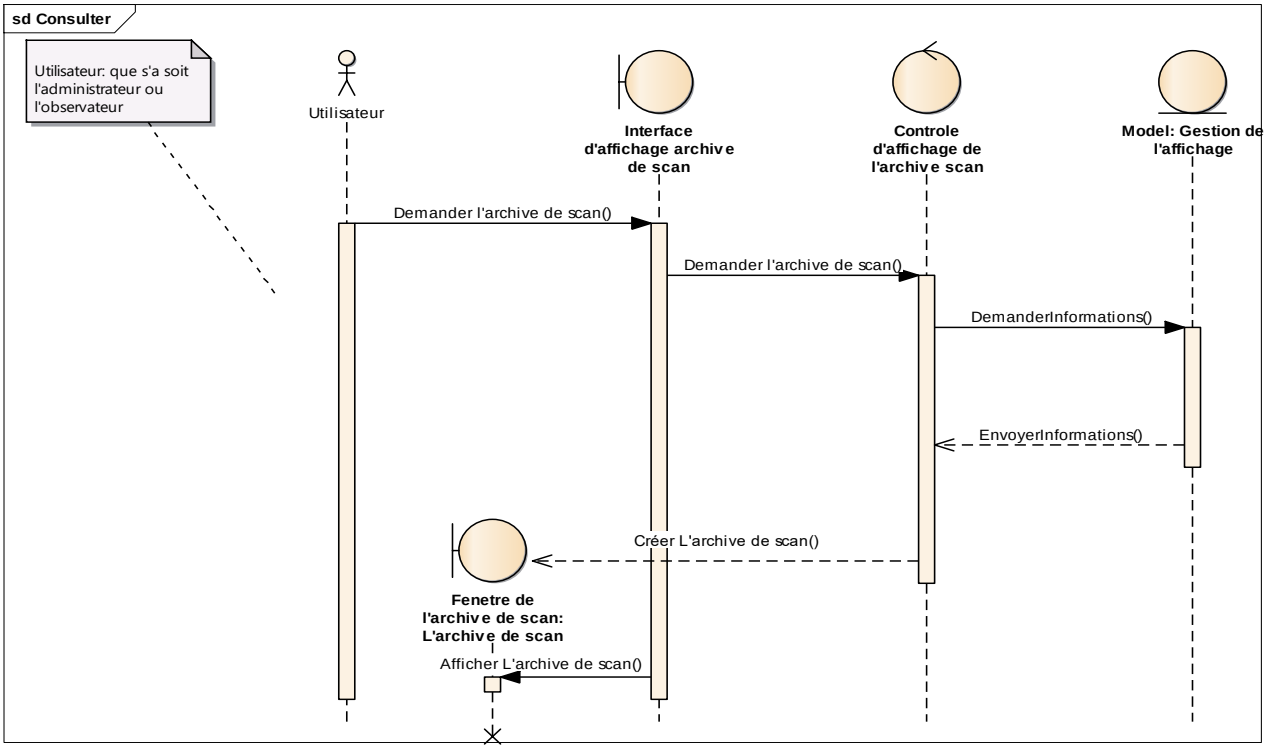



*Figure 13 :* **Diagramme de séquence de cas « Consulter Archive de scan »**

- **Conception de cas d'utilisation « Consulter Interfaces » :**

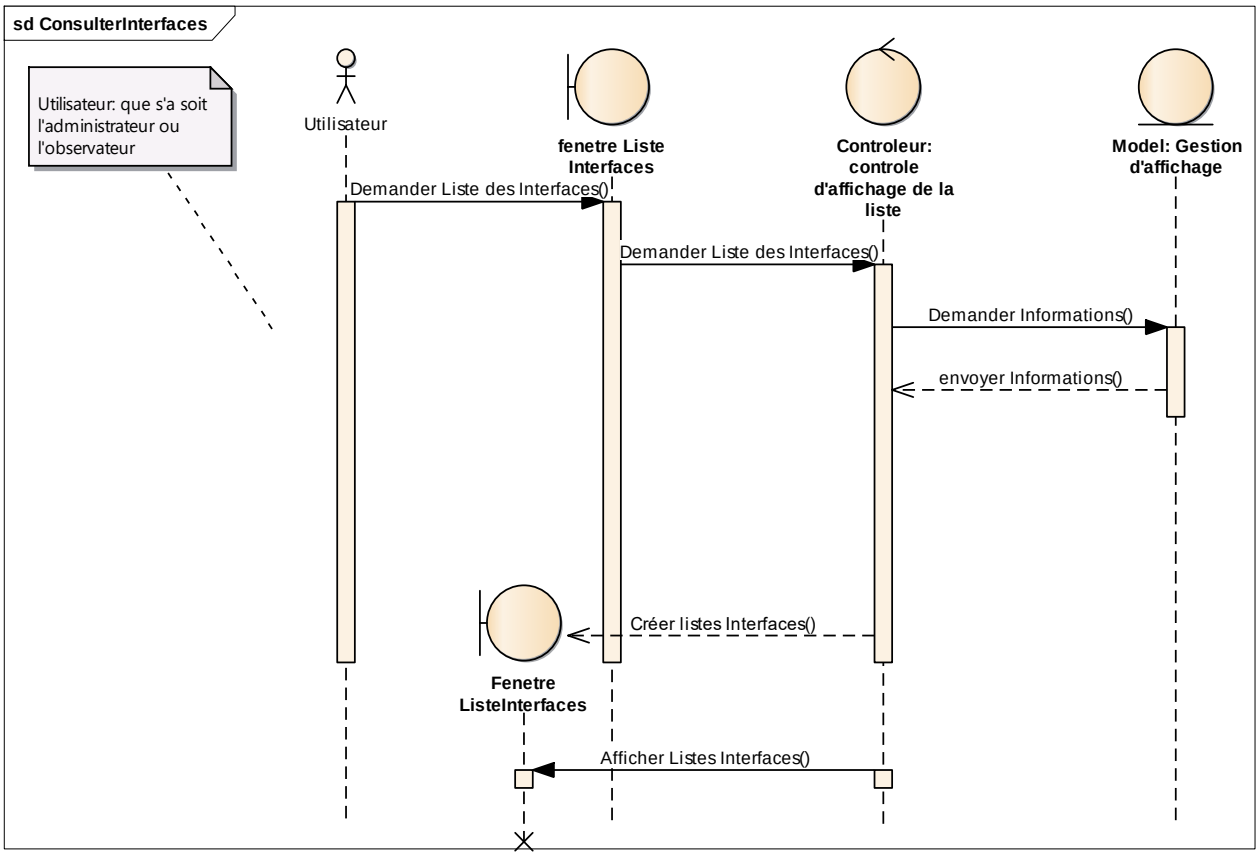



***Figure 14 :*** **Diagramme de séquence de cas « Consulter Interfaces »**

- **Conception de cas d'utilisation « Ajouter Equipement » :**

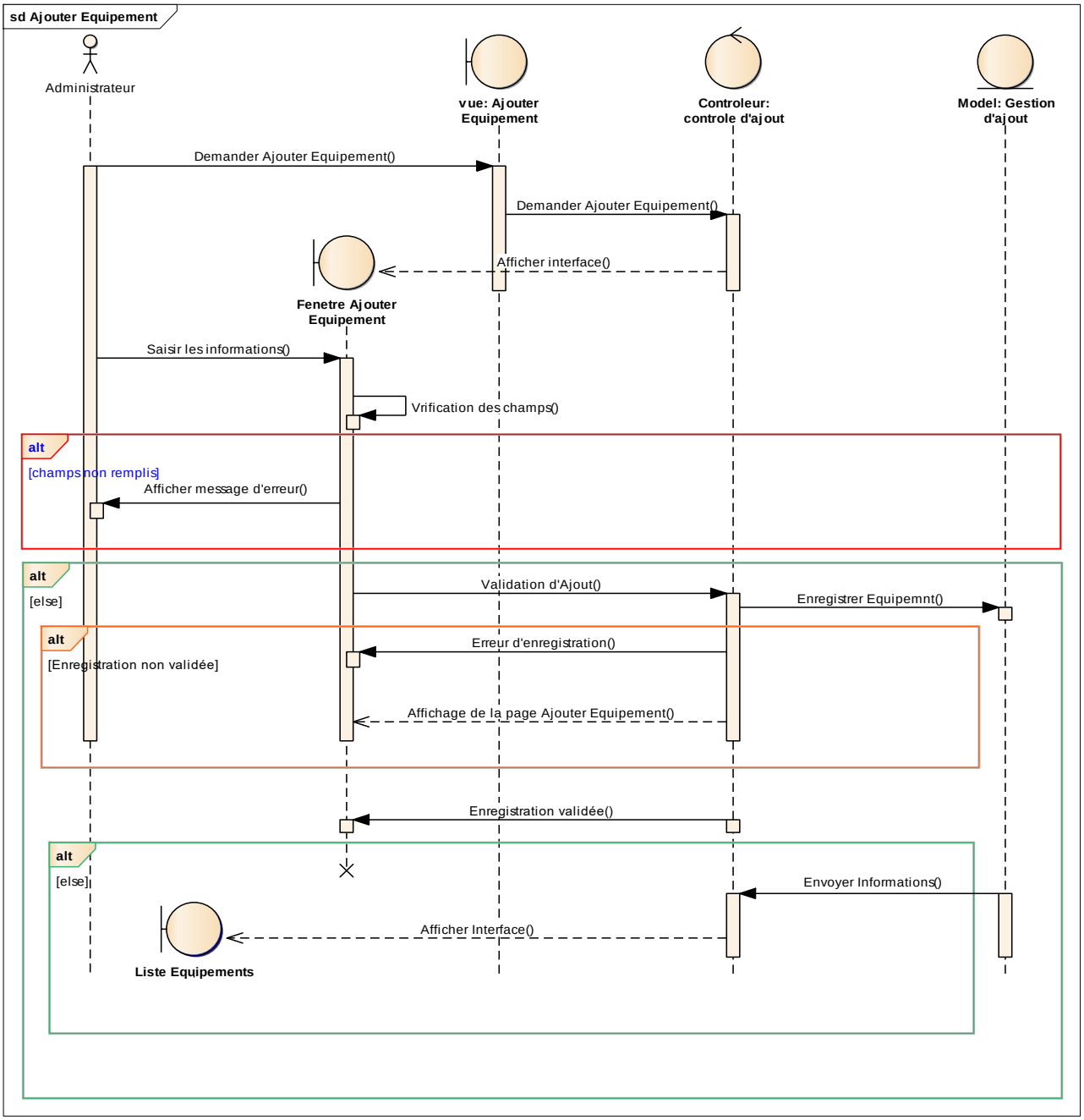
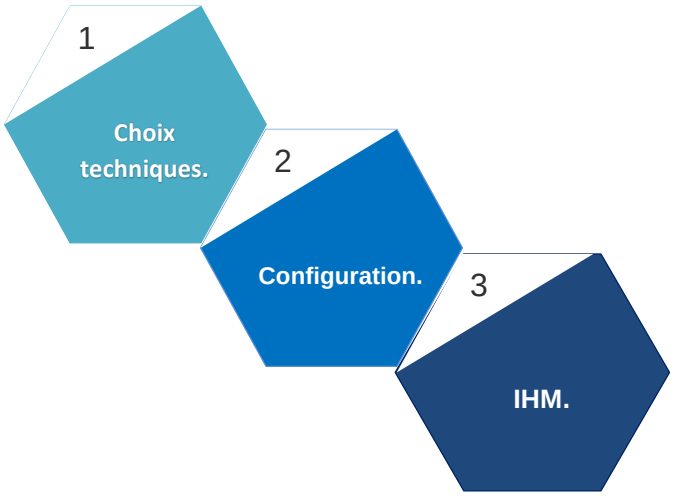



***Figure 15 :*** **Diagramme de séquence de cas « Ajouter Equipement »**

## Conclusion

A travers ce chapitre, j'ai présenté ma conception de l'application. J'ai fourni, dans un premier temps, une conception globale à travers un schéma général décrivant l'organisation de mon système. Ensuite, j'ai présenté la conception détaillée de l'application à travers les diagrammes de séquence. A présent, je suis capable d'entamer la partie réalisation.

# Chapitre 3 :
# Réalisation



**Introduction**

Dans cette partie, il est question de décrire l'aspect d'implémentation .je commencerai alors par présenter les choix technologiques. Puis, je spécifierai l'environnement matériel et logiciel supportant mon application. Ensuite, je présenterai mon application.

# I. Choix techniques

## 1. Choix de langage de travail

### a. HTML5

**HTML5** (HyperText Markup Language 5) est un langage dit de « marquage » (de « balisage ») dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents. La version 5.0 du langage HTML définit deux syntaxes de DOM : HTML5 et XHTML5. Cette version apporte de nouvelles possibilités en terme de création d'« applications Web riches » bénéficiant de l'intégration d'éléments multimédias et d'interactivité.

### b. CSS3

Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

### c. JavaScript

**JavaScript** est un langage de script orienté objet principalement utilisé dans les pages HTML. A l'opposé des langages serveurs (qui s'exécutent sur le site), JavaScript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions (lors du passage de la souris au-dessus d'un élément, du redimensionnement de la page...).

### d. XML

Le **XML**, acronyme d'eXtensible Markup Language (qui signifie : langage de balisage extensible), est un langage informatique qui sert à enregistrer des données textuelles. Ce langage, grosso-modo similaire à l'HTML de par son système de balisage, permet de faciliter l'échange d'information sur l'internet. Contrairement à l'HTML qui présente un nombre finit de balises, le XML donne la possibilité de créer de nouvelles balises à volonté.

### e. SVG

**SVG** est un langage XML utilisé pour décrire des graphiques en 2 dimensions, c'est un langage bénéficiant d'une recommandation du W3C. **SVG** permet de décrire des images riches avec des caractéristiques tel que des inclinaisons, de la transparence, des effets de filtres et des animations.

### f. Java

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

## Les concepts utilisés :

- *DOM* : Le **Document Object Model** est un standard du W3C qui décrit une interface indépendante de tout langage de programmation et de toute plate-forme, permettant à des programmes informatiques et à des scripts d'accéder ou de mettre à jour le contenu, la structure ou le style de documents HTML et XML. Le document peut ensuite être traité et les résultats de ces traitements peuvent être réincorporés dans le document tel qu'il sera présenté.
- *AJAX :* L'architecture informatique **Ajax** (acronyme d'***Asynchronous JavaScript and XML***) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005. Ajax combine JavaScript, les CSS, JSON, XML, le DOM et le XMLHttpRequest afin d'améliorer maniabilité et confort d'utilisation des applications internet riches.

# 2. Outils de développement

- L'**environnement d'exécution Java** (abr. ***JRE*** pour *Java Runtime Environment*), parfois nommé simplement « Java », est une famille de logiciels qui permet l'exécution des programmes écrits en langage de programmation Java. Le JRE se compose d'une machine virtuelle, de bibliothèques logicielles utilisées par les programmes Java et d'un plugin pour permettre l'exécution de ces programmes depuis les navigateurs web.

- Le **Java Development Kit** (**JDK**) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java.

**Eclipse** est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java**.**

Entreprise Architecte un outil d'analyse et de création UML, couvrant le développement de logiciels du rassemblement d'exigences, en passant par les étapes d'analyse, les modèles de conception et les étapes de test et d'entretien. Cet outil graphique basé sur Windows, peut être utilisé par plusieurs personnes et conçu pour vous aider à construire des logiciels faciles à mettre jour. Il comprend un outil de production de documentation souple et de haute qualité.

**Microsoft Visio** (officiellement Microsoft office Visio) est un logiciel de diagrammes et de synoptiques pour Windows qui fait partie de la suite bureautique Microsoft Office, permet de faire des diagrammes plus avancés, grâce à des modèles supplémentaires, parmi ses modèles Le modèle Diagramme de réseaux simples contient des formes standards pour serveurs, ordinateurs et d'autres parties de votre réseau pour planifier un réseau que vous envisagez de créer.

Notepad++ est un éditeur de code source qui prend en charge plusieurs langages. Ce programme, codé en C++ avec STL et win32 api, a pour vocation de fournir un éditeur de code source de taille réduite mais très performant. En optimisant de nombreuses fonctions tout en conservant une facilité d'utilisation et une certaine convivialité, Notepad++ contribue à la limitation des émissions de dioxyde de carbone dans le monde : en effet, en réduisant l'utilisation de CPU, la consommation d'énergie des ordinateurs chute considérablement, en conséquence de quoi, la terre est plus verte.

***KompoZer*** est un éditeur HTML WYSIWYG basé sur Nvu, c'est-à-dire un logiciel permettant de créer des pages web (utilisant le code HTML) et en voyant le résultat final.

**Photoshop** est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par Adobe. Il est principalement utilisé pour le traitement de photographies numériques, mais sert également à la création d'images *ex nihilo*. Il travaille essentiellement sur images matricielles car les images sont constituées d'une grille de points appelés pixels. L'intérêt de ces images est de reproduire des gradations subtiles de couleurs.

**GNS3** est un simulateur d'équipements Cisco. Cet outil permet donc de charger de véritable IOS Cisco et de les utiliser en simulation complète sur un simple ordinateur. Pour caricutariser, **GNS3** permet d'avoir un routeur Cisco virtuel sur son ordinateur.

Le **diagramme de Gantt** est un outil en ordonnancement et en gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet. Il s'agit d'une représentation d'un graphe connexe, valué et orienté, qui permet de représenter graphiquement l'avancement du projet.

Sothink Logo Maker Professional peut vous aider à créer vos propres logos. Avec cette application, vous pouvez créer visuellement attrayant logos, même si vous ne l'avez pas reçu tout type de formation formelle sur le design graphique. Le programme dispose d'une interface attrayante, similaire à celle de Microsoft Office 97. De plus, vous pouvez le personnaliser avec des styles différents.

**Inkscape** est un logiciel libre de dessin vectoriel sous licence GNU GPL. Il gère des fichiers conformes avec les standards XML, SVG et CSS du W3C.

**GIMP** (***GNU Image Manipulation Program***) est un outil d'édition et de retouche d'image. GIMP a des outils utilisés pour la retouche et l'édition d'image, le dessin à main levée, réajuster, photomontages, convertir entre différents formats d'image, et plus de tâches spécialisées.

- Le **Planificateur de tâches** (en anglais, ***Task Scheduler*** ou ***System Agent***) est un composant du système d'exploitation Windows de Microsoft. Il s'agit d'un programme informatique permettant de planifier l'exécution automatique et périodique de tâches (commandes, scripts, programmes). Par exemple, vider la corbeille tous les dimanches à midi, vérifier la disponibilité de mises à jour toutes les semaines, etc.

## 3. Plateforme de développement

J'ai décidé de travailler avec Java Entreprise Edition pour les raisons précises suivantes :

- ✓ L'utilisateur n'a pas besoin d'installer quoique ce soit pour visualiser notre topologie
- ✓ C'est le navigateur Internet qui se charge de contacter le serveur via une URL pour visualiser notre topologie

- La plateforme Java entreprise (Java EE) est un ensemble de spécifications coordonnées et pratiques qui permettent ensemble des solutions pour le développement, le déploiement, et de la gestion des applications multi tiers centralisées sur un serveur. Construit sur la plateforme de Java 2 édition standard (Java SE), la plateforme Java EE ajoute les possibilités nécessaires pour fournir une plateforme complète, stable, sécurisée, et rapide de Java au niveau entreprise. La plateforme entreprise fournit un ensemble de services permettant aux composants de dialoguer entre eux :

  - HTTP et HTTPS
  - Servlet
  - Java Server Pages (JSP)
  - Le JavaBeans
  - Java DataBase Connectivity (JDBC)
  - Java EE Connector Architecture
  - Gestionnaires de ressources

**a) JSP**

Le Java Server Pages ou JSP est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique. Depuis la version 2.0 des spécifications, la syntaxe JSP est complètement conforme au standard XML. 51 La syntaxe du JSP ajoute des balises XML, appelées actions JSP, qui peuvent être utilisées pour appeler des fonctions. De plus, cette technique permet la création de bibliothèques de balises JSP (taglib) qui agit comme des extensions au HTML ou au XML. Les bibliothèques de balises offrent une méthode indépendante de la forme pour étendre les fonctionnalités d'un serveur HTTP. Il existe aussi un langage de script particulier, appelé Expression Language (EL) destiné à réduire l'injection de code java au sein des pages JSP ainsi qu'à étendre les possibilités des taglibs, tel que la JSTL.

**b) JavaBeans**

**JavaBeans** est une technologie de composants logiciels écrits en langage Java. La spécification JavaBeans de Oracle définit les composants de type JavaBeans comme « des composants logiciels réutilisables manipulables visuellement dans un outil de conception ». Ils sont utilisés pour encapsuler plusieurs objets dans un seul objet : le « bean » (ou haricot en français). Le « bean » regroupe alors tous les attributs des objets encapsulés, et peut définir d'autres attributs si besoin.

Ainsi, il représente une entité plus globale que les objets encapsulés de manière à répondre à un besoin métier.

## Librairies utilisées :

### a) JDBC

JDBC (Java DataBase Connectivity) est une interface de programmation pour les programmes utilisant la plateforme Java. Elle permet aux applications Java d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC. Normalement, il s'agit d'une base de données relationnelle, et des pilotes JDBC sont disponibles pour tous les systèmes connus de bases de données relationnelles.

### b) JSTL

La JSTL en tant que bibliothèque de balises JSP, propose de développer une page JSP sans y utiliser de code Java directement. Elle utilise ainsi une syntaxe proche des langages utilisés par les web designers (HTML ou XHTML) ce qui leur permet de concevoir des pages dynamiques complexes sans connaissances préalable du langage Java.

### c) JDOM

JDOM (acronyme de l'anglais Java Document Object Model), est une bibliothèque open source pour manipulation des fichiers XML en Java. Elle intègre DOM et SAX, et supporte XPath et XSLT. Elle utilise des analyses syntaxiques externes pour construire les documents.

- La plateforme JSE

**Java SE** destinée typiquement aux applications pour poste de travail. Cette plateforme contient outre toutes les API de base :

- Toutes les API spécialisées dans le poste client (JFC et donc Swing, AWT et Java2D)
- Ainsi que des API d'usage général comme JAXP (pour le *parsing* XML)
- Et JDBC (pour la gestion des bases de données).

## Le package : java.net

Le package java.net des API JSE contient une collection de classes et interfaces qui fournissent les détails de communication de bas niveau, ce qui permet d'écrire des programmes qui mettent l'accent sur la résolution du problème à la main. Ces classes fournissent la communication réseau indépendant du système.

Le package java.net fournit un support pour les deux protocoles réseau communs : TCP & UDP.

Méthodes de la classe InetAddress :

| | Méthodes avec Description |
|---|---|
| 1 | **static InetAddress getByAddress(byte[] addr)**<br><br>Retourne un objet InetAddressn, étant donné l'adresse IP. |
| 2 | **static InetAddress getByAddress(String host, byte[] addr)**<br><br>Crée un InetAddress basée sur le nom d'hôte fourni et l'adresse IP. |
| 3 | **static InetAddress getByName(String host)**<br><br>Détermine l'adresse IP d'un hôte, étant donné le nom de l'hôte |
| 4 | **String getHostAddress()**<br><br>Renvoie la chaîne d'adresse IP dans la présentation textuelle. |
| 5 | **String getHostName()**<br><br>Obtient le nom d'hôte pour cette adresse IP. |
| 6 | **static InetAddress InetAddress getLocalHost()**<br><br>Renvoie le local host. |
| 7 | **String toString()**<br><br>Convertit cette adresse IP en une chaîne. |

***Table 5 :*** **Tableau des méthodes de la classe InetAddress.**

## 4. SGBD

***MySQL Workbench*** (anciennement *MySQL administrator*) est un logiciel de gestion et d'administration de bases de données. L'une des spécificités de ce client de gestion de base de données réside dans sa possibilité de visualiser en temps réel, via des indicateurs graphiques, la charge du serveur MySQL, le pourcentage d'occupation en mémoire, les connexions courantes, le nombre de requêtes continues sur une base de données. Il permet aussi de voir l'occupation disque, ainsi que l'espace alloué aux tables et aux fichiers de logs. Enfin, les tables sont disponibles sous forme de diagramme, permettant ainsi la modélisation des données.

### 5. Serveur d'application

**Apache Tomcat** est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'*Apache Software Foundation*. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

# II. Configuration

## 1. Configuration matérielle

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

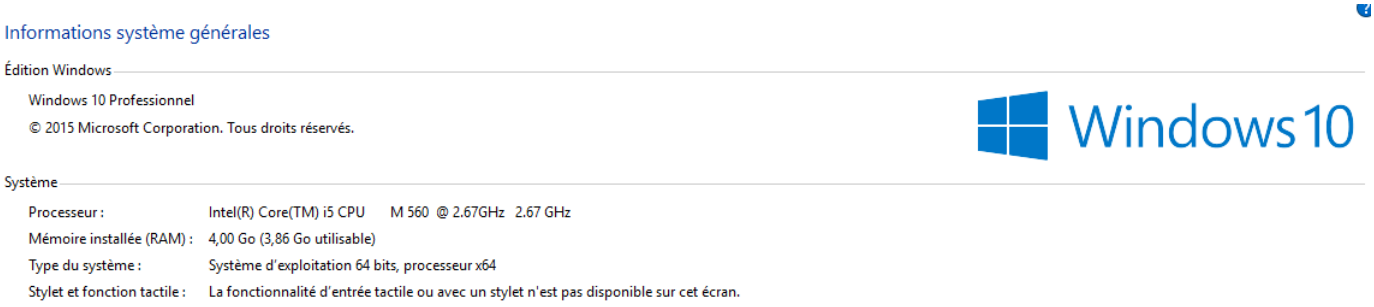


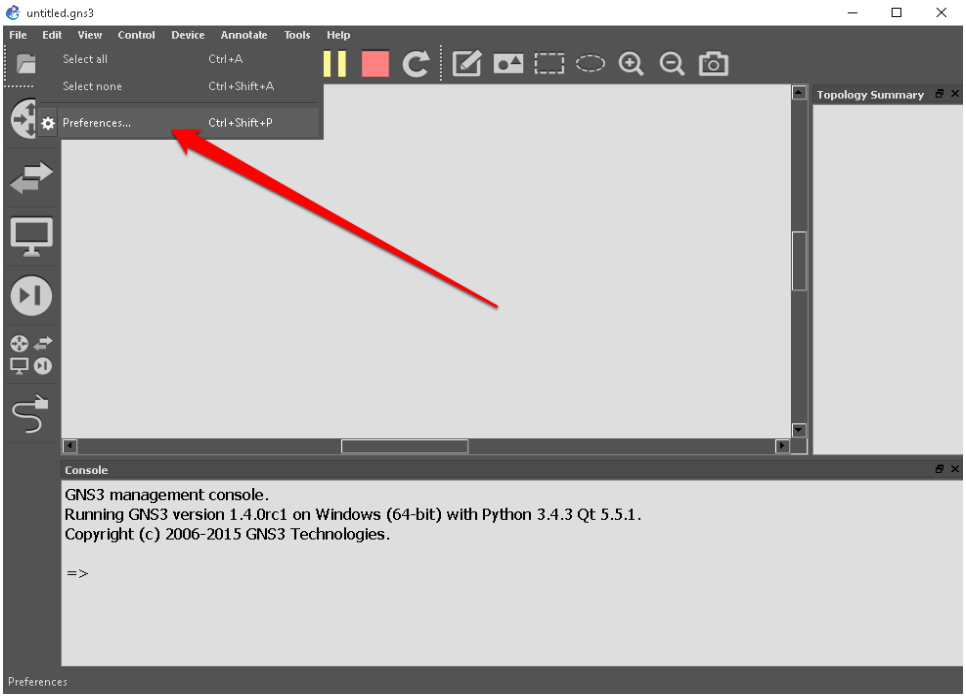

***Figure 16 :*** **caractéristiques du PC**

## 2. Installation et configuration de GNS3

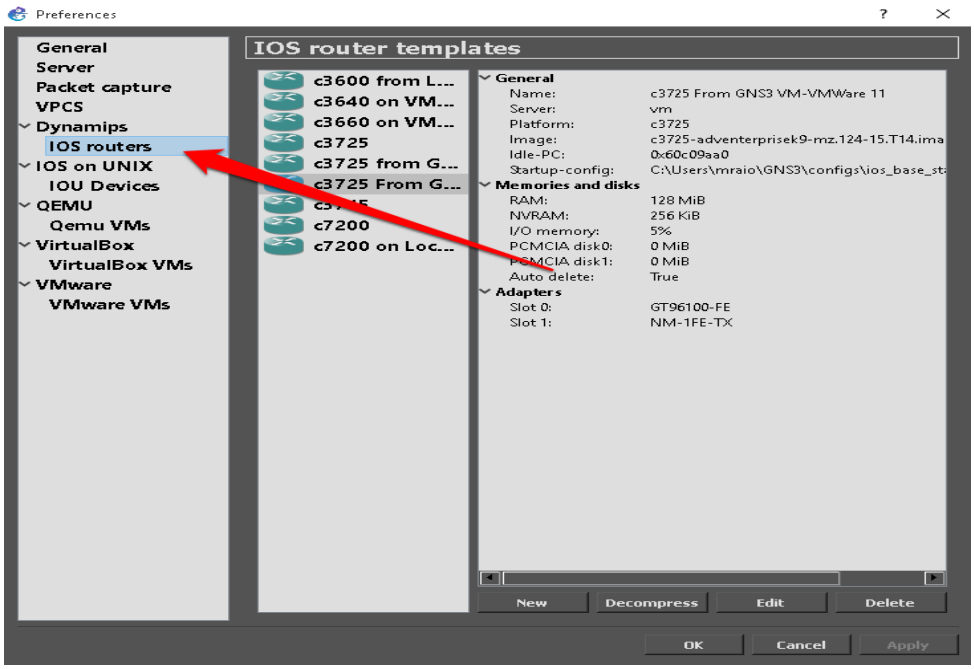
GNS3 (Graphical Network Simulator) est un simulateur de réseau graphique qui permet l'émulation des réseaux complexes. GNS3 permet l'émulation à l'aide de Cisco Internetwork Operating Systems. Il permet d'exécuter un IOS Cisco dans un environnement virtuel sur un ordinateur. GNS3 est une interface graphique pour un produit appelé Dynagen.

### *Addition des images IOS*

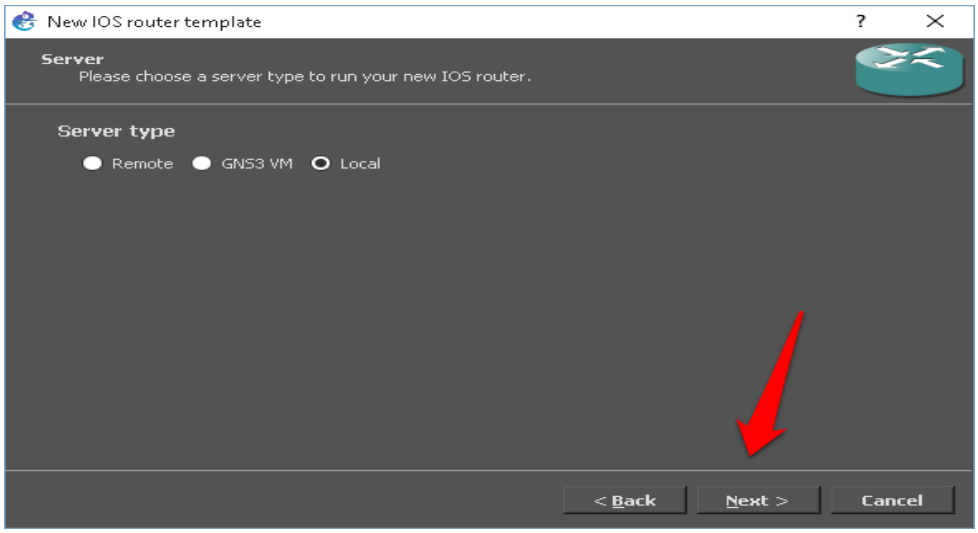
Etape 1 : aller vers Edit -> Préférences



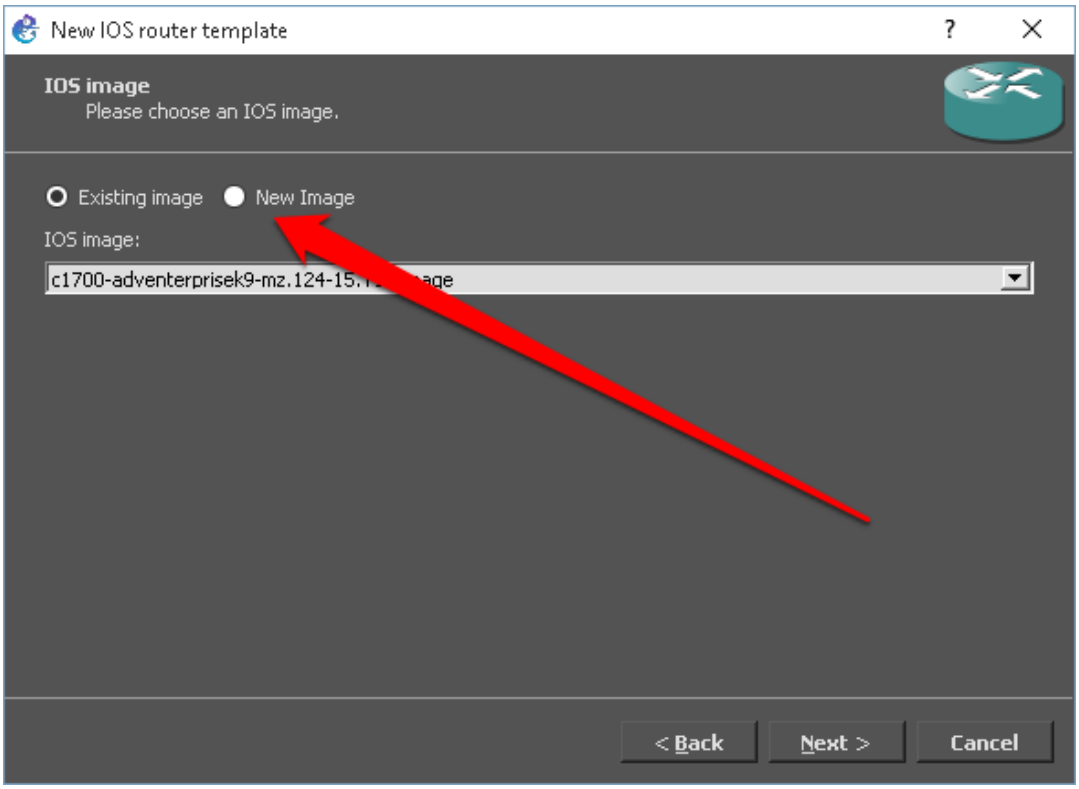

Etape 2 : Cela va vous montrer la fenêtre des préférences, développez Dynamips en cliquant sur la flèche à côté. Vous verrez "routeurs IOS" affichés.




Etape 3 : Sélectionnez les routeurs IOS, cela va afficher la liste des routeurs IOS actuellement configurés. Cliquez sur Nouveau pour ajouter un routeur.

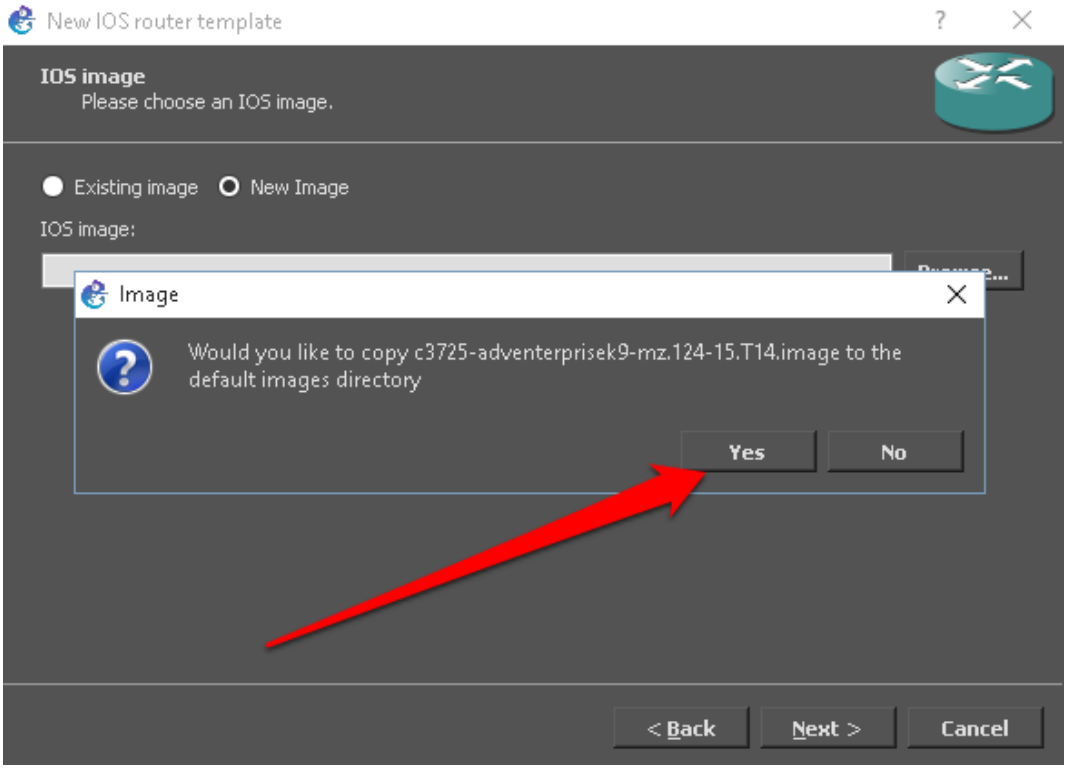



Cliquez sur Suivant. Noter que : Cette capture d'écran montre des options pour télécharger l'image IOS à différents endroits GNS3 Server.



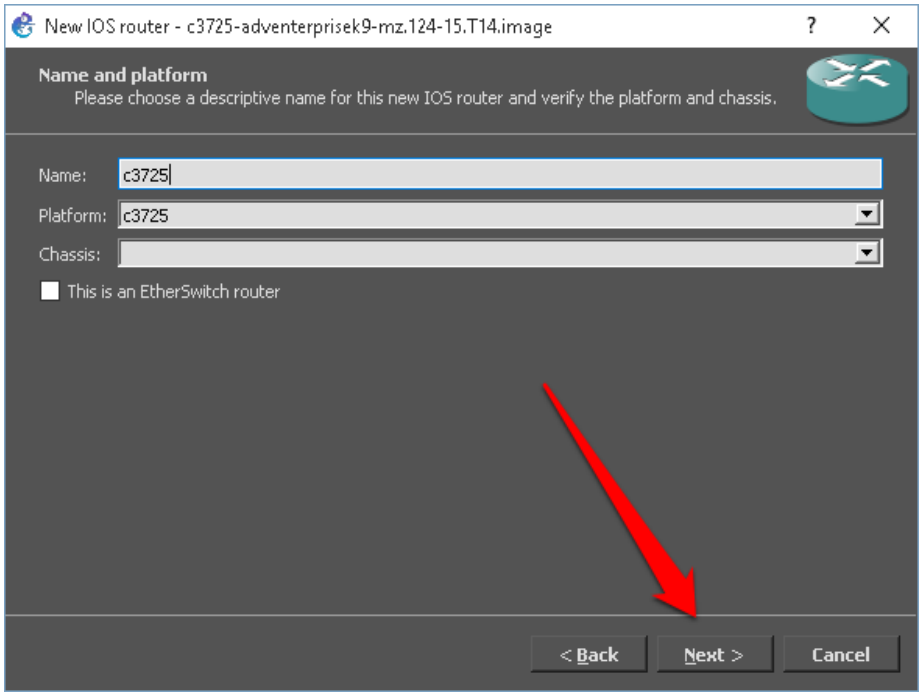

Étape 4 : Vous êtes maintenant dans l'assistant routeur IOS, Tout d'abord, sélectionnez une image IOS à ajouter. Si vous ne l'avez pas fait avant, vous devez sélectionner "New Image".

Ensuite, sélectionnez "Parcourir". Sélectionnez l'image que vous souhaitez télécharger.

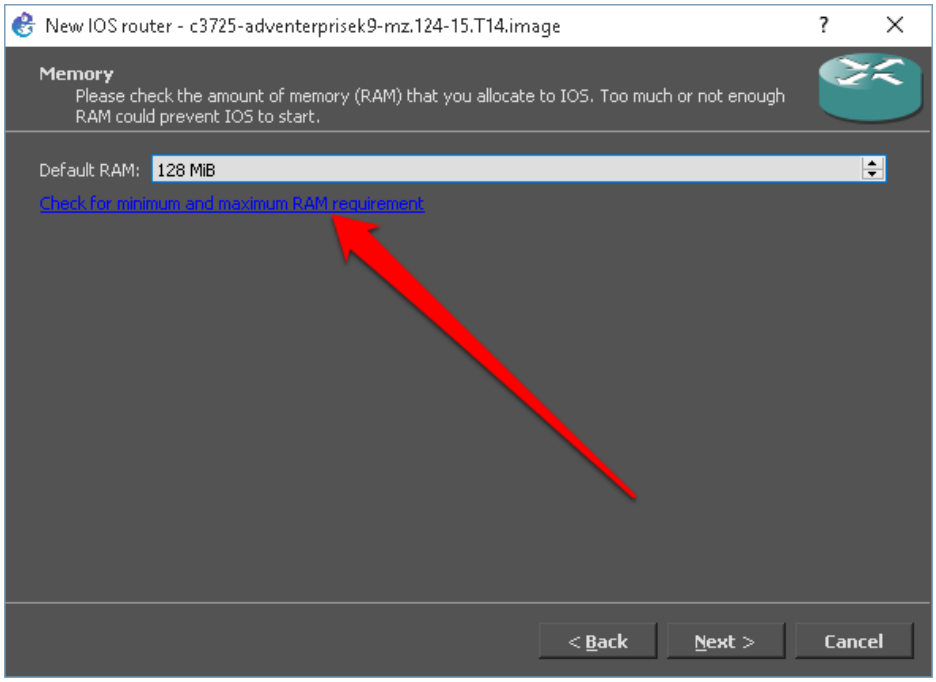



Cliquez sur "Oui." Qu'est-ce que GNS3 va faire est de copier votre fichier image dans le dossier GNS3 par défaut pour la facilité de recréer plus d'images dans l'avenir.

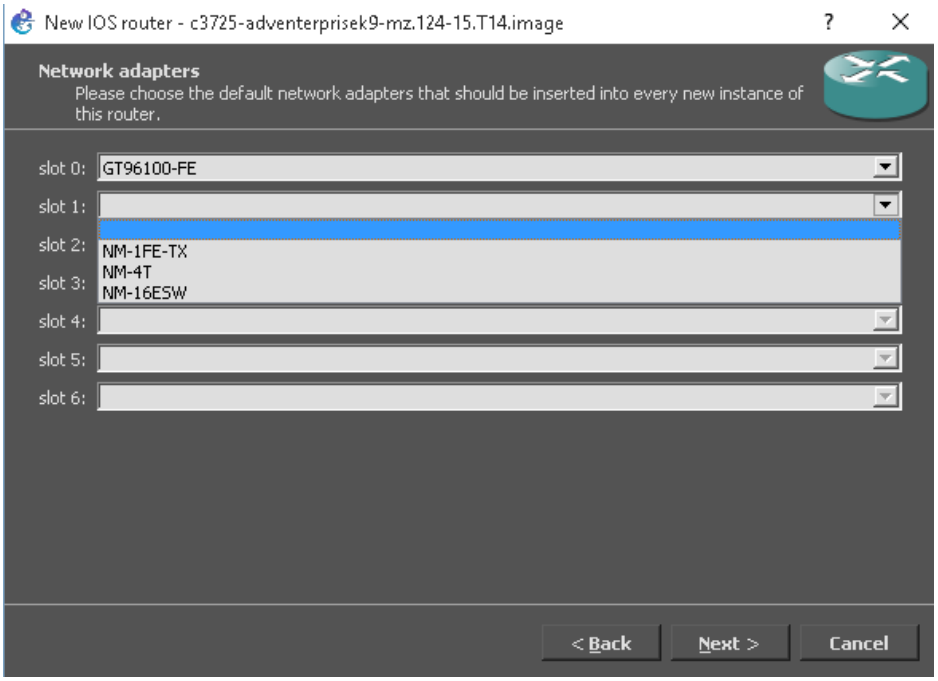
Étape 5 : Vous pouvez maintenant choisir un nom descriptif pour votre routeur. Vérifiez également la plate-forme et le châssis sont corrects pour votre image.



Étape 6 : Vous pouvez changer le nom de cette image et spécifier la plate-forme et châssis. Une caractéristique supplémentaire est de cliquer sur le "Ceci est un routeur EtherSwitch." Utilisez cette option si vous prévoyez d'ajouter le module NM-16ESW Ethernet pour transformer ce routeur dans un commutateur. Sinon, cliquez sur Suivant.



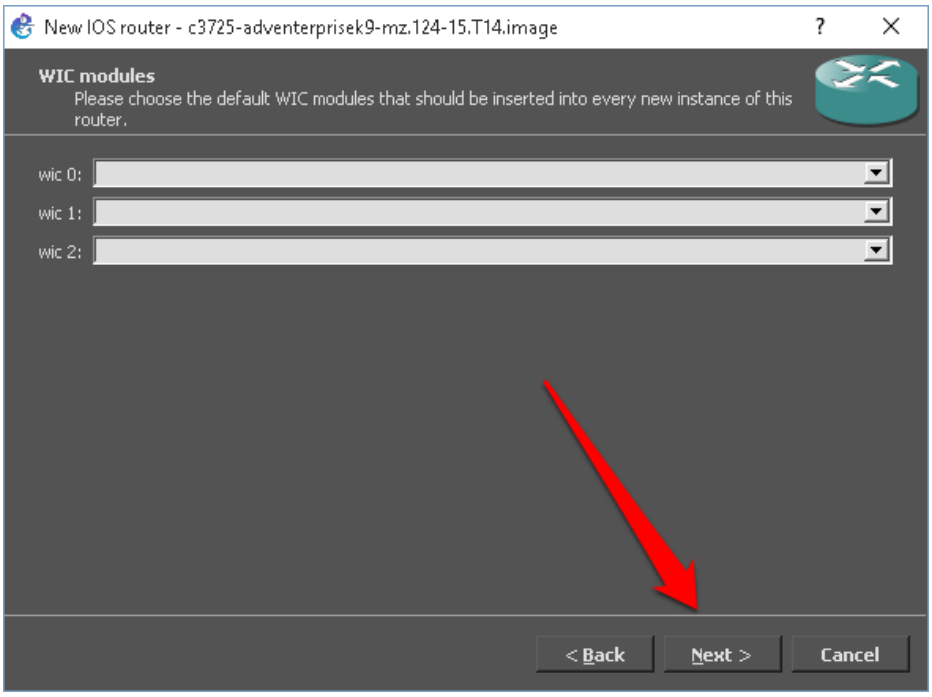

Étape 7 : Choisissez la quantité de RAM à allouer au routeur. Cliquez sur Suivant.



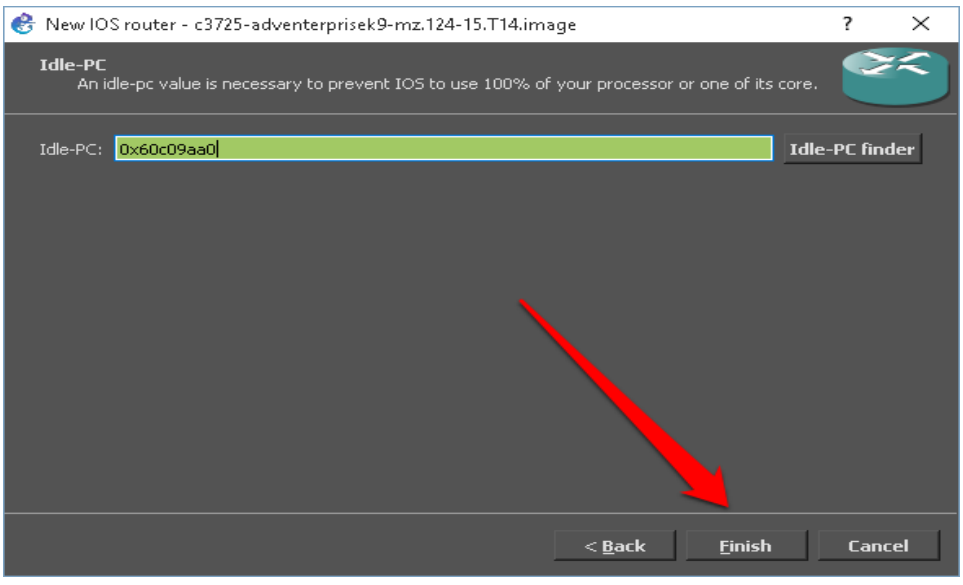
Sélectionnez le type de module que vous souhaitez utiliser.

Etape 8 : Comme mentionné précédemment, ajouter toutes les cartes WIC au routeur que vous souhaitez ajouter.

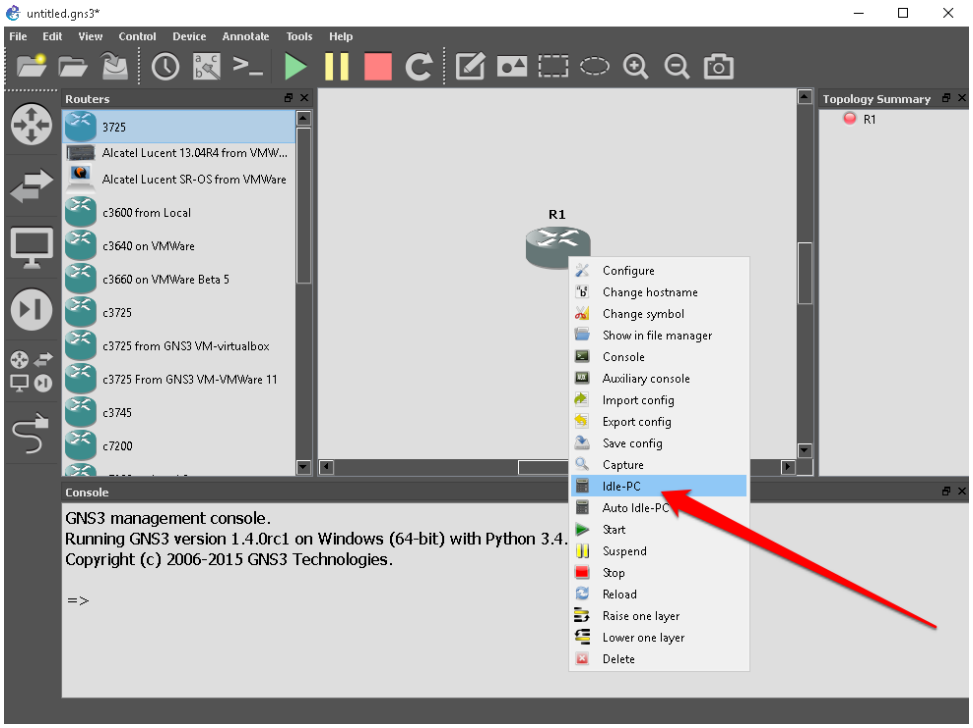
Vous pouvez laisser ces vides et les ajouter plus tard lors de la création de votre topologie si vous le souhaitez. Les avantages pour les modules ajoutant désormais les enregistrer comme préférences pour cette image. Cela réduit le temps de configurer chaque routeur unique vous faites glisser sur le panneau de la topologie.




Étape 9 : Vous pouvez ajouter des cartes WIC, mais si vous ne les avez pas un besoin pour eux, vous pouvez simplement utiliser les modules dans les SLOTS pour la connectivité. De toute façon, cliquez sur "Suivant".



Étape 10 : Idle-PC finder varie entre les systèmes d'exploitation. Le New GNS3 va trouver une valeur pour vous. Si cette valeur ne vient pas tout de suite, puis cliquez sur "Terminer", puis démarrez le routeur dans le volet de la topologie et clic droit. L'une des options est "Idle-PC."




# III. Interface-Homme-Machine

Ici, je vais présenter quelques interfaces de l'application

- Authentification :

  L'accès aux services de l'application nécessite une authentification.

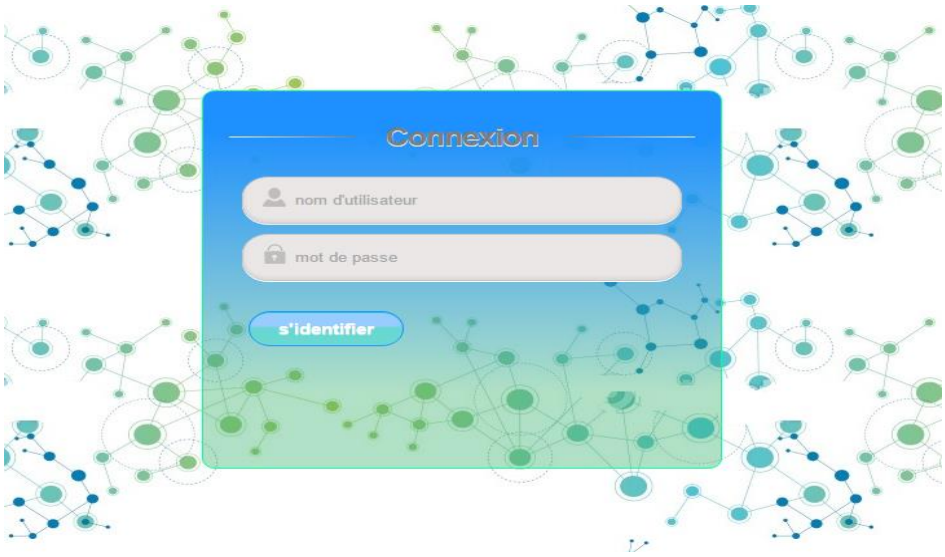



*Figure 17 :* **La page d'authentification**

Si le login ou le mot de passe est incorrect l'application va demander à l'utilisateur de s'authentifier à nouveau en affichant la page d'authentification.

- Accueil de l'administrateur :

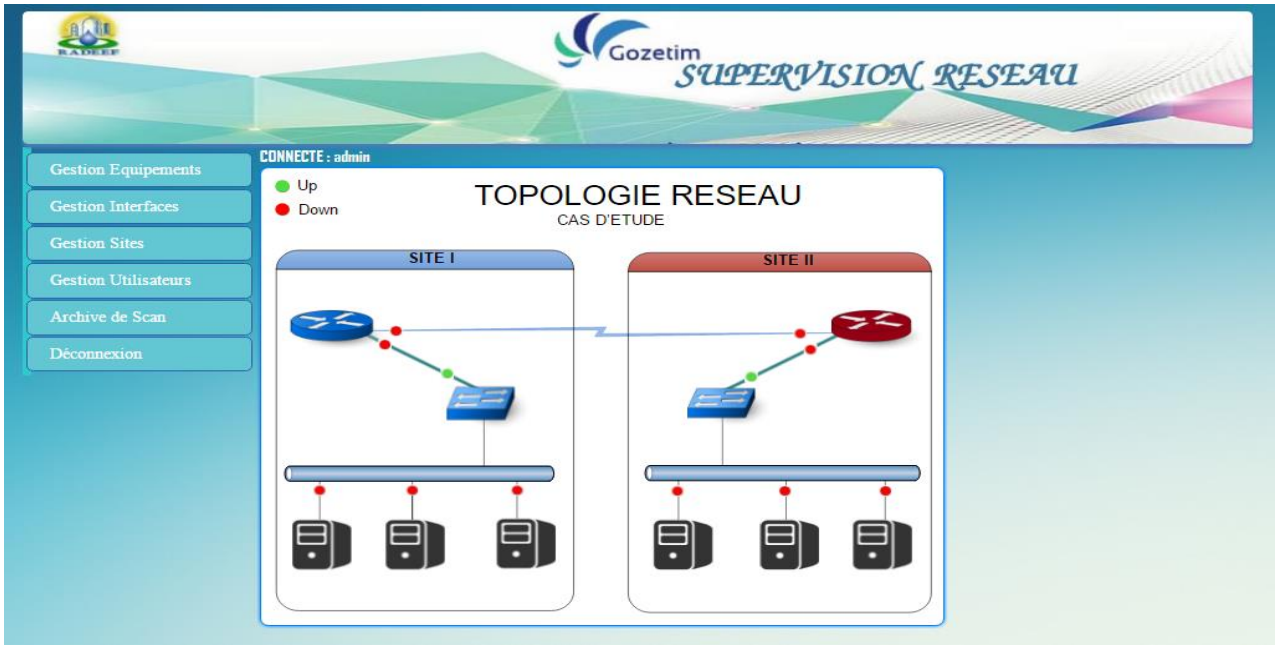



*Figure 18 :* **L'accueil de l'administrateur**

- Création d'un nouvel équipement :

***Figure 19 :*** **La page de création d'un nouvel équipement**

- Liste des utilisateurs :

***Figure 20 :*** **La page de la liste des utilisateurs**

- Supprimer Utilisateur :

***Figure 21 :*** **la page « Supprimer un Utilisateur »**

- Consulter l'archive scan :

*Figure 22 :* **La page de l'archive de scan**

- Accueil de l'observateur :

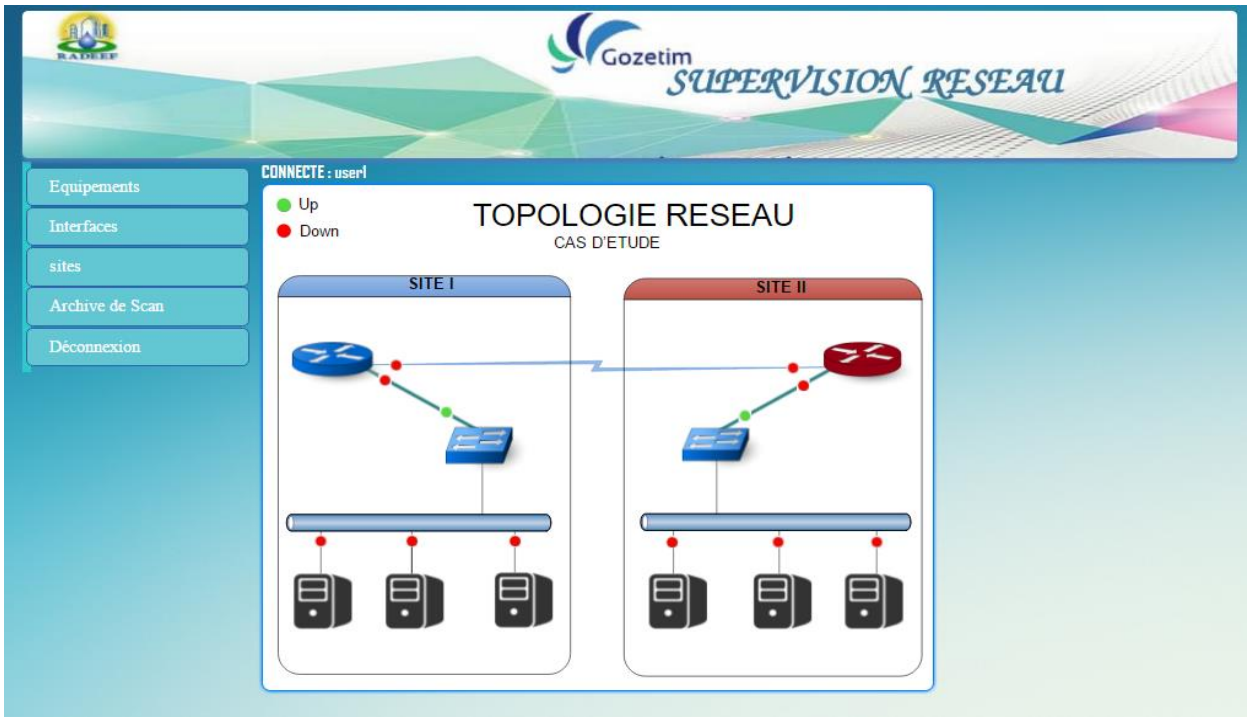



*Figure23 :* **L'accueil de l'observateur**

- Consulter Interfaces :

| Libelle Equipement | Ordre_interface | L'adresse IP | Nom_interface |
|---|---|---|---|
| Cisco RV016 16-port 10/100 VPN Router - Multi WAN | 1 | 127.0.0.1 | Se1/0 |
| | 2 | 173.7.8.4 | Fa0/0 |
| TP-LINK 24-Port Gigabit Ethernet Rackmount Switch (TL-SG1024) | 1 | 127.0.0.2 | Fa0/0 |
| | 2 | 45.7.8.3 | Se1/0 |
| | 3 | 23.76.78.45 | F0/2 |
| NETGEAR 16-Port Gigabit Ethernet Desktop Switch (GS316) | 1 | 127.0.0.3 | Fa0/0 |
| Linksys 24-Port Business Gigabit | 1 | 127.0.0.4 | Fa0/0 |
| Dell Server Evolutivity MR-S | 1 | 192.168.1.89 | Fa0/0 |
| LDLC Server Evolutivity MP-S | 1 | 192.168.5.45 | Fa0/0 |
| PowerEdge C Server Evolutivity MP-S | 1 | 145.8.5.4 | Fa0/0 |
| HDD 400Go | 1 | 10.0.4.0 | Fa0/0 |
| HP ProLiant ML110 Gen9 (794994-425) | 1 | 10.6.9.34 | Fa0/0 |
| Lenovo ThinkServer TS140 - 70A5001YFR - Xeon E3 1226 | 1 | 127.0.0.5 | Fa0/0 |
| | 2 | 23.56.104.1 | F0/2 |
| | 3 | ll | mp |
| dell-366fgjj | 1 | 23.67.104.1 | Se0/1 |

*Figure24 :* **la page « Consulter Interfaces »**

**Conclusion**

Au cours de ce chapitre, j'ai présenté en détails la projection de la conception du plan théorique sur le plan pratique. J'ai décrit la plate-forme logicielle (GNS3). Ensuite, j'ai montré les interfaces les plus significatives de mon application.

# CONCLUSION GENERALE

Mon stage a été une expérience enrichissante qui m'a permis d'une part d'appliquer et d'approfondir mes connaissances acquises tout au long de ma formation à la faculté des sciences et techniques et d'autre part, de me familiariser avec la conduite des projets informatiques et d'explorer le milieu professionnel.

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

Le développement de mon application, m'a également permis de m'améliorer en matière de conduite de projet et de les mettre en pratique.

En fait, la réalisation de ce projet de fin d'études, m'a donnée une véritable occasion pour rencontrer les difficultés et l'environnement du travail qu'un informaticien devra affronter lors de la conception et de la réalisation d'une application. Ce projet m'a demandée beaucoup d'efforts afin de respecter les attentes.

Enfin, l'application que j'ai réalisé présente un aspect modulaire qui facilite la tâche pour les modifications futures. Je souhaite, ainsi, qu'elle apporte l'aide nécessaire aux différents agents de la RADEEF et restera un noyau riche pour préparer un avenir prospère. Mon 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

## Bibliographie

- Modélisation UML -- A. Benabbou.
- IHM en JAVA -- A. Majda
- TCP/IP – K. Zenkouar
- Cours JEE – S.Najah

## Webographie

- https://openclassrooms.com/
- http://java.developpez.com/
- http://stackoverflow.com/
- www.memoireonline.com
- www.commentçamarche.net
- www.oracle.com
- fr.wikipedia.org
- http://www.jmdoudoux.fr/

# ANNEXE

## 1. Client-serveur

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

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

✓ Avantages de l'architecture client/serveur

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

- **Des ressources centralisées** : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction
- **Une meilleure sécurité** : car le nombre de points d'entrée permettant l'accès aux données est moins important
- **Une administration au niveau serveur** : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés
- **Un réseau évolutif** : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.

## 2. Client léger

Le terme « client léger » par opposition au client lourd, désigne une application accessible via une interface web (en HTML) consultable à l'aide d'un navigateur web, où la totalité de la logique métier est traitée du côté du serveur. Pour ces raisons, le navigateur est parfois appelé client universel.

L'origine du terme lui-même provient de la pauvreté du langage HTML, qui ne permet de faire des interfaces relativement pauvres en interactivité, si ce n'est pas le biais du langage JavaScript.

Le fait que l'essentiel des traitements soit réalisé du côté du serveur et que l'interface graphique est envoyée au navigateur à chaque requête permet une grande souplesse de mise à jour. En contrepartie, l'application doit s'affranchir des différences d'interprétation du code HTML par les différents navigateurs et l'ergonomie de l'application possède un champ réduit.