******

**TPI 2017**

*Travail professionnel individuel*

*Informaticien-ne CFC*

**StreamScan**

### Table des matières

1 Les grandes lignes du projet 2

1.1 Analyse de l’état actuel 3

1.2 Analyse de l’état désiré 3

1.3 Cahier des charges / exigences du système 3

2 Analyse préliminaire 4

2.1 Objectifs du système 4

2.2 Variantes 4

2.3 Choix de variante 4

2.4 Rentabilité 5

2.5 Analyse de risque 5

2.6 Sécurité de l’information et protection des données 5

3 Concept 6

3.1 Exigences du système 6

3.2 Architecture du système 6

3.3 Plan d’intégration des systèmes 6

3.4 Concept d‘implémentation 6

3.5 Concept de formation 6

3.6 Concept de tests 6

3.7 Moyens nécessaires 6

4 Réalisation 7

4.1 Spécifications détaillées 7

4.2 Design du système 7

4.3 Configuration xyz 7

5 Test 8

5.1 Procédure de test 8

5.2 Protocol de test 8

5.3 Signature du protocole de test 8

6 Liste des sources et références 9

7 Glossaire 10

8 Signatures 11

9 Annexes 12

# Résumé du rapport du TPI

Le Management Summary est une présentation conceptuelle du travail effectué et du résultat attendu d’au maximum une page A4 qui permettent au lecteur une rapide compréhension du rapport de travail.

# Introduction

StreamScan est une application permettant l’obtention et l’enregistrement de diverses données matérielles et logicielles depuis une application serveur lancée sur la machine dont on souhaite extraire les informations.

Le produit StreamX, développé par Infoteam, est une passerelle de communication pour différents protocoles. Il est installé sur plusieurs types de machines (MOXA, Beckhoff,…), sous différents OS (Windows XP, Windows 7, Linux,…) et est configuré selon le client.

L’installateur de StreamX est mis à jour régulièrement et mis à disposition des clients sur la plate-forme d’échange afin qu’ils aient accès aux dernières fonctionnalités.

L’application **StreamScan** sera utilisée afin de facilité l’obtention des informations des machines ainsi que leurs version de StreamX.

L’application va permettre de :

* Reproduire des situations sur la plate-forme de tests dans un environnement similaire à l’environnement des clients
* Identifier les installations à mettre à jour
* Montrer au client son acquis et son évolution potentielle
* Gagner du temps lors d’intervention chez les clients

## Analyse de la situation initiale

En l’état actuel, aucun moyen ne permet de connaître ces informations de manière centralisée ce qui engendre une perte de temps lors d’interventions.

La partie serveur, installée sur chaque passerelle de communication, a déjà été réalisée en prérequis au travail, afin d’acquérir les connaissances nécessaires à la réalisation de ce projet.

Celle-ci utilise diverses librairies afin de récupérer les informations système (OS, nom du produit, version,…), matérielles (nom du produit, fabriquant,…), de la carte réseau (IP, passerelle, masque de sous-réseau,…) et les informations relatives à **StreamX**. Le client peut alors s’y connecter afin de récupérer ces informations.

Pour plus d’informations concernant la partie serveur, se référer à la documentation de cette dernière (cf. : Annexes).

## Analyse de l’état désiré

Le produit fini permettre d’obtenir les informations d’une machine StreamX grâce à son IP et stockant les données dans une base de données MySQL.

Celles-ci sont ensuite listées selon l’entreprise et l’ouvrage de la machine. Un filtre (recherche) peut être appliqué permettant ainsi une meilleure lisibilité. Les situations critiques sont mis en évidence (ex : ancienne version).

Les informations peuvent être mises à jour en sélectionnant la machine.

A quoi doit ressembler la solution/le produit fini? Vous pouvez éventuellement illustrer le nouveau processus.

## Cahier des charges / exigences du système

Voici les exigences de l’application :

* Obtention des informations d’une machine StreamX grâce à son IP
* Sauvegarde des informations récoltées dans une base de données MySQL selon l’ouvrage et l’entreprise
* Présentation des informations de la base de données sur une interface web en mettant en évidence certaines situations critiques comme un disque plein ou une version ancienne du produit.
* Recherche des informations d’une machine selon un critère
* Développement de l’application en ASP.NET
* Accès à l’application géré par une page de login
* Rédaction d’un manuel utilisateur, en plus de la documentation technique
* Réalisation de l’application en anglais
* Le cahier des charges **complet** est disponible en annexes

Cahier des charges détaillé découlant du mandat de projet. Définition exacte et mesurable des objectifs à atteindre.

Décrire et structurer les exigences du système désiré.

# Analyse

## Objectifs

Les objectifs sont les mêmes que spécifiés dans le cahier des charges, à savoir :

* Pouvoir obtenir les infos d’une machine StreamX
* Stocker ces infos dans la base de données
* Récupérer l’ensemble des infos des machines et les afficher au client
* Permettre à ce dernier d’effectuer une recherche sur ces infos
* Ajout/modification/suppression d’une entreprise/ouvrage
* Réaliser une documentation utilisateur

## Rentabilité

Le projet étant interne à l’entreprise et non pas pour un client, le critère de rentabilité ne rentre pas en compte. Il doit cependant être réalisé avant la fin du stage.

Quelle est l’utilité économique du projet?

## Analyse de risque

Quels risques sont engendrés en cas de non réalisation du projet? Que se passe-t-il en cas d’échec du projet ? Quels sont les risques les plus importants du projet ?

## Fonctionnalités

Depuis le cahier des charges, on peut définir les fonctionnalités suivantes :

* Obtention des infos d’une machine StreamX (scan)
* Voir la liste des machines StreamX à partir de la base de données en fonction de l’entreprise et de l’ouvrage
* Mettre à jour les informations de la base de données à partir d’un scan

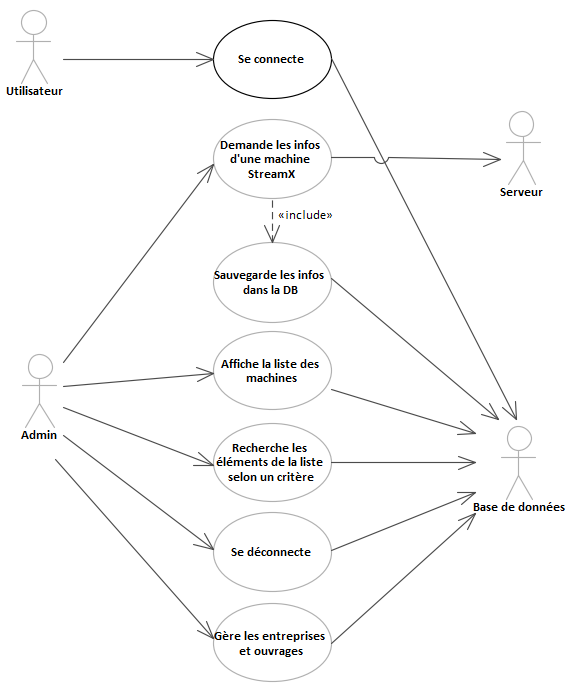
## Planification des étapes du développement

La partie développement sera organisée dans l’ordre suivant :

1. Structuration de l’application (création des pages, méthodes, fonctionnalités globales)
2. Partie authentification (page de connexion)
3. Partie récupération des infos d’une machine StreamX (scan) + Tests fonctionnels
4. Partie ajout, modification et suppression des entreprises/ouvrages
5. Partie mise à jour des données du scan dans la base de données
6. Partie récupération des informations des machines à partir de la base de données
7. Tests fonctionnels de l’application

## Diagramme d’utilisation

L’outil « Enterprise Architect » a été utilisé car il est simple et c’est l’outil que l’on utilise à l’école.



**Acteurs**

Utilisateur Utilisateur non-authentifié

Admin Utilisateur authentifié

Serveur Serveur de l’application, disposé sur la machine StreamX

Base de données Base de données MySQL de l’application

**Cas d’utilisation**

Se connecte

Connecte l’utilisateur en vérifiant ses identifiants dans la base de données

Demande les infos d’une machine StreamX

Demande au serveur ses informations

Affiche la liste des machines

Affiche la liste des machines en contactant la base de données

Recherche les éléments de la liste selon un critère

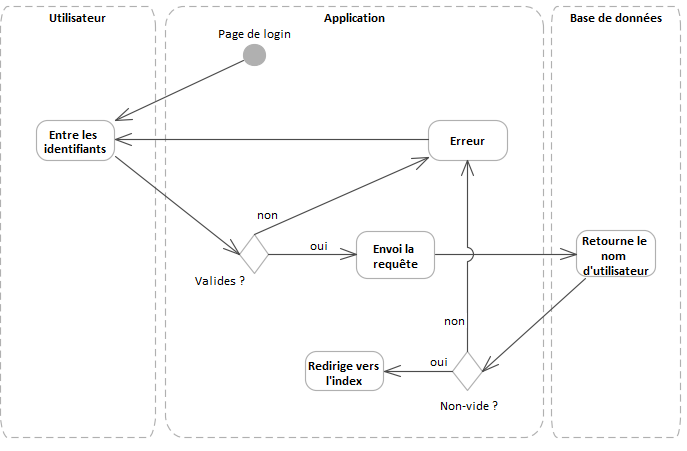
Affiche les éléments de la base de données selon un critère spécifié par l’admin

Se déconnecte

Déconnecte l’admin de l’application

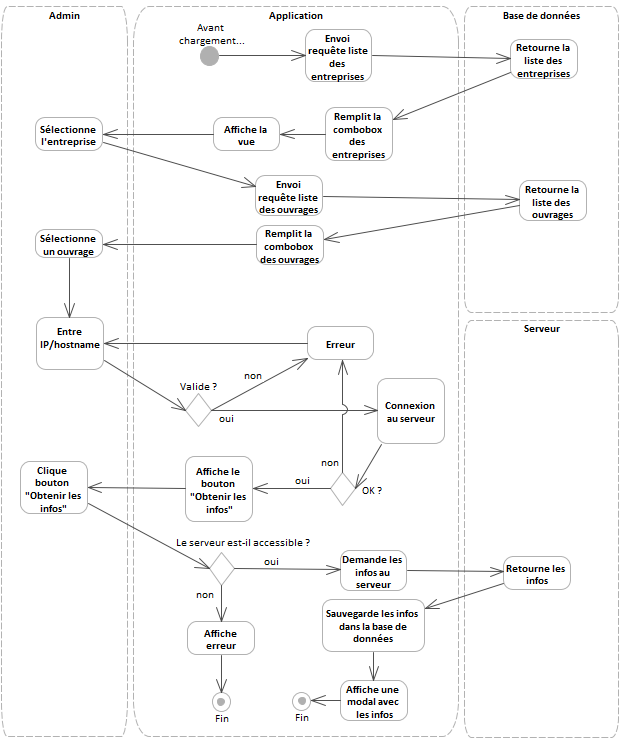
## Diagrammes d’activité

### Se connecte



Suite au chargement de la page, l’utilisateur saisie ses identifiants. S’ils sont invalides, on affiche un erreur- Dans le cas contraire, une requête à la base de données afin de contrôle l’existence de l’utilisateur. Elle retourne le nom de l’utilisateur en question. Si celui-ci est vide, on dit à l’utilisateur que ses identifiants sont incorrects. Sinon, on le redirige vers l’index.

### Demande les infos d’une machine StreamX



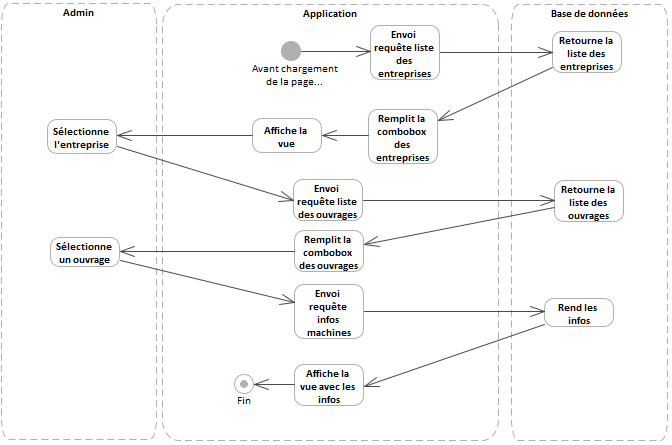
Premièrement, la liste des entreprises est obtenue à partir de la base de données. Elle est ensuite appliquée à la combobox des entreprises et on affiche la vue.

Le client sélectionne l’entreprise voulue et valide. La liste des ouvrages est ensuite récupérée et est ajoutée à la combobox des ouvrages.

Le client sélectionne alors un ouvrage ainsi que l’adresse de la machine dont il souhaite récupérer les informations. Si l’adresse entrée est valide, on contrôle que le serveur soit accessible. Si ce n’est pas le cas, on avertit l’admin via une erreur. Dans le cas contraire, on affiche le bouton « Obtenir les infos ».

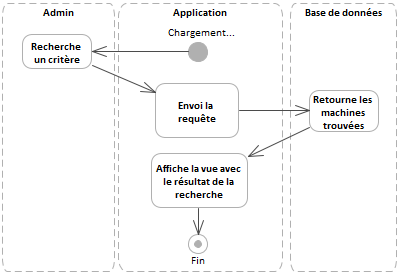
En cliquant sur le bouton, on vérifie que le serveur soit toujours accessible et si c’est le cas, on envoie la requête au serveur. Les infos sont ensuite sauvegardées dans la base de données puis on affiche fenêtre modal avec les infos.

### Affiche la liste des machines



Avant le chargement de la page, on récupère la liste des entreprises qu’on met dans une combobox puis on affiche la vue. L’admin sélectionne l’entreprise. Les ouvrages de cette entreprise sont récupérés depuis la base de données puis on remplit une autre combobox. Suite à la sélection de l’ouvrage, on récupère les machines correspondantes et on les affiche.

### Recherche les éléments de la liste selon un critère

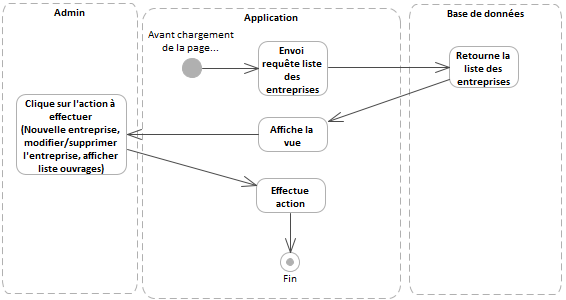


Suite au chargement de la liste des machines, le client peut rechercher un critère. En le validant, la requête est envoyée au serveur qui retourne les machines correspondantes au critère. La vue est ensuite mise à jour pour correspondre au résultat.

### Se déconnecte

Aucun diagramme d’activité n’a été conçu en raison de sa très grande simplicité.

### Gestion des entreprises et des ouvrages

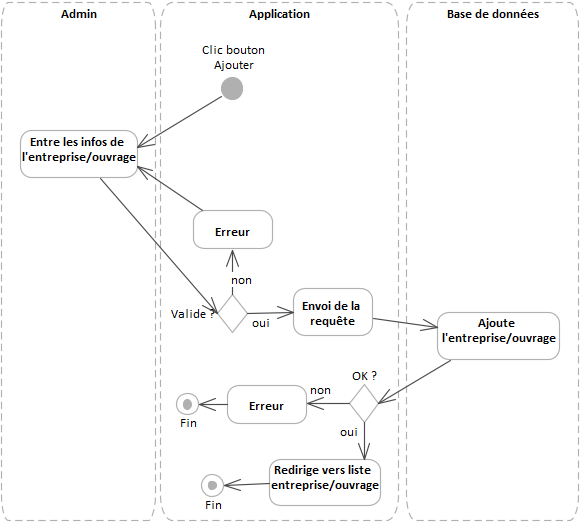


Premièrement, on récupère la liste des entreprises et on affiche la vue. Le client peut ensuite effectuer plusieurs actions :

* Ajouter une nouvelle entreprise
* Modifier une entreprise existante
* Supprimer une entreprise existante
* Afficher les ouvrages correspondants à une entreprise

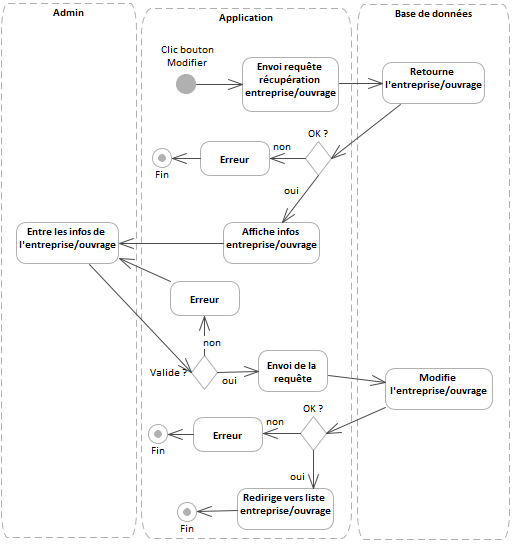
Note : La dernière action affiche une vue similaire à celle-ci, c’est pourquoi aucun diagramme n’y est consacré.

#### Ajoute une entreprise/ouvrage



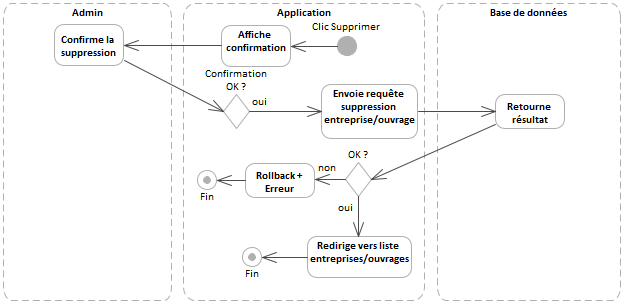
Lorsque l’admin appuie sur le bouton « Ajouter », celui-ci doit entrer les infos de l’entreprise/ouvrage et valide. Si les données entrées sont valides, alors la requête est envoyée à la base de données qui ajoute l’entreprise/ouvrage. Si tout c’est bien passé, l’admin est redirigé vers la liste des entreprises/ouvrages.

#### Modifie une entreprise/ouvrage



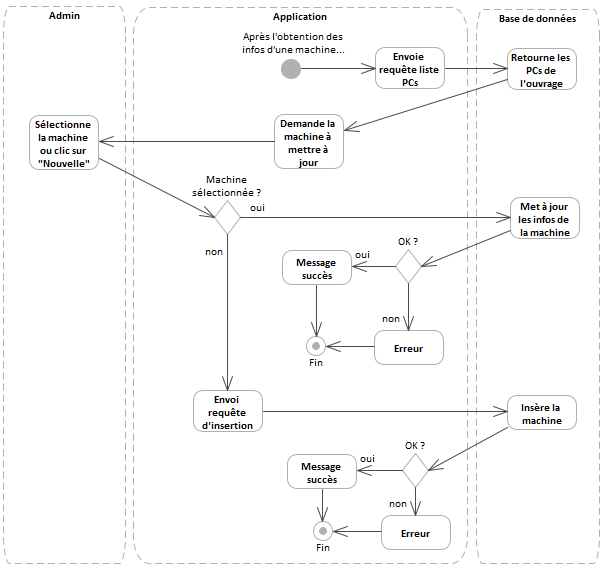
Lorsque l’admin appuie sur le bouton « Modifier », une requête est envoyée à la base de données afin de récupérer les infos de l’entreprise/ouvrage. L’utilisateur modifie les infos de l’entreprise/ouvrage et valide. Si les données entrées sont valides, alors la requête est envoyée à la base de données qui ajoute l’entreprise/ouvrage. Si tout c’est bien passé, l’admin est redirigé vers la liste des entreprises/ouvrages.

#### Supprime une entreprise/ouvrage



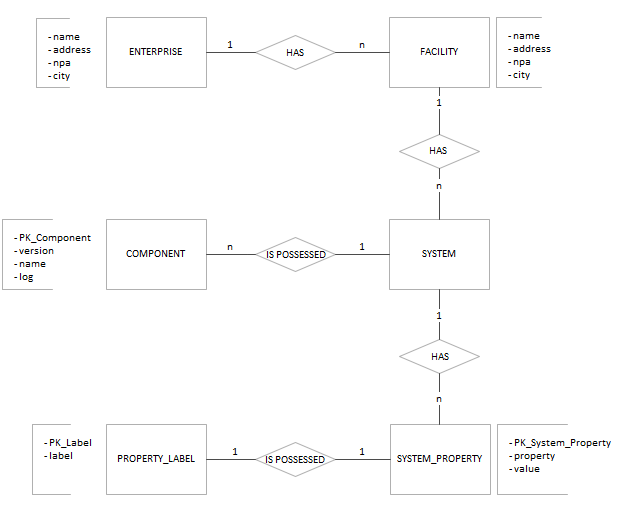
Lorsque l’admin appuie sur le bouton « Supprimer », une fenêtre de confirmation est affichée. Si l’admin appuie sur « OK », on envoie la requête de suppression au serveur. Si elle s’est correctement déroulée, on affiche la liste des entreprises/ouvrages. Sinon on fait un « rollback » de la requête et on affiche une erreur.

### Sauvegarde des informations d’une machine StreamX dans la base de données



Après l’obtention des infos d’une machine, on récupère tous les PCs de l’ouvrage. Puis, on demande au client quelle machine il veut mettre à jour ou s’il souhaite en ajouter une nouvelle. Dans le premier cas un envoie une requête « INSERT » à la base de données et dans le deuxième cas, une requête « UPDATE ».

## Modèle de Conception de Données (MCD)



Ici, un modèle dit « optimisé » a été choisi. Les tables « Enterprise », « Facility » et « Component » sont modélisées selon la manière « classique ». Tandis qu’un PC est modélisé sous deux tables, à savoir : « System » et « System\_Property ». La raison à cela est qu’il est facile d’imaginer l’ajout d’autres propriétés pour un PC. Cette solution a été choisie afin de rendre cet ajout plus simple et de ne pas avoir à changer la structure de la base de données.

Une entreprise (Enterprise) a plusieurs ouvrages (Facility) mais un ouvrage ne correspond qu’à une seule entreprise.

Un ouvrage a plusieurs PC (System) mais un PC n’appartient qu’à un seul ouvrage.

Un PC a plusieurs composants (Component) mais un composant n’est possédé que par un seul PC.

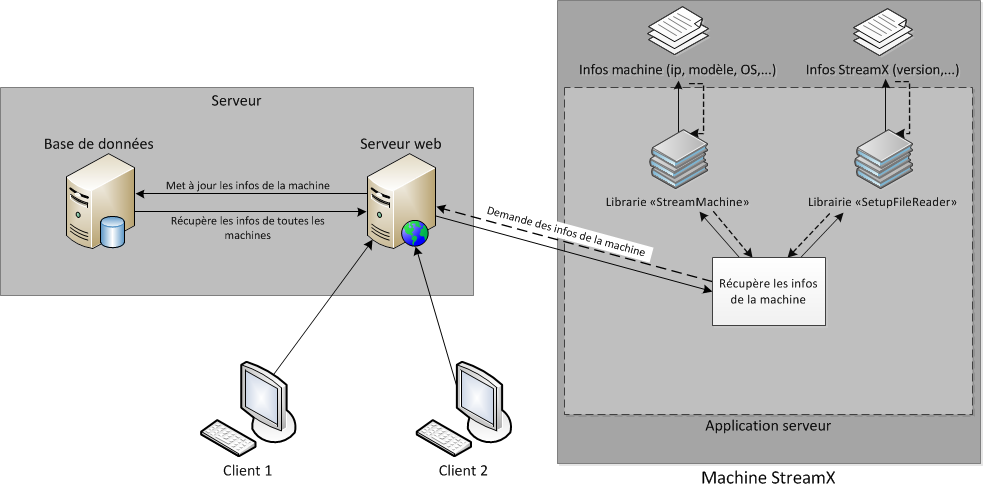
Un PC a plusieurs propriétés (System\_Property) mais une propriété ne caractérise qu’un seul PC.

Une propriété ne possède qu’un label et un label n’est possédé que par une seule propriété.

# Conception

## Exigences du système

## Architecture du système



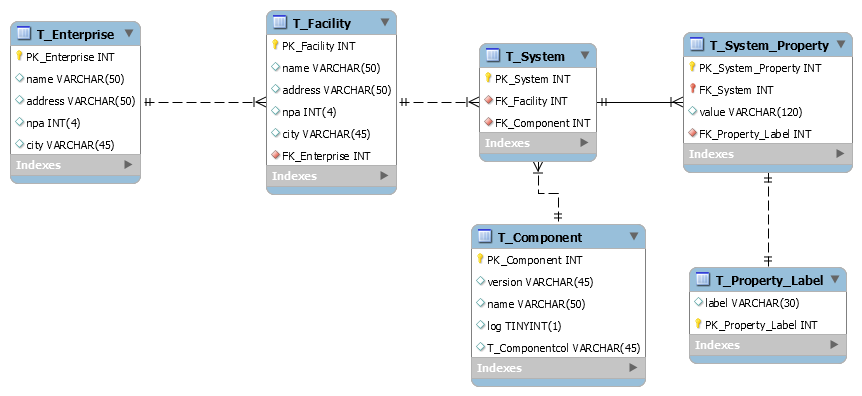
L’application cliente est composée d’un serveur web ainsi que d’une base de données.

La base de données, conformément au MCD, s’occupe de stocker les entreprises, les ouvrages ainsi que les machines StreamX. Elles seront mises à disposition de l’application afin de les afficher.

Le serveur web héberge l’application cliente StreamScan. Il récupère les données d’une machine StreamX, en contactant l’application serveur StreamScan disposée sur cette dernière, et les stocke dans la base de données.

L’application serveur utilise plusieurs librairies afin de récupérer les informations de la machine et de les transmettre au client.

## Modèle Logique de Données (MLD)



À noter que dans la table « T\_System\_Property », on définit la « FK\_System » et la « FK\_Property\_Label » en tant que PK afin qu’un « PC » puisse avoir plusieurs propriétés mais qu’une seule de chaque type.

### Longueur des types

Enterprise, Facility

Name, address Longueur semblant adéquate pour un nom d’entreprise/ouvrage ou une adresse

Npa Un NPA ne dépasse jamais 4 caractères

City Définit en prenant le nom de ville le plus long en France

System Property

Value La valeur pouvant être un texte, une longueur conséquente a été prise

Component

Version Longueur semblant adéquate pour une version, ne sachant pas encore le nombre de digit la composant

Name Longueur générale

Log Ne peut accueillir qu’un entier de longueur 1 (= booléen)

Property Label

label Longueur semblant adéquate pour un label

## Plan d’intégration des systèmes

## Concept d‘implémentation

## Concept de formation

## Concept de tests

Comment les tests sont-ils effectués? Blackbox, Whitebox … .

## Moyens nécessaires

Conseil: D’autres chapitres peuvent s’ajouter à cette partie en fonction du projet.

# Réalisation

## Spécifications détaillées

## Design du système

## Configuration xyz

Conseil: D’autres chapitres peuvent s’ajouter à cette partie en fonction du projet.

# Test

## Procédure de test

Représentation du processus de test..Qui teste quoi à l’aide de quels moyens ?

## Protocol de test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Objet testé | Description du test | Attente | Résultat | Visa |
| 1.0 |  |  |  |  |  |

Protocole de test détaillé avec Visa.

## Signature du protocole de test

|  |  |  |
| --- | --- | --- |
| Date | Nom | Signature |
|  |  |  |

# Bibliographie: liste des sources et références

Conseil: Cette partie peut contenir une liste des sources et éléments littéraires utilisés.

# Glossaire trier par ordre alphabetique

|  |  |
| --- | --- |
| Terme | Signification |
| StreamX | Passerelle de communication développée par Infoteam |
|  |  |
|  |  |

# Signatures

Je soussigné déclare que les informations contenues dans ce rapport de travail pratique individuel rendu ce jour le XX.XX.20XX dans le cadre de la procédure de qualification de mon CFC d’informaticien/-ne, ne sont pas plagiées. Toutes les informations de sources extérieures ainsi que les informations fournies par des tiers durant le déroulement du travail sont consignées.

L’apprenti/-e doit signer la documentation avant la remise afin de témoigner de la justesse des informations qui y figurent.

|  |  |  |
| --- | --- | --- |
| Date | Nom | Signature |
|  |  |  |

# Annexes

Code imprimé, protocole de séances, etc.: spécifier complètement et annexer à la documentation.

Listings de scripts et programmes. Le travail fourni par la candidate ou le candidat doit être entièrement documenté. Omettre le code généré automatiquement s’il n’est pas nécessaire à la compréhension.

Dans le cas où des manuels ont été créés, vous pouvez les ajouter ici en tant qu’annexe.

## Première visite des experts

Les points suivants ont étés notés suites à la première visite des experts :

* Garder une trace dans le journal d’un changement, remarque d’un expert
* Le journal ne doit pas dépasser 120 heures (8h/jour)
* Bien noter s’il y a un changement dans le cahier des charges
* Journal : Précision à l’heure (pas plus précis)