

REPUBLIQUE TUNISIENNE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE VIRTUELLE DE TUNIS



RAPPORT DE PROJET DU FIN D'ETUDES

Sujet :

Développement d'un outil de
supervision d'un système
d'exploitation (Tunisie Télécom)

Elaboré par

Makrem Mannai

Boutheina Khabouchi

Youssef Hammami

Encadré par :

Mourad Zouari & Kamel Khediri

Société d'accueil : *Tunisie Telecom*

Année Universitaire 2010/2011

Remerciements

Nous voudrions remercier nos encadreurs, Mr Mourad Zouari et Mr Kamel Khdhiri pour leur soutien et ses recommandations judicieuses. Aussi, remercions-nous tout personnel de Tunisie Telecom ayant contribué de près ou de loin dans le bon déroulement du projet.

Nous devons chaque bricbe de notre connaissance à nos enseignants à l'Université Virtuelle du Tunis qui ont si bien mené leur noble quête d'enseigner tous ce qu'il faut. Nous les remercions non seulement pour le savoir qu'ils nous ont transmis, mais aussi pour la fierté et l'ambition que leurs personnes nous aspirent.

Non loin de tout projet ambitieux, il existe des gens qui partagent sans jamais se lasser ses meilleurs et ses pires moments. Ce sont des personnes qui nous ont soutenus dans chacun de nos instants de faiblesse, et qui, sans leurs extrême attention et gentillesse, nous ne serions pas ce que nous sommes devenu aujourd'hui. En tête de liste de ces gens nous placerons nos familles bien aimées qui n'ont jamais hésité à nous offrir le meilleur qu'elles pouvaient. Viennent alors tous nos camarades de promotion ainsi que tous nos amis qui nous ont aidés de façon directe ou tout simplement par leur présence et leur soutien moral.

Sommaire

INTRODUCTION GÉNÉRALE.....	1
CHAPITRE I : CADRE GÉNÉRAL DU PROJET.....	2
I. Introduction.....	3
II. Présentation de l'organisme d'accueil.....	3
A. Historique de Tunisie Telecom.....	3
B. Organigramme de l'organisme.....	4
III. Présentation du sujet.....	6
A. Intitulé.....	6
Définition du projet :.....	6
B. Intérêts du projet.....	6
C. Conclusion.....	6
CHAPITRE II : ÉTUDE DE L'EXCISANT.....	7
I. Introduction :.....	8
II. Etude de l'existant :.....	8
A. Critique de l'existant.....	8
1. Critique.....	8
2. Solution proposée :.....	9
III. Conclusion.....	9
CHAPITRE III : ANALYSE ET SPÉCIFICATION DES BESOINS.....	10
I. Introduction.....	11
II. Spécification des besoins fonctionnels et non fonctionnels.....	11
A. Spécification des besoins fonctionnels.....	11
B. Spécification des besoins non fonctionnels.....	12
III. Spécification semi-formelle des besoins.....	13
A. . Méthode de conception.....	13
1. UML.....	13

IV.	Diagrammes de cas d'utilisation.....	14
A.	Diagramme des cas d'utilisation de l'administrateur	14
B.	Diagramme des cas d'utilisation du superviseur	14
C.	Diagramme des cas d'utilisation global de système	16
V.	Conclusion	16
 CHAPITRE IV : CONCEPTION.....		17
I.	Introduction.....	18
II.	Conception de l'application	18
III.	Modélisation conceptuelle.....	18
A.	Conception de la base de données en se basant sur le langage UML.....	18
1.	Le diagramme de classe global de système	19
2.	Le diagramme de séquence.....	20
IV.	Conclusion.....	24
 CHAPITRE V : APPLICATION GRAPHIQUE DE LA SUPERVISION.....		25
I.	Présentation générale du l'application :	26
II.	Principe de Fonctionnement de l'application de la supervision :	26
A.	Description de la partie A :	26
B.	Description de la partie B :.....	27
III.	Description de l'application de la supervision :	27
IV.	description de la première partie	28
A.	notion sur java :	28
1.	Qu'est-ce que la technologie java	28
2.	Pourquoi ai-je besoin de Java ?.....	28
3.	Java est-il disponible gratuitement au téléchargement ?	28
4.	Pourquoi devrais-je passer à la dernière version de Java ?.....	28
B.	Concept de socket	28
C.	Adresses IP.....	29
1.	La classe InetAddress.....	30
D.	Sockets TCP	30
1.	Le modèle client/serveur	30
2.	La classe Socket.....	32
3.	La classe ServerSocket.....	33

4. Options des sockets serveurs	34
V. Description de la deuxième partie.....	36
A. Présentation :	36
B. Caractéristiques	36
C. Systèmes d'exploitation supportés	36
D. Utilisation	36
E. Dénomination	37
F. Historique	37
G. Quelque nation sur MySQL	38
VI. Description de la troisième partie.....	39
A. notion sur PHP :.....	39
1. Qu'est-ce que PHP?	39
2. Origines de PHP.....	39
3. SGBD supportés par PHP	40
4. Implantation du code Php	40
VII. Réalisation de Projet :	43
 CONCLUSION GÉNÉRALE.....	 48

Introduction générale

La supervision est la "surveillance du bon fonctionnement d'un système ou d'une activité". Elle permet de surveiller, rapporter et alerter les fonctionnements normaux et anormaux des systèmes informatiques.

Elle répond aux préoccupations suivantes :

- Technique : surveillance du réseau, de l'infrastructure et des machines,
- Applicative : surveillance des applications et des processus métiers,
- Contrat de service : surveillance respect des indicateurs,
- Métier : surveillance des processus métiers de l'entreprise.

En cas de dysfonctionnement, le système de supervision permet d'envoyer des messages sur la console de supervision, ou bien d'envoyer un courriel à l'opérateur. Mais si le dysfonctionnement se produit en dehors des heures de bureau, et en l'absence de système approprié, l'alerte n'est pas reçue par l'opérateur, et les utilisateurs des applications ne sont pas prévenus du dysfonctionnement. C'est pourquoi il peut être utile de compléter le superviseur par un logiciel de gestion des alertes, qui envoie automatiquement un courriel, un SMS, ou un appel téléphonique à un opérateur sous astreinte.

D'où l'intérêt de notre projet de fin d'études qui consiste à développer une application de la supervision performante et facile à utiliser.

Notre rapport est composé en 5 chapitres :

- Le premier Chapitre présente le cadre générale du notre projet.
- Le deuxième Chapitre c'est une étude de l'existant qui présente les différentes problématiques.
- Le troisième chapitre présente une analyse et spécification des besoins
- Le quatrième chapitre présente une conception du notre application
- Le cinquième chapitre présente la partie réalisation de l'application de la supervision

Chapitre I : Cadre général du projet

I. Introduction

Dans ce chapitre, nous allons donner une brève description de la société d'accueil et présenter son organigramme. Ensuite, nous allons expliquer le sujet de notre projet et ses objectifs. Enfin, nous allons poser la problématique et les solutions envisagées.

II. Présentation de l'organisme d'accueil

A. Historique de Tunisie Telecom



logo de Tunisie Telecom

Tunisie Télécom (اتصالات تونس) est le nom commercial de l'opérateur historique de télécommunications en Tunisie.

La loi portant création de l'Office national des télécommunications, dont le nom commercial est *Tunisie Télécom*, est promulguée le 17 avril 1995 et entre en vigueur le 1^{er} janvier 1996. Devenu société anonyme de droit public fin 2002, il change de statut juridique, par un décret du 5 avril 2004, pour devenir une société anonyme dénommée

« Tunisie Télécom ». Elle connaît une privatisation partielle en juillet 2006 avec l'entrée dans son capital, à hauteur de 35 %, de l'émirati Tec om-DIG.

Tunisie Télécom propose des services dans le domaine des télécommunications fixes et mobiles. En juin 2006, il est fort de 1 259 000 abonnés au réseau fixe (RTCP), dont il détient

le monopole, et de 3 265 000 abonnés au réseau GSM (la première ligne est inaugurée le 20 mars 1998, faisant de lui le leader sur ce marché devant l'opérateur privé Tunisiana.

Depuis 2008, Tunisie Télécom offre la possibilité aux détenteurs de cartes bancaires nationales d'alimenter le solde de leurs lignes prépayées via les distributeurs automatiques de billets de l'Arab **Tunisien** Bank (service Mobilink).

Le 21 mars 2009, Tunisie Télécom lance une nouvelle marque, Elissa, avec des offres spécifiquement conçues pour les jeunes de moins de 25 ans. Il est également un fournisseur d'accès à Internet (Frame Relay, ADSL, X.25, LS, RNIS et WLL pour la téléphonie rurale).

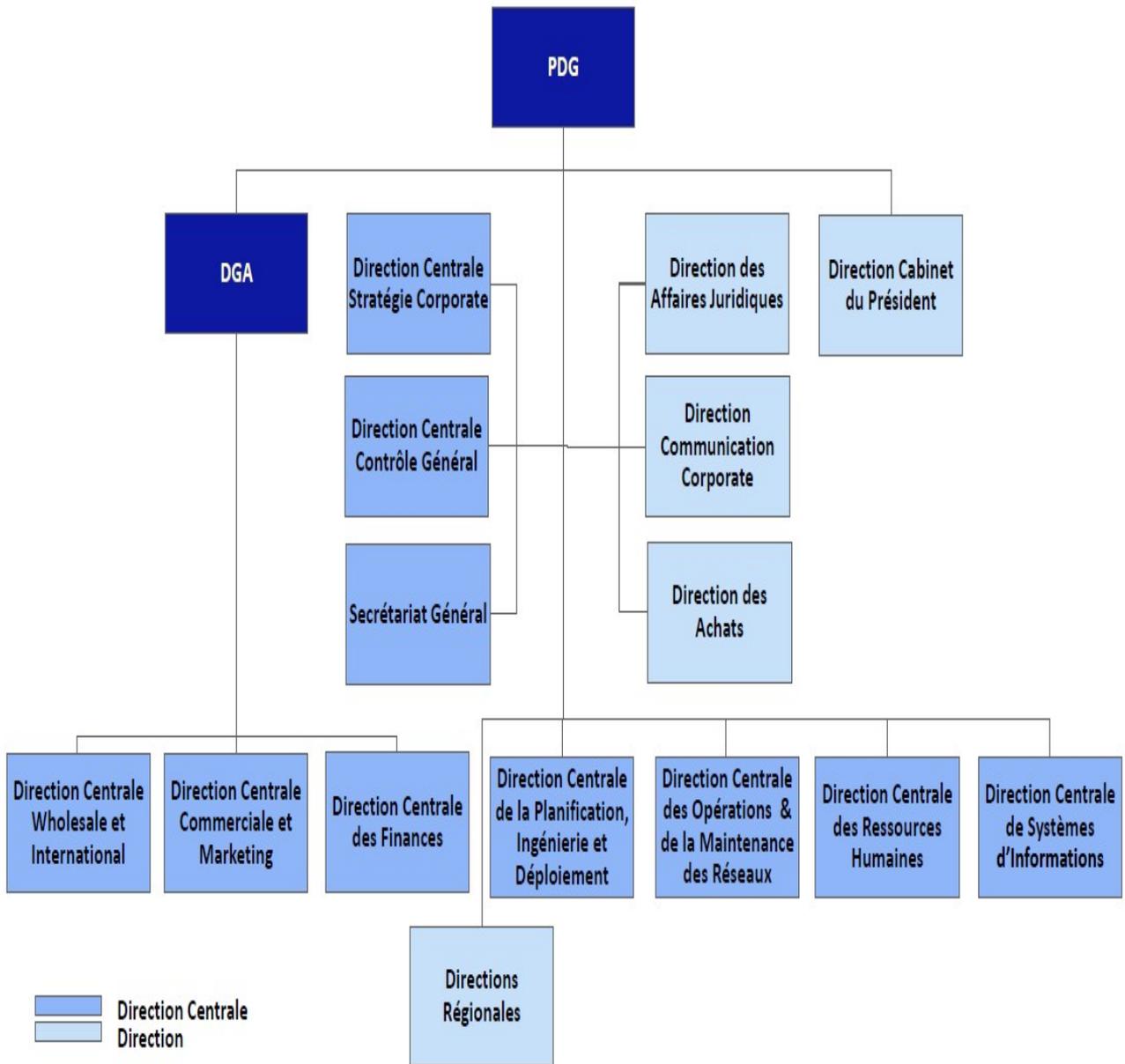
Tunisie Télécom met en place, exploite et commercialise le premier réseau GSM en Mauritanie (Mattel) à partir de mai 2000. Elle conclut également une convention de coopération technique avec Djibouti Télécom pour le développement de ses réseaux de télécommunications.

Tunisie Telecom El Kasbah est organisé en plusieurs directions, comme exemple le centre de service à valeur ajoutée où nous avons réalisé ce projet. Le centre de service à valeur ajoutée présente plusieurs travailleurs occupants les postes des Techniciens et des ingénieurs compétents. Aussi, il comprend plusieurs types de serveurs normés, des serveurs de type SUN (SPARC, NITRA...) installer par des systèmes d'exploitation Solaris(UNIX), Redhat (LUNIX) et Windows Server, et de type IBM installer par Windows Server (2003, 2008), AIX (UNIX) et Ubuntu, en plus le type Siemens.

B. Organigramme de l'organisme

Pour concrétiser la mise en place de l'ONT, le conseil d'administration s'est réuni en vue d'établir un organigramme transitoire modélisant la structure interne de l'office.

Cet organigramme fait l'objet de plusieurs reformes et restructurations afin de s'adapter aux nouvelles exigences d'efficience et d'efficacité. En fait, l'organigramme de Tunisie Telecom se présente comme suit :



L'organisation Fonctionnelle de Tunisie Telecom

III. Présentation du sujet

A. Intitulé

Le sujet consiste à concevoir et développer une application de supervision permettant à aider les administrateurs à intervenir en mode proactive afin de minimiser l'indisponibilité des systèmes. En effet l'enregistrement de l'historique des événements de l'activité sur le système informatique et leur analyse sont des points importants dans l'administration des systèmes d'information. Cela permet la construction d'indicateurs pour avoir une vue d'ensemble sur différentes informations (Occupation CPU, Occupation mémoire, etc.) et la détection d'activités suspectes, d'anomalies ou d'erreurs afin d'agir plus rapidement en cas de besoin.

B. Intérêts du projet

Afin de développer un outil de supervision du système d'information de l'entreprise, il a été demandé de chercher des outils ou de développer des scripts permettant d'analyser les différents comportements des systèmes.

Ce travail nécessite comme pré requis de :

- Prendre connaissance de l'architecture des systèmes existants,
- Etudier les interactions entre les différents composants,
- Etudier aussi les performances des systèmes

A titre d'exemple : dans une même plateforme, on peut trouver trois serveurs (Web, Applicatif et Base de données) donc la panne d'un seul peut affecter le reste.

C. Conclusion

Dans ce chapitre, nous avons étudié le contexte général de notre projet, ainsi que l'ensemble des tâches à réaliser. Pour les accomplir, quelques notions relatives à ce travail doivent être définies. Nous avons consacré le chapitre suivant pour illustrer les notions de base nécessaires à l'élaboration de notre projet.

Chapitre II : Étude de l'existant

I. Introduction :

Dans cette partie, nous allons détailler les limites de la solution actuelle. Ensuite, nous allons présenter les meilleures solutions de travail et de sécurité proposées.

II. Etude de l'existant :

Tunisie Telecom est une société qui comporte un nombre important de directions et de services déployés sur l'ensemble du territoire Tunisien.

Tunisie Telecom a comme but de gérer les réseaux et les services publics de Télécommunications. Le DATA CENTER est un centre d'hébergement des serveurs des applications de l'entreprise (AD, exchange, ERP, BSCS ...), servant à abriter des systèmes informatiques et les composants associés, comme les télécommunications et les systèmes de stockage.

Tous les serveurs sont interconnectés par un réseau local dédié qui leur permet de communiquer de manière rapide et sûre. Dans ce centre, il y a un équipe qui partage l'administration des serveurs, qui contrôle et fait le suivi de l'état des serveurs et de la maintenance préventive. On doit noter que au sein de la direction des opérations IT, il y'a un centre de supervision qui assure le suivi des déférentes couches (system, application métier, réseau, Base de données, ...).

A. Critique de l'existant

1. Critique

Au sein de l'équipe infrastructure, l'administrateur assure une vérification manuelle quotidiennement. Cette vérification est nécessite beaucoup et considérée insuffisante pour des systèmes très critique.

Le but du projet est de développer une application permettant d'assurer la supervision de ces machines, le suivi d'occupation RAM, CPU, SWA et disque en temps réel.

Ce projet est né dans l'objectif de faciliter la tâche des administrateurs système. Avec cet outil on peut assurer ce check par exemple toutes les 5 minutes (paramètre fixé par l'utilisateur) et on peut détecter les problèmes qui se produisent instantanément.

2. Solution proposée :

L'évolution du nombre de serveurs chez la Tunisie Telecom et l'obligation de la sécurité dans nos jours a certainement présenté quelques obligations de réagir pour être dans les normes de cette évolution sociétaire. En effet, malgré tous les points forts des équipements logiciels et matériels chez la Tunisie Telecom l'utilisation de certains serveurs ne satisfait pas la sécurité.

D'où vient la nécessité de la mise en place d'un outil de supervision du système qui permet de suivre l'état des systèmes. Cet outil de supervision s'adapte au domaine d'application, il permet aux administrateurs système de recevoir des mails d'alertes.

III. Conclusion

L'étude de l'existant permet de mieux présenter les différents problèmes au sein de la société Tunisie Telecom ainsi l'étude de la solution proposée.

Chapitre III : Analyse et Spécification des besoins

I. Introduction

Ce chapitre présente l'analyse du système d'information à développer. Nous allons présenter les besoins fonctionnels et non fonctionnels de l'application. Enfin, on trouve la représentation de la spécification formelle des besoins à travers des diagrammes de cas d'utilisation qui seront d'une grande utilité dans le processus de développement de l'application.

II. Spécification des besoins fonctionnels et non fonctionnels

L'analyse du sujet nous a permis de dégager les fonctionnalités qui seront mise à la disposition de l'utilisateur. Dans cette partie, nous allons recenser les fonctionnalités que l'application doit offrir à ses différents utilisateurs.

Notre application doit fournir à la fois des fonctionnalités aussi bien pour l'administrateur que pour le superviseur.

A. Spécification des besoins fonctionnels

➤ Côté administrateur

L'administrateur est un technicien ou ingénieur responsable du bon fonctionnement de l'application.

Le logiciel doit permettre :

- ✓ L'accès et l'identification

L'administrateur aura comme rôle :

- ✓ Gérer les plateformes.
- ✓ Editer les serveurs.
- ✓ Gérer les comptes utilisateurs.
- ✓ Gestion des privilèges :
- ✓ Ajout, suppression et modification des privilèges des groupes enregistrés.
- ✓ Régler les Seuils d'alertes

➤ **Côté superviseur**

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

Le logiciel doit permettre :

- ✓ L'accès et identification

Le superviseur a comme rôle :

- ✓ La vérification de l'état des serveurs.
- ✓ La consultation des historiques.
- ✓ L'élaboration des rapports.
- ✓ La comparaison des alertes existantes et les nouvelles.

B. Spécification des besoins non fonctionnels

Les besoins non fonctionnels présentent les exigences internes pour le système et cachées vis à vis les utilisateurs, notre application doit être facile à utiliser, avec une bonne ergonomie. Il doit garantir un temps de réponse court.

➤ **L'ergonomie**

Le système doit présenter des interfaces graphiques conviviales bien structurées du point de vue contenu informationnel.

➤ **La sécurité des données**

Sécuriser les données revient à appliquer une stratégie d'identification, d'authentification, l'autorisation et contrôler 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.

III. Spécification semi-formelle des besoins

A. . Méthode de conception

1. UML

a) 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.

Pierre-Alain Muller ajoute :

« UML est dans le domaine public, soutenue par le marché : Microsoft, HP, IBM, Oracle... Successeur naturel des méthodes de Booch, OMT et OOSE, UML est le fruit de l'expérience et des besoins de la communauté des utilisateurs. »

b) 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.

Objectif :

- 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) utilisée ce qui rend le passage à la génération de code moins évidente...

IV. Diagrammes de cas d'utilisation

A. Diagramme des cas d'utilisation de l'administrateur

Ce diagramme donne à l'administrateur la possibilité de créer des plateformes et des serveurs. En plus, il gère les comptes utilisateurs ainsi que la gestion des privilèges. Enfin, il gère les seuils d'alertes. Chaque administrateur doit s'authentifier avant d'accéder à l'application.

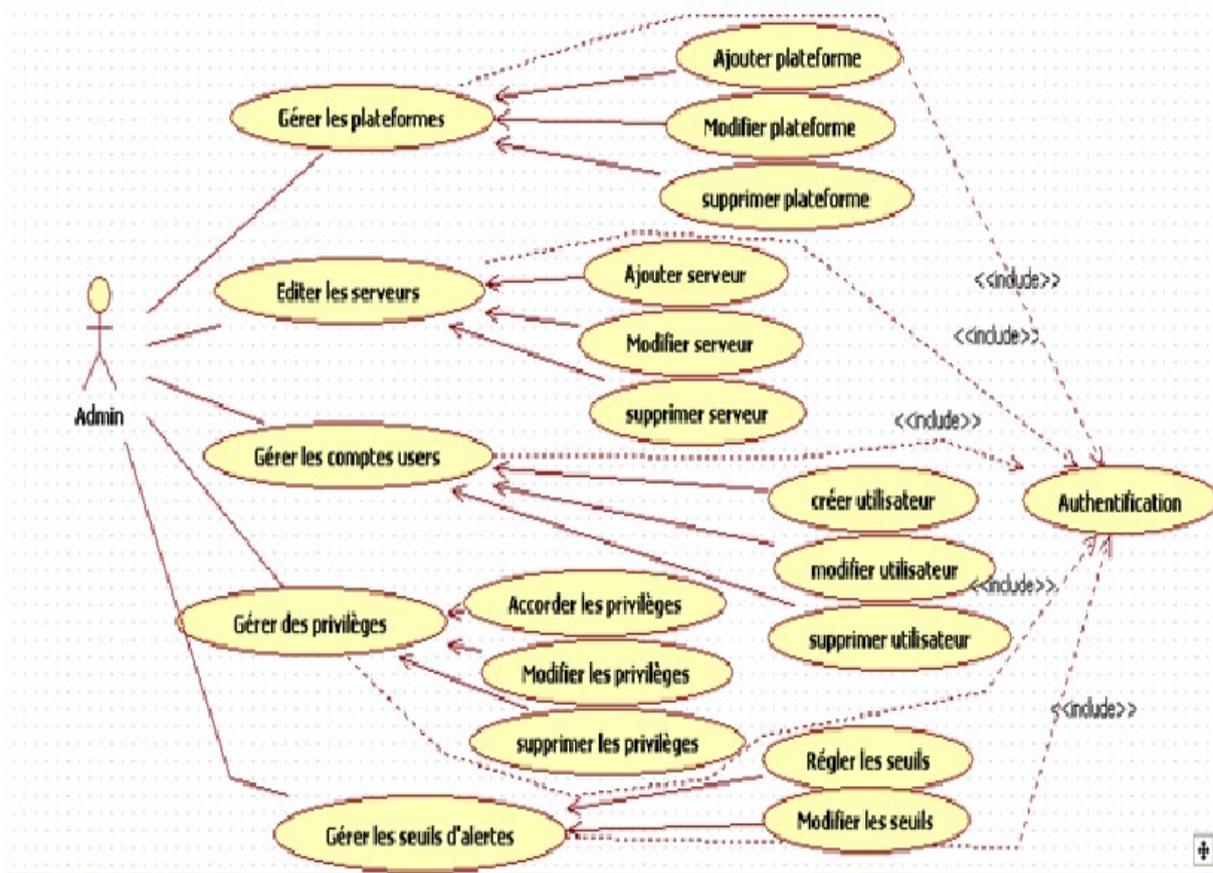


Diagramme des cas d'utilisation de l'administrateur

B. Diagramme des cas d'utilisation du superviseur

Ce diagramme donne à chaque employé la possibilité de vérifier l'état des serveurs. En plus, il a la possibilité d'élaborer des rapports. Chaque employé doit s'authentifier avant d'accéder à son groupe.

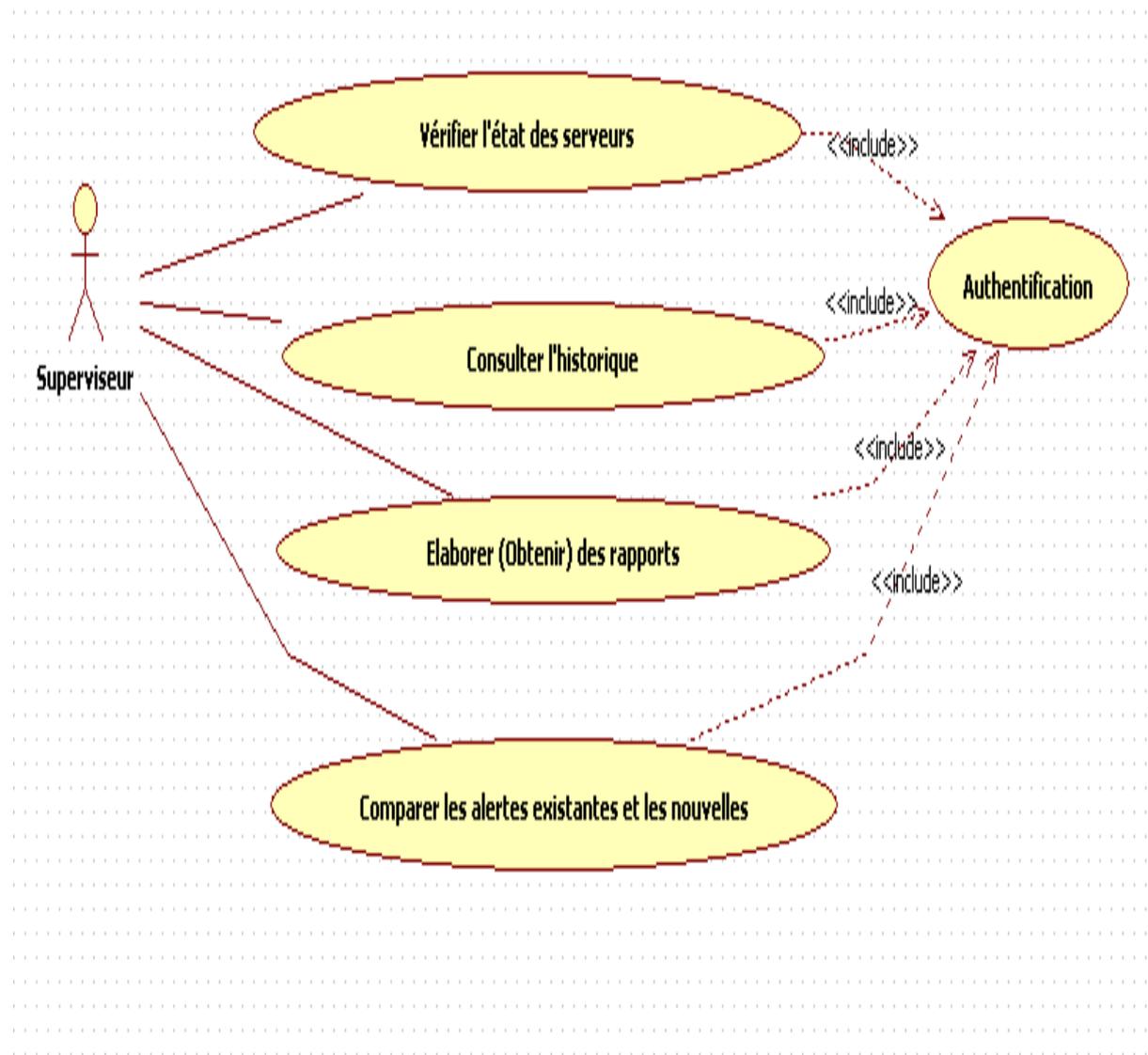


Diagramme des cas d'utilisation du superviseur

C. Diagramme des cas d'utilisation global de 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 au problème de la détermination et de la compréhension des besoins.

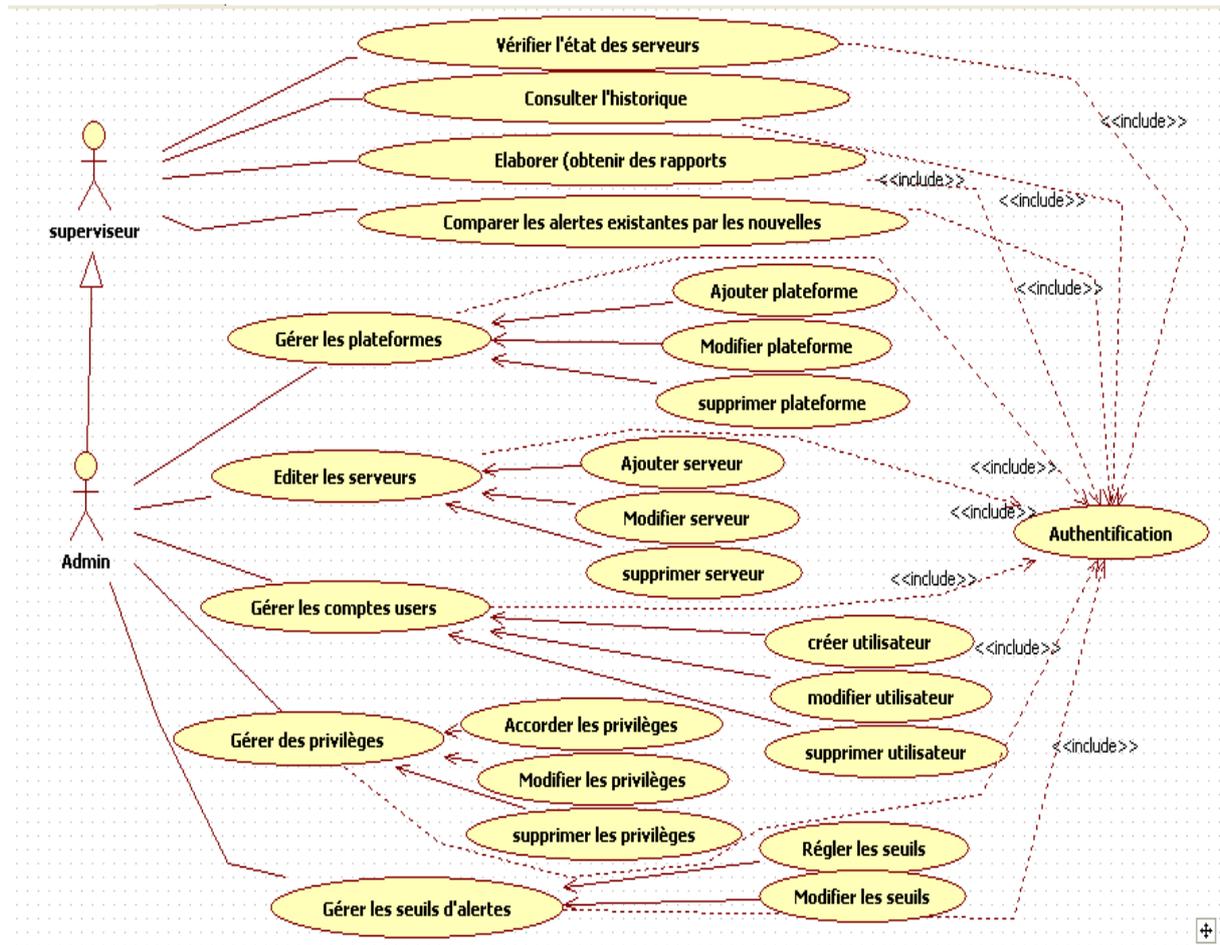


Diagramme des cas d'utilisation global

Dans ce diagramme, nous avons décrit les différentes fonctions effectuées par l'administrateur et le superviseur.

V. Conclusion

Ce chapitre précise les besoins fonctionnels que l'application développée doit offrir aux utilisateurs et les besoins non fonctionnels après l'identification des acteurs ainsi que les cas d'utilisations de chaque acteur.

Chapitre IV : Conception

I. Introduction

Ce chapitre présente la partie conception de notre projet. Vu que notre application se présente à la fois comme un système gérant des fonctionnalités complexes d'une part, et comme un produit multimédia d'autre part, nous présentons la conception en suivant trois grandes parties à savoir : conception générale, conception graphique et la conception détaillée de l'application.

II. Conception de l'application

La conception est un processus créatif, c'est la phase la plus importante dans le cycle de développement d'un projet. Le support de cette phase par des techniques et des outils appropriés est important pour produire une application de haute qualité et pour minimiser l'effort de développement.

III. Modélisation conceptuelle

Dans cette partie, nous allons présenter le résultat de l'étude que nous avons faite. On va étudier les besoins de notre application qui sont déjà précisés dans le chapitre précédent en se basant sur les diagrammes d'UML : Nous allons présenter en premier lieu la conception de la base de données sous forme d'un diagramme de classes et en deuxième lieu la conception des traitements sous forme des diagrammes de séquence.

A. Conception de la base de données en se basant sur le langage UML

Dans cette partie, nous allons modéliser la base de données sous forme d'un diagramme de classe.

1. Le diagramme de classe global de système

Le diagramme de classes constitue un élément très important de la modélisation : il permet de définir les composantes du système final. Néanmoins, on constate souvent qu'un diagramme de classes proprement réalisé permet de structurer le travail de développement de manière très efficace; il permet aussi, dans le cas des travaux réalisés en groupe (ce qui est pratiquement toujours le cas dans les milieux industriels), de séparer les composantes de manière à pouvoir répartir le travail de développement entre les membres du groupe

Le passage du diagramme de classe à la base de données nécessite le suivi de ces règles :

1. classe A <-> classe B avec cardinalité 1, 1 de deux coté

Dans ce cas, a priori, il vaut mieux regrouper le contenu de A et B dans une même table

2. classe A <-> classe B avec cardinalité 1, n de deux coté

C'est-à-dire qu'une entité 'a' peut avoir plusieurs entités 'b' qui lui sont attachées. Dans ce cas, vous allez créer 2 tables --> table A et table B. Dans la table B, tu prévois un attribut supplémentaire contenant l'ID de l'élément A auquel il est rattaché...

3. classe A <-> classe B avec cardinalité n, n de deux coté

Dans ce cas, il faut créer 3 tables :

Une table A, une table B, et une troisième table qui donne les "liens" entre les éléments a et les éléments b : cette table est constituée de 2 colonnes : l'une présente l'ID de la table A, et l'autre l'ID de la table B.

Tout ceci est très global, il y a des nuances en pratique.

Après la suivie des règles de passage du diagramme de classe vers le modèle relationnel, nous avons obtenu ce diagramme de classe :

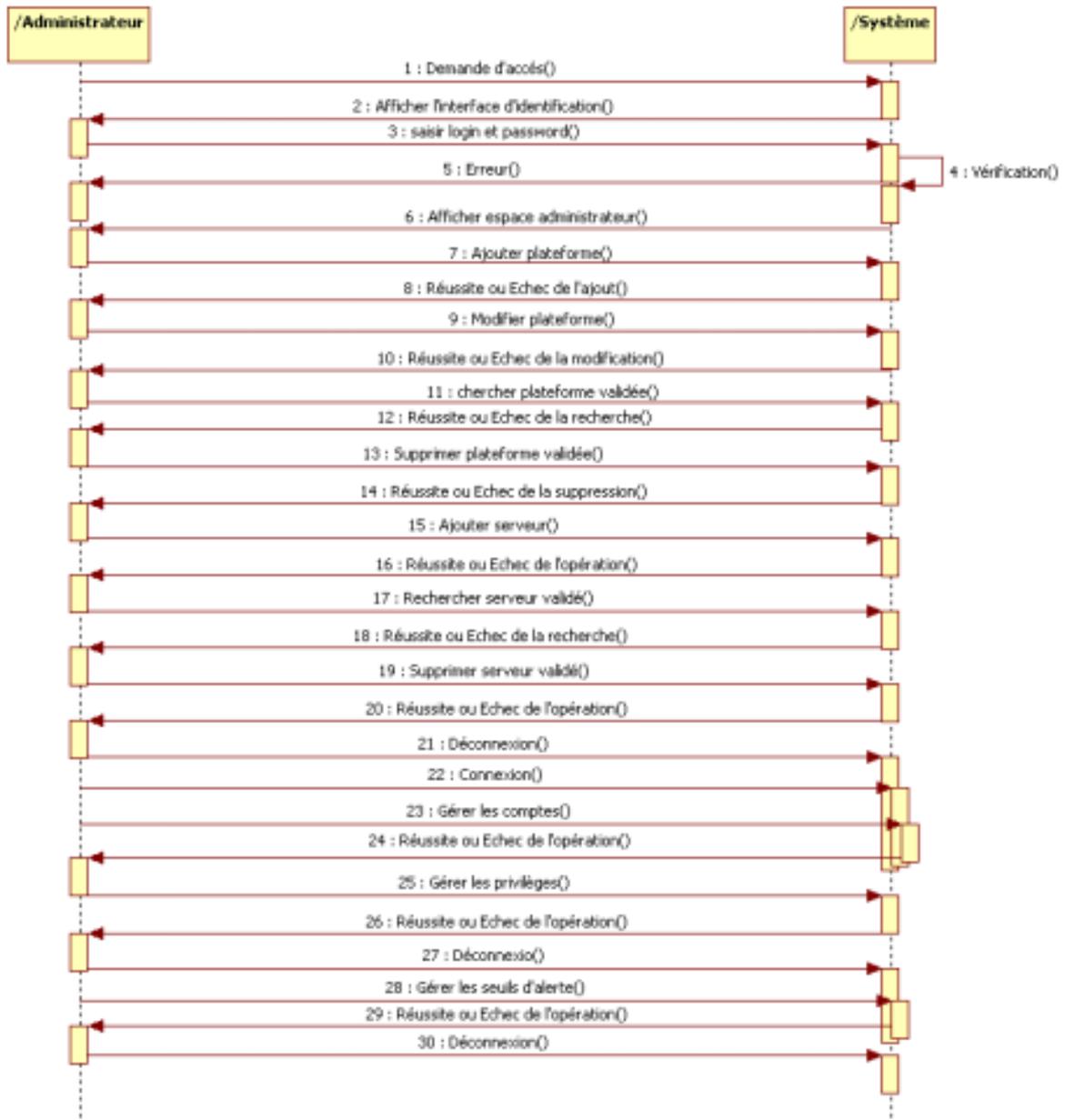


Diagramme de séquence de l'administrateur

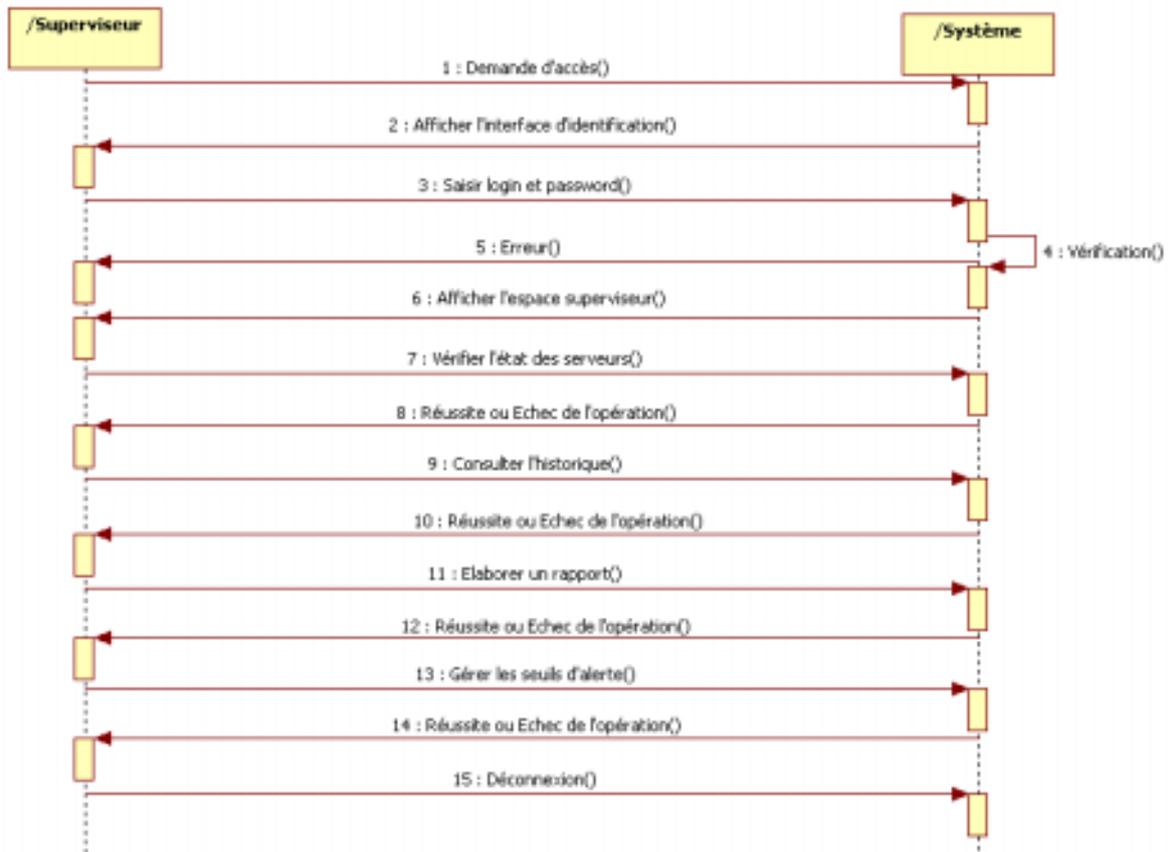


Diagramme de séquence de l'employé

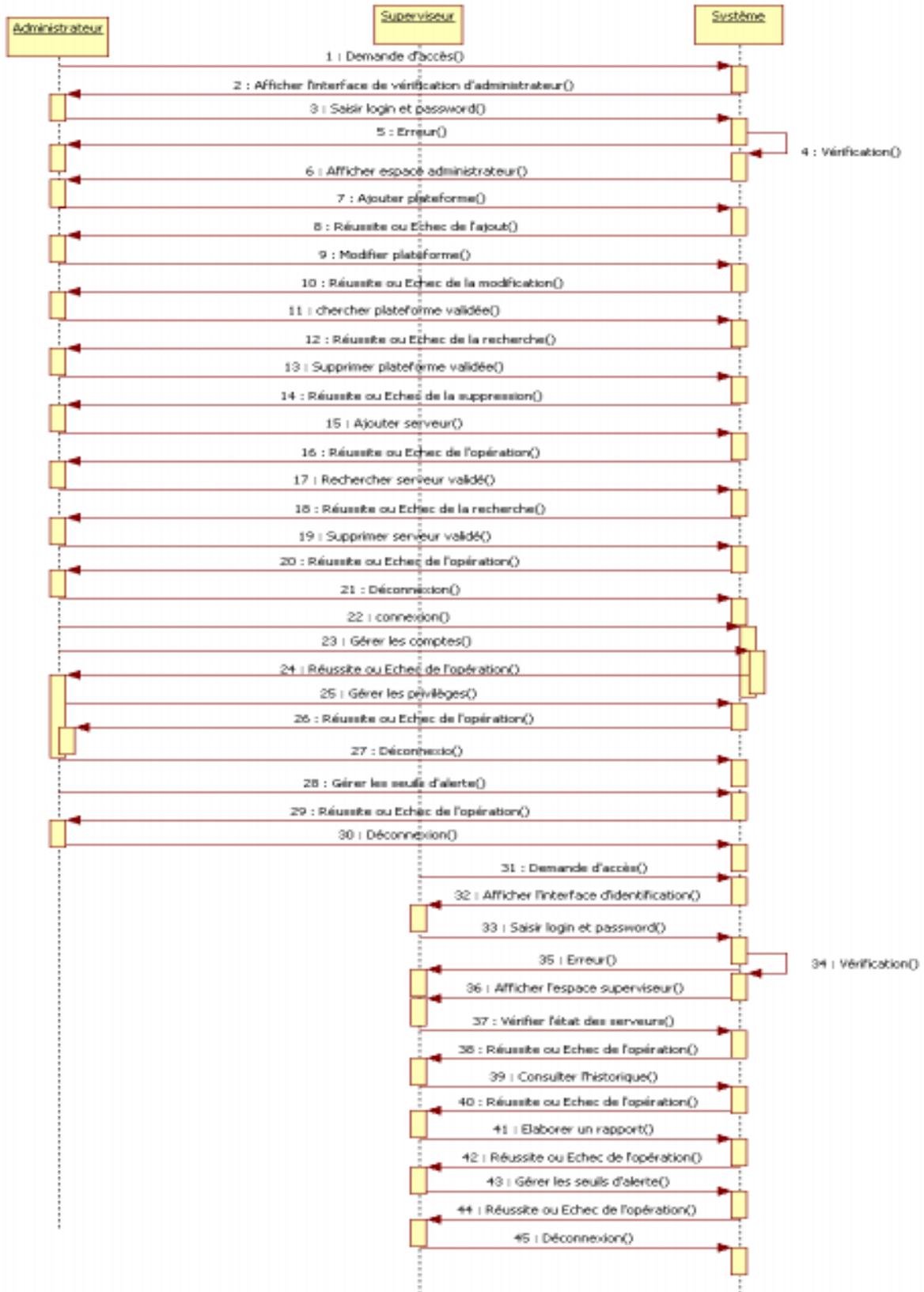


Diagramme de séquence global

IV. Conclusion

Après avoir accomplir la conception de notre application, nous allons entamer la partie "Réalisation". Dans le chapitre suivant, nous allons présenter l'environnement de travail, les outils de développement utilisés, ainsi que quelques imprimes écran des tests faits pour vérifier que notre système répond bien au cahier des charges.

Chapitre V : Réalisation

I. Présentation générale de l'application :

Notre application consiste à assurer la disponibilité et le suivi de la performance qui permet a une entreprise d'assurer la haute disponibilité et de la haute performance pour ces serveurs.

La capacité de gestion des performances comprend la surveillance des serveurs qui aideront les administrateurs informatiques à gérer convenablement leurs ressources.

II. Principe de Fonctionnement de l'application de la supervision :

Le schéma suivant montre le principe de fonctionnement de notre application de la supervision en mettant en valeur les principales connexions et relation entre les différents composants.

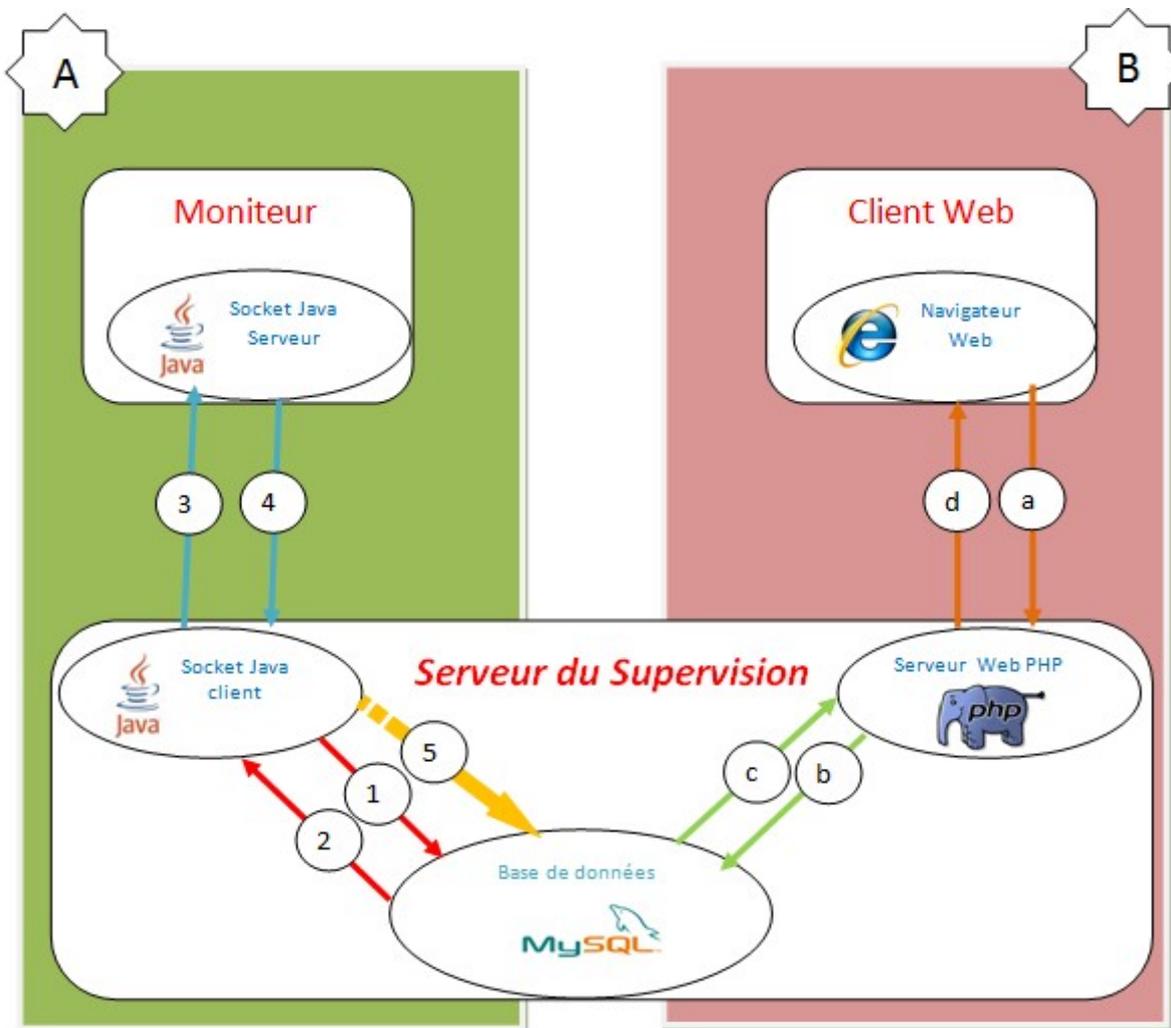


Schéma du principe de fonctionnement de l'application de la supervision

A. Description de la partie A :

1. Le client Socket-java demande qui est le client dont le serveur censé faire sa collecte
2. La réponse de la base de donnée à la requête « 1 »
3. Le client Socket-java contacte le serveur Socket –java du client en question pour la collecte d'information
4. Envoi des résultats des informations collectés du serveur Socket au client Socket
5. Insertion des informations collectées dans la base de données MySQL

B. Description de la partie B :

- a) Le client web se connecte au serveur web et demande de l'information collecté 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 du notre application de la supervision avec le langage PHP et d'interpréter graphiquement les données collectées.

IV. Description de la première partie

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.

A. notion sur java :

1. Qu'est-ce que la technologie java

Java est un langage de programmation et une plate-forme informatique créée par Sun Microsystems en 1995. Il s'agit de la technologie sous-jacente qui permet l'exécution de programmes dernier cri, notamment des utilitaires, des jeux et des applications professionnelles. Le langage Java est utilisé sur plus de 850 millions d'ordinateurs de bureau et un milliard de périphériques dans le monde, dont des périphériques mobiles et des systèmes de diffusion télévisuelle.

2. Pourquoi ai-je besoin de Java ?

Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé, et leur nombre grandit chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts !

3. Java est-il disponible gratuitement au téléchargement ?

Oui, vous pouvez télécharger Java gratuitement. Pour obtenir la dernière version, rendez-vous sur la page <http://java.com>.

4. Pourquoi devrais-je passer à la dernière version de Java ?

La dernière version de Java comprend d'importantes améliorations en termes de performance, de stabilité et de sécurité pour les applications Java exécutées sur votre ordinateur. L'installation de cette mise à jour gratuite vous permet de profiter pleinement d'applications Java toujours plus sécurisées et efficaces.

B. Concept de socket

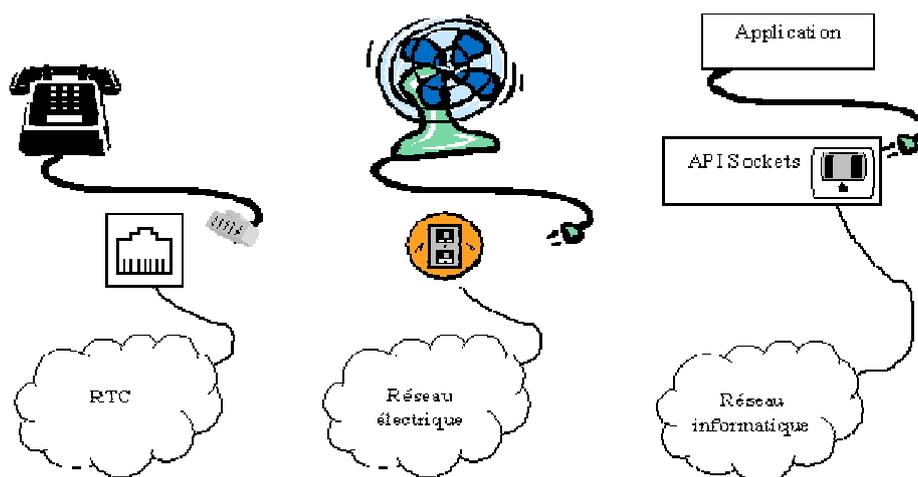
Les Sockets forment une API (Application Program Interface): ils offrent aux programmeurs une interface entre le programme d'application et les protocoles de communication. En aucun cas, les sockets ne forment une norme de communication ou une couche de protocole à l'instar de TCP/IP.

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 interprocess à l'intérieur d'un même système.

L'interface des "sockets" n'est pas liée à une pile de protocoles spécifique. Dans ce cours, 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 :



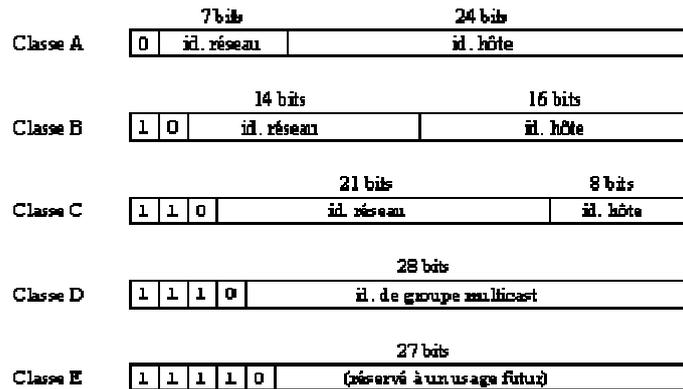
La métaphore des prises

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

C. 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.



Les classes d'adresses IP

1. La classe InetAddress

La classe `java.net.InetAddress` permet de représenter les adresses IP. Chaque objet de cette classe possède deux champs `hostName` et `address` contenant respectivement une chaîne de caractère et un tableau d'octets. Le champ `hostName` stocke le plus souvent le nom de l'hôte et le champ `address` l'adresse IP.

Cette classe ne possède pas de constructeur publics. Pour créer un objet de type `InetAddress` il faut donc utiliser l'une des méthodes suivantes :

```
public static InetAddress getByName (String nom_hote)
```

```
public static InetAddress [ ] getAllByName (String nom_hote)
```

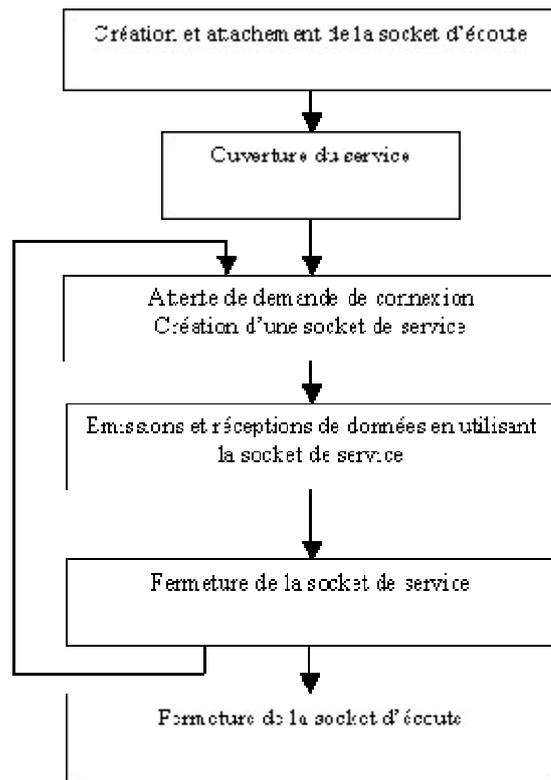
```
public static InetAddress getLocalHost ()
```

D. Sockets TCP

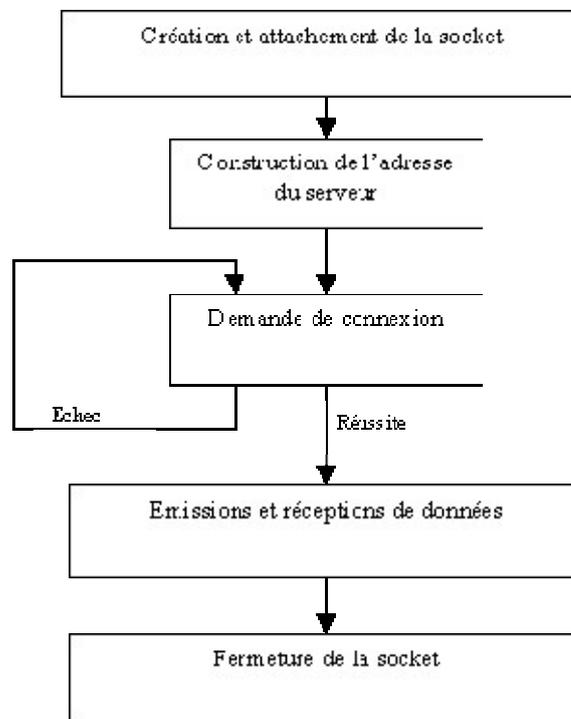
1. Le modèle client/serveur

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, attends des demandes de connexion qu'un processus, appelé client, lui envoie. Une fois l'étape d'établissement de connexion effectuée le fonctionnement redeviens 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 service que de clients.



Fonctionnement du serveur



Fonctionnement du client

2. *La classe Socket*

La classe Socket représente en Java les sockets utilisés côtés client ou les sockets de service.

a) **Constructeurs**

```
public Socket (String hote, int port) throws UnknownHostException, IOException
```

Ce constructeur crée un socket TCP et tente de se connecter sur le port indiqué de l'hôte visé. Le premier paramètre de ce constructeur représente le nom de la machine serveur. Si l'hôte est inconnu ou que le serveur de noms de domaine est inopérant, le constructeur générera une UnknownHostException. Les autres causes d'échec, qui déclenchent l'envoi d'une IOException sont multiples : machine cible refusant la connexion sur le port précisé ou sur tous les ports, problème lié à la connexion Internet, erreur de routage des paquets...

Voici un exemple d'utilisation de ce constructeur :

```
Socket leSocket = new Socket("www.irit.fr", 80);
```

```
public Socket (InetAddress adresse, int port) throws IOException
```

Ce constructeur fonctionne comme le premier, mais prends comme premier paramètre une instance de la classe InetAddress. Ce constructeur provoque une IOException lorsque la tentative de connexion échoue mais il ne renvoie pas d'UnknownHostException puisque cette information est connue lors de la création de l'objet InetAddress. `public Socket(String hote, int port, InetAddress adresseLocale, int portLocal) throws IOException`

b) **Méthodes informatives**

```
public InetAddress getInetAddress ()
```

```
public int getPort ()
```

Ces méthodes renvoient l'adresse Internet et le port distants auquel le socket est connecté.

```
public InetAddress getLocalAddress ()
```

```
public int getLocalPort ()
```

Ces méthodes renvoient l'adresse Internet et le port locaux que le socket utilise.

c) **Communication avec un socket**

```
public InputStream getInputStream () throws IOException
```

Cette méthode renvoie un flux d'entrées brutes grâce auquel un programme peut lire des informations à partir d'un socket. Il est d'usage de lier cet InputStream à un autre flux offrant d'avantage de fonctionnalités (un DataInputStream par exemple) avant d'acquérir les entrées.

Voici un exemple d'utilisation de cette méthode :

```
DataInputStream fluxEnEntree = new DataInputStream(leSocket.getInputStream());  
  
public OutputStream getOutputStream () throws IOException
```

Cette méthode renvoie un flux de sortie brutes grâce auquel un programme peut écrire des informations sur un socket. Il est d'usage de lier cet OutputStream à un autre flux offrant d'avantage de fonctionnalités (un DataOutputStream par exemple) avant d'émettre des données.

Voici un exemple d'utilisation de cette méthode :

```
DataOutputStream fluxEnSortie = new  
DataOutputStream(leSocket.getOutputStream());
```

d) Fermeture d'un socket

```
public void close() throws IOException
```

Bien que Java ferme tous les sockets ouverts lorsqu'un programme se termine ou bien lors d'un « garbage collect », il est fortement conseillé de fermer explicitement les sockets dont on n'a plus besoin à l'aide de la méthode close.

Une fois un socket fermé on peut toujours utiliser les méthodes informatives, par contre toute tentative de lecture ou écriture sur les « input/output streams » provoque une IOException.

e) Options des sockets

Java permet l'accès à un certain nombre d'options qui modifient le comportement par défaut des sockets. Ces options correspondent à celles que l'on manipule en C via ioctl (ou ioctlsocket avec Windows).

```
public boolean getTcpNoDelay() throws SocketException
```

```
public void setTcpNoDelay(boolean valide) throws SocketException
```

Si l'on programme une application très interactive on pourra valider l'option TCP_NODELAY.

```
public int getSoLinger() throws SocketException
```

```
public void setSoLinger(boolean valide, int secondes) throws SocketException
```

3. *La classe ServerSocket*

Cette classe permet de créer des sockets qui attendent des connexions sur un port spécifié et lors d'une connexion retournent un Socket qui permet de communiquer avec l'appelant.

a) **Constructeurs**

```
public ServerSocket (int port) throws IOException
```

Ce constructeur crée un socket serveur qui attendra les connexions sur le port spécifié. Lorsque l'entier port vaut 0, le port est sélectionné par le système. Ces ports anonymes sont peu utilisés car le client doit connaître à l'avance le numéro du port de destination. Il faut donc un mécanisme, comme le portmapper des RPC, qui permet d'obtenir ce numéro de port à partir d'une autre information.

b) **Accepter et clore une connexion**

```
public Socket accept () throws IOException
```

Cette méthode bloque l'exécution du programme serveur dans l'attente d'une demande de connexion d'un client. Elle renvoie un objet Socket une fois la connexion établie. Si vous ne voulez pas bloquer l'exécution du programme il suffit de placer l'appel à accepté dans un thread spécifique.

```
public void close () throws IOException
```

Cette méthode ferme le socket serveur en libérant le port. Les sockets serveurs sont eux aussi fermés automatiquement par le système à la fermeture de l'application.

c) **Méthodes informatives**

```
public InetAddress getInetAddress ()
```

```
public int getLocalPort ()
```

Ces méthodes renvoient l'adresse Internet et le port locaux sur lequel le socket attend les connexions.

4. **Options des sockets serveurs**

Seule l'option SO_TIMEOUT est disponible pour les sockets serveurs.

```
public int getSoTimeout() throws SocketException
```

`public void setSoTimeout(int ms) throws SocketException`

Par défaut, l'appel à `accept()` se bloque jusqu'à la réception d'une demande de connexion. Lorsque `SO_TIMEOUT` est initialisée, cette attente ne dépasse pas le temps imparti, exprimé en millisecondes. Tout dépassement de temps se solde par une `InterruptedException`. Le socket reste malgré tout connecté.

La valeur par défaut est 0 qui signifie, ici, un laps de temps infini ce qui convient à la plupart des serveurs, conçus en général pour s'exécuter indéfiniment. L'option doit être validée avant l'appel à `accept()` pour être prise en compte.

Le paramètre doit être ≥ 0 sans quoi une exception `SocketException` est générée.

V. Description de la deuxième partie

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.

A. Présentation :

MySQL est un système de gestion de base de données (SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle et Microsoft SQL Server.

MySQL AB a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains¹. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle Database et MySQL. Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010^{2,3}.

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

B. Caractéristiques

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multiutilisateurs.

C'est un logiciel libre développé sous double licence en fonction de l'utilisation qui en est faite : dans un produit libre ou dans un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la licence publique générale GNU (GPL) qui s'applique. Ce type de licence double est utilisé par d'autres produits comme le Framework de développement de logiciels Qt (pour les versions antérieures à la 4.5).

C. Systèmes d'exploitation supportés

MySQL fonctionne sur de nombreux systèmes d'exploitation différents, incluant AIX, IBM i-5, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetWare, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows (2000, XP, Vista et 7).

Les bases de données sont accessibles en utilisant les langages de programmation C, C++, VB, VB .NET, C#, Delphi/Kylix, Eiffel, Java, Perl, PHP, Python, Ruby et Tcl ; une API spécifique est disponible pour chacun d'entre eux. Une interface ODBC appelée MyODBC est aussi disponible. En Java, MySQL peut être utilisé de façon transparente avec le standard JDO.

Depuis le rachat de MySQL AB par Sun Microsystems, MySQL est devenu officiellement la base de données à utiliser conjointement avec le langage de programmation Java, ce qui donne une notoriété supplémentaire au SGBD auprès des entreprises utilisant Java.

D. Utilisation

MySQL fait partie du quatuor LAMP : Linux, Apache, MySQL, PHP. Il appartient également à ses variantes WAMP (Windows) et MAMP (Mac).

Le couple PHP/MySQL est très utilisé par les sites Web et proposé par la majorité des hébergeurs Web. Plus de la moitié des sites Web fonctionnent sous Apache, qui est le plus souvent utilisé conjointement avec PHP et MySQL.

Wikipédia utilise MySQL, ainsi que de nombreuses entreprises, dont Google, Yahoo!, YouTube, Adobe, Airbus, Alstom, Crédit agricole, AFP, Reuters, BBC News, Ernst & Young, Alcatel-Lucent. Cependant certaines de ces entreprises ont préféré développer ou utiliser des systèmes de bases de données différents et plus efficaces à grande échelle, comme par exemple BigTable pour Google.

Depuis la version 5, il est possible d'utiliser le PL/SQL originellement développé par Oracle Corporation afin d'utiliser des procédures et fonctions stockées ainsi que des déclencheurs. Par ailleurs, MySQL supporte la norme SQL2 (utilisation des JOIN), ce qui fait de lui un SGBD sûr puisque la conformité à cette norme garantira sa compatibilité avec les requêtes normalisées.

Son absence par défaut de support des transactions et de l'intégrité automatique des tables (sauf en utilisant certains moteurs comme InnoDB) rendent son adoption plus compliquée par exemple dans les sociétés bancaires, cependant, ses performances généralement meilleures que la majorité des autres systèmes concurrents, et son prix d'implantation nettement inférieur, lui permettent d'obtenir un certain succès auprès des entreprises ayant besoin d'une base de données ayant un bon ratio performance prix.

L'intérêt d'utiliser les outils du www en remplacement du client serveur 2 ou 3 tiers, a poussé certaines banques telles que le Crédit mutuel, le Crédit agricole, le LCL, la Banque de développement du Canada à proposer MySQL au catalogue de leurs produits informatiques internes.

E. Dénomination

MySQL est l'œuvre d'une société suédoise, MySQL AB, fondée par David Axmark, Allan Larsson et Michael "Monty" Widenius.

Le nom MySQL vient de la combinaison de My, fille du co-fondateur Michael "Monty" Widenius, avec l'acronyme SQL (selon la documentation de la dernière version en anglais).

D'autre part, le dossier base et une grande partie des bibliothèques utilisées par les développeurs utilisent le préfixe My.

Le nom du logo de MySQL (le dauphin) Sakila, a été choisi par les créateurs de MySQL sur la base d'un concours. Il a été proposé par Ambrose Twebaze, développeur du Swaziland. D'après Ambrose, le nom Sakila puise ses origines du siswati, la langue locale du Swaziland.

F. Historique

La première version de MySQL est apparue le 23 mai 1995. Il a d'abord été créé pour un usage personnel à partir de mSQL en s'appuyant sur le langage de bas niveau ISAM qu'ils trouvaient trop lent et trop rigide. Ils ont créé une nouvelle interface SQL en gardant la même API que mSQL.

MySQL est passé en licence GPL à partir de la version 3.23.19 (juin 2000)

- Version 4.0 : première version en octobre 2001, stable depuis mars 2003
- Version 4.1 : première version en avril 2003, stable depuis octobre 2004
- Version 5.0 : première version en décembre 2003, stable depuis octobre 2005
- Version 5.1 : première version en novembre 2005, Release Candidate distribuée depuis septembre 2007
- Version 5.2 : distribuée en avant-première (ajout du nouveau moteur de stockage Falcon) en février 2007, cette ligne a ensuite été renommée 6.0
- Version 5.4 :
- Version 6.0 : première version alpha en avril 2007

En 2003, MySQL AB et SAP AG concluent un accord. Le résultat s'appellera MaxDB. Il est issu de l'intégration du système de SAP (SAP DB) dans MySQL.

Le 20 avril 2009, Oracle Corporation annonce racheter Sun Microsystems pour 7,4 milliards de dollars⁷, créant la crainte de voir MySQL disparaître – ou du moins ne plus être développé – au profit d'Oracle, le SGBD d'Oracle Corporation. MySQL est cependant loin d'avoir toutes les fonctionnalités d'Oracle, et pourrait donc être vu au contraire par la société comme un produit gratuit d'appel banalisant l'usage de SQL et préparant donc ses futures ventes.

G. Quelque nation sur MySQL

MySQL dérive directement de SQL (Structured Query Language) qui est un langage de requête vers les bases de données exploitant le modèle relationnel.

Il en reprend la syntaxe mais n'en conserve pas toute la puissance puisque de nombreuses fonctionnalités de SQL n'apparaissent pas dans MySQL (sélections imbriquées, clés étrangères...)

Le serveur de base de données MySQL est très souvent utilisé avec le langage de création de pages web dynamiques : PHP.

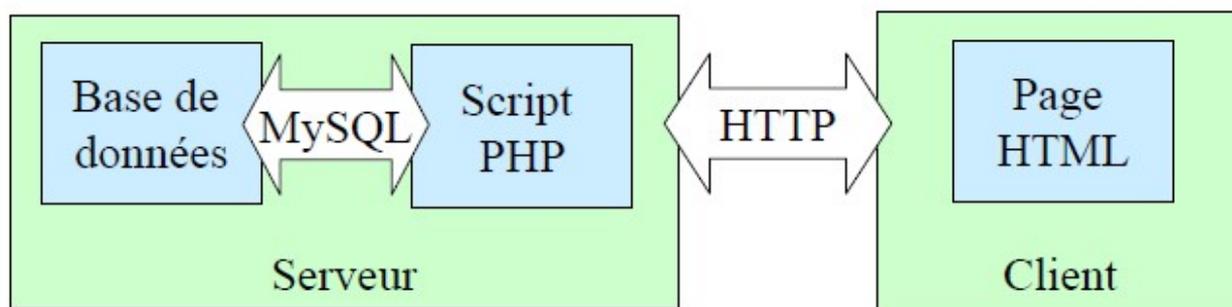


Schéma explicatif du fonctionnement du MySQL

VI. Description de la troisième partie

La troisième partie consiste à créer l'interface web de notre application de supervision avec le langage PHP et d'interpréter graphiquement les données collectées.

A. notion sur PHP :

1. Qu'est-ce que PHP?

PHP est un langage interprété (un langage de script) exécuté du côté serveur (comme les scripts CGI, ASP, ...) et non du côté client (un script écrit en Javascript ou une applet Java s'exécute sur votre ordinateur...). La syntaxe du langage provient de celles du langage C, du Perl et de Java. Ses principaux atouts sont :

- Une grande communauté de développeurs partageant des centaines de milliers d'exemples de script PHP ;
- La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL) ;
- La simplicité d'écriture de scripts ;
- La possibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGI, pour lesquels il faut écrire des lignes de code pour afficher chaque ligne en langage HTML) ;
- La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur de nombreuses plateformes : Unix, Linux, Windows, MacOS X, Solaris, etc...) ;
- L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, etc.).

2. Origines de PHP

Le langage PHP a été mis au point au début d'automne 1994 par Rasmus Lerdorf. Ce langage de script lui permettait de conserver la trace des utilisateurs venant consulter son CV sur son site, grâce à l'accès à une base de données par l'intermédiaire de requêtes SQL. Ainsi, étant donné que de nombreux internautes lui demandèrent ce programme, Rasmus Lerdorf mit en ligne en 1995 la première version de ce programme qu'il baptisa *Personal Sommaire Page Tools*, puis *Personal Home Page v1.0* (traduisez *page personnelle version 1.0*).

Etant donné le succès de PHP 1.0, Rasmus Lerdorf décida d'améliorer ce langage en y intégrant des structures plus avancées telles que des boucles, des structures conditionnelles, et y intégra un package permettant d'interpréter les formulaires qu'il avait développé (*FI*, Form Interpreter) ainsi que le support de mSQL. C'est de cette façon que la version 2 du langage, baptisée pour l'occasion *PHP/FI version 2*, vit le jour durant l'été 1995. Il fut rapidement utilisé sur de nombreux sites (15000 fin 1996, puis 50000 en milieu d'année 1997).

A partir de 1997, Zeev Suraski et Andi Gurmans rejoignirent Rasmus pour former une équipe de programmeurs afin de mettre au point PHP 3 (Stig Bakken, Shane Caraveo et Jim Winstead les rejoignirent par la suite). C'est ainsi que la version 3.0 de PHP fut disponible le 6 juin 1998.

A la fin de l'année 1999 la version 4.0 de PHP, baptisée PHP4, est apparue. PHP en est aujourd'hui à sa cinquième version.

3. SGBD supportés par PHP

PHP permet un interfaçage simple avec de nombreux systèmes de gestion de bases de données (SGBD), parmi lesquels :

- Adabas D
- dBase
- Empress
- FilePro
- Informix
- Interbase
- mSQL
- MySQL
- Oracle
- PostgreSQL
- Solid
- Sybase
- Velocis
- Unix dbm

4. Implantation du code Php

a) L'interprétation du code par le serveur

Un script PHP est un simple fichier texte contenant des instructions écrites à l'aide de caractères ASCII 7 bits (des caractères non accentués) incluses dans un code HTML à l'aide

de balises spéciales et stocké sur le serveur. Ce fichier doit avoir l'extension « .php » pour pouvoir être interprété par le serveur. Ainsi, lorsqu'un navigateur (le client) désire accéder à une page dynamique réalisé en PHP :

- le serveur reconnaît l'extension d'un fichier PHP et le transmet à l'interpréteur PHP
- Dès que l'interpréteur rencontre une balise indiquant que les lignes suivantes sont du code PHP, il ne lit plus les instructions: il les exécute!
- L'interpréteur exécute l'instruction puis envoie les sorties éventuelles au serveur
- A la fin du script, le serveur transmet le résultat au client (le navigateur).

Un script PHP est interprété par le serveur, les utilisateurs ne peuvent donc pas voir le code source!

Le code PHP stocké sur le serveur n'est donc jamais visible directement par le client puisque dès qu'il en demande l'accès, le serveur l'interprète! De cette façon aucune modification n'est à apporter sur les navigateurs.

b) Implantation au sein du code HTML

Pour que le script soit interprété par le serveur deux conditions sont nécessaires :

- Le fichier contenant le code doit avoir l'extension telle que *.php* et non *.html*
- Le code PHP contenu dans le code HTML doit être délimité par des balises du type `<? et ?>`

Un script PHP doit :

- comporter l'extension *.php*
- être imbriqué entre les délimiteurs `<? et ?>`

Pour des raisons de conformité avec certaines normes (XML et ASP par exemple), plusieurs balises peuvent être utilisées pour délimiter un code PHP :

```
<ol>
<li><? et ?></li>
<li><?php et ?></li>
<li><script language="php"> et </script></li>
<li><%php et %></li>
</ol>
```

c) Un exemple de script simple

Voici ci-dessous l'exemple classique de script PHP :

```
<html>
<head><title>Exemple</title></head>
<body>
<?php
    echo "Hello world";
?>
</body>
</html>
```

On notera bien évidemment que la fonction *echo* permet d'afficher sur le navigateur la chaîne délimitée par les guillemets.

d) *Caractéristiques du langage Php*

✓ **L'interprétation du code**

Un code PHP (celui compris entre les délimiteurs `<?php` et `?>`) est un ensemble d'instructions se terminant chacune par un point-virgule (comme en langage C). Lorsque le code est interprété, les espaces, retours chariot et tabulation ne sont pas pris en compte par le serveur. Il est tout de même conseillé d'en mettre (ce n'est pas parce qu'ils ne sont pas interprétés que l'on ne peut pas les utiliser) afin de rendre le code plus lisible (pour vous, puisque les utilisateurs ne peuvent lire le code source: il est interprété).

✓ **Les commentaires**

Une autre façon de rendre le code plus compréhensible consiste à insérer des commentaires, des lignes qui seront tout simplement ignorées par le serveur lors de l'interprétation. Pour ce faire, il est possible, comme en langage C, d'utiliser des balises qui vont permettre de délimiter les explications afin que l'interpréteur les ignore et passe directement à la suite du fichier.

Ces délimiteurs sont `/*` et `*/`. Un commentaire sera donc noté de la façon suivante :

```
/* Voici un commentaire! */
```

Il y a toutefois quelques règles à respecter :

- Les commentaires peuvent être placés n'importe où à l'intérieur des délimiteurs de script PHP
- Les commentaires ne peuvent contenir le délimiteur de fin de commentaire (`*/`)
- Les commentaires ne peuvent être imbriqués
- Les commentaires peuvent être écrits sur plusieurs lignes
- Les commentaires ne peuvent pas couper un mot du code en deux

Il est possible aussi d'utiliser un type de commentaire permettant de mettre toute la fin d'une ligne en commentaire en utilisant le double *slash* (`//`). Tout ce qui se situe à droite de ce symbole sera mis en commentaire.

✓ **Typologie**

La manière d'écrire les choses en langage PHP a son importance. Le langage PHP est par exemple sensible à la casse (en anglais *case sensitive*), cela signifie qu'un nom contenant des majuscules est différent du même nom écrit en minuscules. Toutefois, cette règle ne s'applique pas aux fonctions, les spécifications du langage PHP précisent que la fonction *print* peut être appelée `print()`, `Print()` ou `RINT()`.

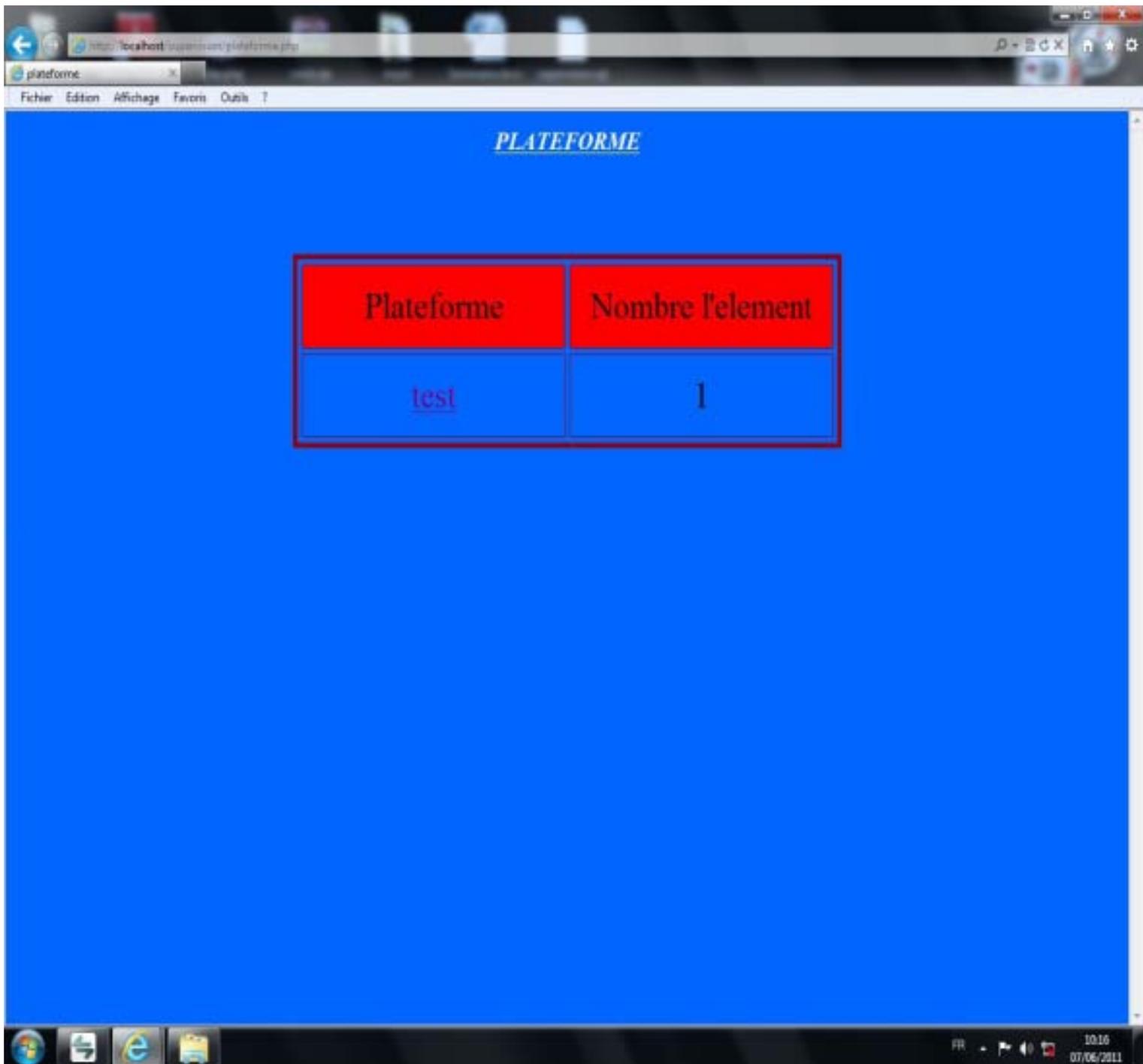
Enfin, toute instruction se termine par un point-virgule.

VII. Réalisation de Projet :

Comme présenté précédemment, notre application consiste à collecter et interpréter les données des machines clientes pour assurer la supervision et par la suite la haute disponibilité.

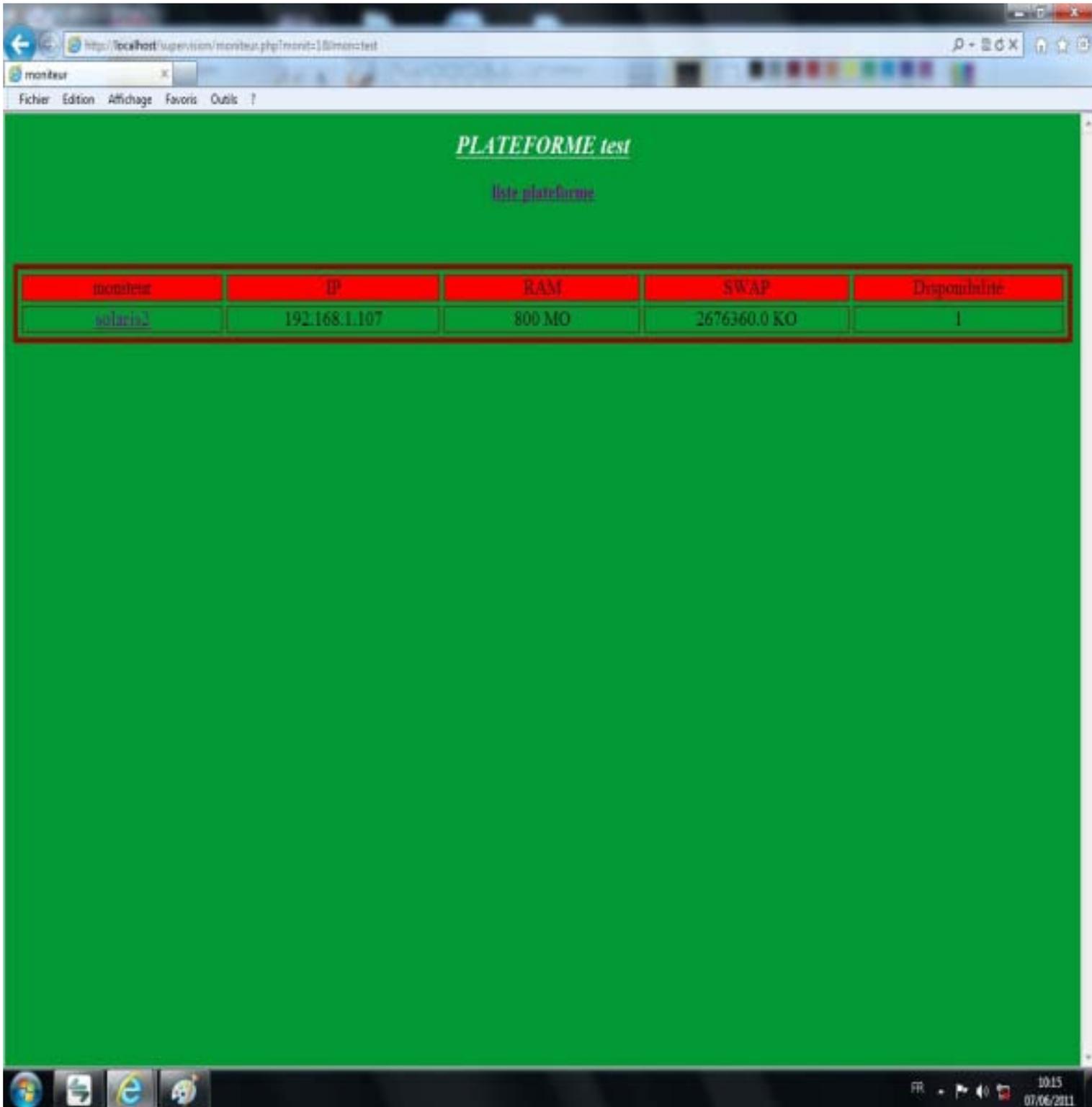
Pour accéder à notre application de la supervision, il suffit d'ouvrir le navigateur web avec l'adresse suivante :

<http://172.16.100.206/supervision/plateforme.php>



Après la connexion à notre application, on aura la liste des toutes les plateformes à superviser. Dans notre exemple on a utilisé la plateforme qui s'appelle *test* avec un seul serveur Sun Solaris 10.

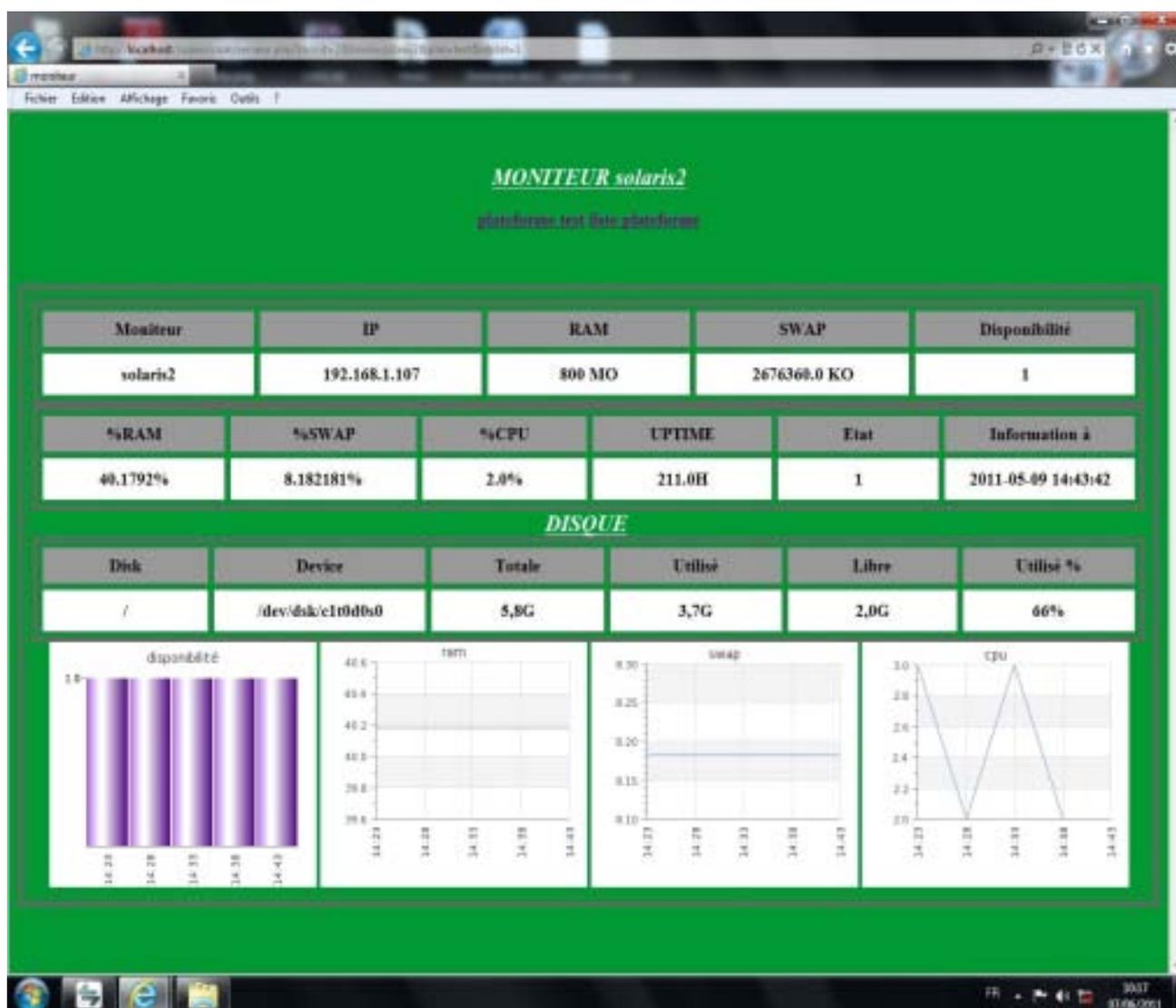
Pour accéder à la plateforme en question, il suffit du cliquer sur son nom. Ensuite, on aura la page web suivante :



Cette deuxième page contient la liste des serveurs de la plateforme *test* avec les détails suivants :

- ✓ **Moniteur** : le nom du serveur
- ✓ **IP** : l'adresse IP du serveur
- ✓ **RAM** : la quantité totale de la RAM physique sur le serveur
- ✓ **SWAP** : la quantité du mémoire virtuelle disponible au système
→ La mémoire swap est un mécanisme de mémoire virtuelle. Le but de ce mécanisme est de permettre à des applications d'utiliser plus de mémoire vive en cas de besoin.
- ✓ **Disponibilité** : Permet d'indiquer si le serveur est 'online' lorsqu'elle contient le nombre 1 et offline si le nombre est 0

Pour avoir plus du détail sur notre serveur, il suffit de cliquer sur le nom du serveur en question. Dans notre exemple on clique sur **solaris2**. On aura alors la page web suivante :



Cette page contient en haut le mon du MONITEUR qui est le nom du serveur à superviser qui est dans notre exemple *solaris2*

Le détail de la machine solaris2 est comme suit :

Moniteur : c'est le mon du serveur à superviser

IP: c'est l'adresse IP du serveur

RAM: c'est la quantité totale de la mémoire physique du serveur

SWAP: c'est la quantité de la mémoire virtuelle sur le serveur

Disponibilité: c'est l'état du serveur si c'est 1 alors il est online et si la disponible est 0 alors le serveur est offline

%RAM : c'est le pourcentage d'utilisation de la RAM à l'instant de collecte

%SWAP : c'est le pourcentage d'utilisation de la mémoire virtuelle SWAP à l'instant de collecte

%CPU: c'est le pourcentage d'utilisation de la CPU à l'instant de collecte

UPTIME: c'est la durée du temps pendant la quel le serveur a été opération (online) depuis le dernier redémarrage

Etat: c'est la disponible du serveur ; si Etat est 1 alors le serveur est online et 0 c'est offline

Information à : c'est la date du dernier collecte des informations

Disk : c'est le nom de la partition disponible sur le client

Device: c'est le mon physique partition dans le système Solaris

Totale: c'est la capacité du partition

Utilisé: c'est la capacité en Go du disk utilisé actuellement

Libre: c'est la capacité en Go du disk non utiliser

Utilisé%: c'est le pourcentage d'utilisation de la partition à l'instant de collecte

Le graphe de la disponibilité : permet d'affiche graphiquement les donner collecter concernant la duré de la disponibilité du ce serveur pour les 30 derniers minutes

Le graphe de la ram: permet d'affiche graphiquement les donner collecter concernant le pourcentage de mémoire utiliser par ce serveur pour les 30 derniers minutes

Le graphe du swap: permet d'affiche graphiquement les donner collecter concernant le pourcentage de mémoire swap utiliser par ce serveur pour les 30 derniers minutes

Le graphe de la cpu: permet d'affiche graphiquement les donner collecter concernant le pourcentage du cpu utiliser par ce serveur pour les 30 derniers minutes

Conclusion Générale

Ce projet de fin d'études dans la société du Tunisie Telecom, nous a permis de pratiquer tous ce que nous avons étudiés à l'université virtuelle de point du vue pratique. Nous avons pu créer une application très puissante et aussi très stable qui permet la supervision des différents serveurs et plateformes.

Ce projet nous a permis de connaître des nouveaux langages de programmation qui étaient complètement inconnu pour nous par exemple le PHP, MySQL. Il nous a permis aussi d'apprendre à utiliser divers logiciels et systèmes d'exploitation.

Nous sommes pleinement satisfait de ce projet qui nous a enrichi nos connaissances et nous a permis de manipuler des nouveaux concepts. Nous avons pu acquérir des nouvelles techniques de conception et de développement des applications.

Enfin ce projet peut être enrichi dans le futur par d'autres fonctionnalités, par exemple :

- la supervision des applications spécifique,
- la supervision du cluster,
- la supervision d'un script bien précis.

Résumé :

Le présent travail, effectué au sien de l'Université Virtuelle de Tunis « UVT », s'inscrit dans le cadre du projet de fin d'étude pour l'obtention du diplôme Licence Appliquée en Sciences de Technique et de l'Information et de Communication.

L'objectif de ce projet est de développer une solution de la supervision en temps réel des serveurs et plateforme.

Les langages de programmation utilisée dans notre projet sont : Java, MySQL et PHP. Nous avons recours, le long de ce projet au Model conceptuelle UML. L'environnement du travail est Sun Solaris 10

Mots Clés :

Java, MySQL, PHP, Supervision, Sun Solaris10.

Abstract :

This work , conducted within the Virtual University of Tunis (UVT), is part of final project study for the diploma in Degree in Applied Science Technology and Information and Communication.

The objective of this project is to develop a solution for real-time monitoring of servers and platform.

Programming languages used in our project are: Java, MySQL and PHP. We use along the project conceptual model UML. The working environment is Sun Solaris 10.

Key Word:

Java, MySQL, PHP, Monitoring, Sun Solaris 10.