|  |
| --- |
| http://www.jobmania.fr/fichiers/ecoles_universites/58-58-communication_so_image_logo-toulouse-iii_v1.0.jpg |
| Rapport du projet de Java |
| Projet Java STRI |
|  |
| **Lamoure – Le Govic – Maury - Loulier** |
| **26/05/2015** |

|  |
| --- |
| ***D:\Users\Josselyn\hubiC\svgStriOnline\Documents communs\Logos\STRI-logo-2011.jpg*** |

Table des matières

[1. Demande formulée 3](#_Toc420502176)

[2. Réponse apportée 3](#_Toc420502177)

[a. Phase préliminaire 3](#_Toc420502178)

[b. Phase de modélisation 3](#_Toc420502179)

[c. Phase de réalisation 4](#_Toc420502180)

[d. Phase de rédaction 5](#_Toc420502181)

[3. Répartition des tâches 5](#_Toc420502182)

[4. Fin du projet 5](#_Toc420502183)

1. Demande formulée

La société STRI désire la mise en place d’un service de gestion d’appareils connectés à son réseau. Cette société possède différents locaux et chaque local possède différentes salles.

Les besoins de STRI sont les suivants :

* Ajouter des entités telles que des locaux, des salles, des appareils, des interfaces et des systèmes d’exploitation.
* Chaque entité possède des informations qui lui sont propres :
  + Exemple : Pour un appareil, il faudra lui indiquer un nom, un OS, une version de cet OS, sa localisation.
* Faire des liens entre les différentes entités :
  + Exemple : un appareil est ajouté dans une salle qui est dépendante d’un local.
* Possibilité de désactiver un appareil sur le réseau et visualiser l’impact :
  + Exemple : Un switch est coupé, tous les ordinateurs reliés à ce switch ne sont plus connectés au réseau.
* Modifier les liens entre les entités et modifier les informations de chaque entité.

1. Réponse apportée

Pour répondre à ces besoins, il a été décidé de développer un logiciel en Java.

Il y aura différentes étapes décrites ci-après.

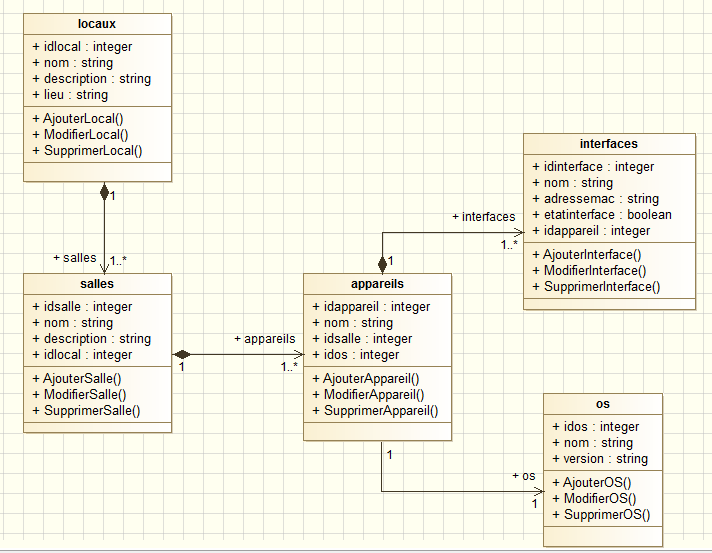
1. Phase préliminaire

Le projet de développement d’un service de gestion d’appareils connectés à un réseau sera fait sur la plateforme Github. Chaque participant découvrira et apprendra à utiliser Github pour un travail collaboratif car Github est un système de gestion de version de code.

|  |
| --- |
| <https://github.com/josselynL/JAVA> |

1. Phase de modélisation

Il est impératif de définir les classes nécessaires à la réalisation des besoins définis :



Depuis ce diagramme, la base de données a été conçue est qui sera importé sur un hébergeur en ligne sous Postgre : « Alwaysdata » :

|  |
| --- |
| Accès : https://phppgadmin.alwaysdata.com/ |
| Login : projetjavastri  Mot de passe : projetjavaSTRI |

1. Phase de réalisation

Dans cette phase, il faut veiller à n’oublier aucunes fonctionnalités. L’utilisateur devra pouvoir accéder à tout ce qui lui est possible de faire. Il faut penser à créer des interfaces faciles d’utilisation tout en

Le logiciel doit donc être :

* Ergonomique
* Intuitif
* Complet
* Evolutif
* Interopérable

Il a donc été décidé une organisation par onglets (un onglet « salles », un onglet « locaux » …).

Chaque onglet permettra l’ajout, la visualisation, ou la suppression des différents objets.

Un bouton permettra d’ajouter un objet dans la base avec ses différentes informations.

1. Phase de rédaction

Il y aura différents document rédigé comme ce rapport du projet Java et aussi la documentation utilisateurs. Ces documents sont importants pour mettre au point les besoins de la société STRI, expliquer les différentes phases du projet pour montrer la méthode employé, visualiser la répartition des tâches.

1. Répartition des tâches

Il y a une répartition des tâches effectuées par affinité. Mais toute personne à participé et suivis chaque tâche afin de se tenir au courant de l’avancement global du projet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Josselyn Loulier | Yohann Le Govic | Simon Lamoure | Thomas Maury |
| Découverte de Github | XXX | XXX |  | XXX |
| Diagramme d’UML | XX | XXX | XXX | X |
| Base de données | X | X | XXX | XX |
| Première interface | XXX | XX | X | XXX |
| Interface finale | XX | XXX | X | XX |
| Connexion de l’interface à la BDD |  |  | XXX | XXX |
| Codage des requêtes | X | X | XXX | XXX |
| Rédaction des documents | XXX | XXX | X | X |

**Légende :**

XXX : Forte participation

XX : Moyenne participation

X : Petite participation

1. Fin du projet

Comme nous sommes un groupe de 4 débutants en programmation, et encore plus en langages objets, nous n’avons pas pu aboutir le projet jusqu’à sa version finale et totalement fonctionnelle.

Pour entièrement répondre aux besoins, il manque les fonctionnalités suivantes :

* Modifier la base de données pour pouvoir gérer plusieurs versions d’un même système d’exploitation
  + Actuellement, un système d’exploitation correspond à une seule version.
* Gérer le type des appareils
  + Actuellement, aucune différence n’est faite sur les appareils, sauf sur la description.
* Prévoir la hiérarchie des incidents suite à la désactivation d’une interface
  + Actuellement, si une interface se voit désactivée, il n’y aura aucun impact sur le reste des appareils.
* Utilisation d’un projet Maven

Nous avions pensé à certaines évolutions du programme :

* Rendu graphique de l’état du réseau et des incidents.
* Accès au logiciel sécurisé par authentification