**Document de conception de l’application de gestion des interventions**

# Préambule

Le logiciel à développer permettra aux agents d’ENEO Cameroun de suivre les opérations de maintenance effectuées sur les installations du réseau de distribution d’ENEO Cameroun

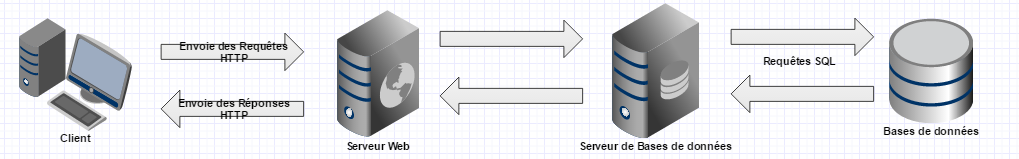
# Conception Générale

## Architecture 3-tiers

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

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

Dans l’architecture 3-tiers, les applications au niveau serveur sont délocalisées, c’est - à - dire que chaque serveur est destiné pour une tâche spécifiée, la figure (**Figure 1**) est un exemple explicatif :



**Figure 1 : Architecture 3-tiers**

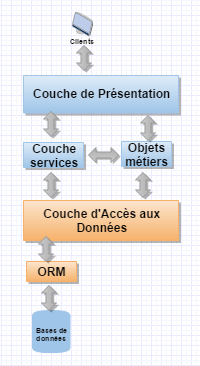
## Modèle en couche

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



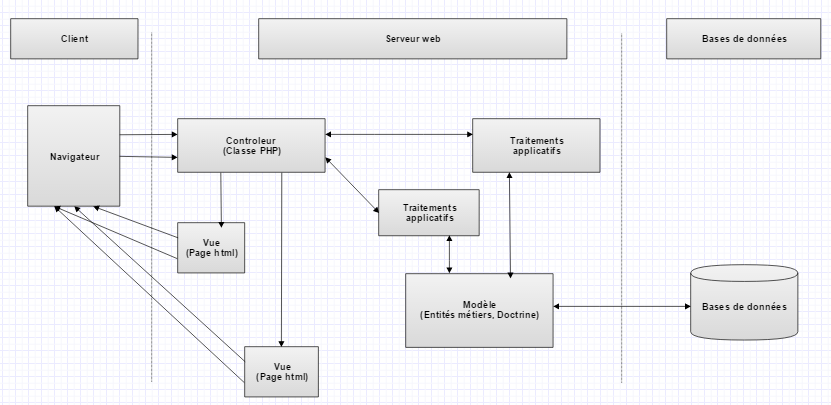
**Figure 2 : Architecture du modèle en couche.**

## Modèle MVC (Model, View, Controller)

Notre application repose sur le modèle en couche. En effet, pour rester dans le principe de l'autonomie, il faut que les couches soient indépendantes. Donc pour obtenir une telle architecture il est indispensable d’utiliser des modèles de conception qui facilite cette tâche fastidieuse.

Le modèle de conception que nous allons choisir et qui satisfait notre besoin est le MVC.

La figure suivante (**Figure 3**) illustre le principe du modèle MVC



**Figure 3 : Modèle MVC**

# Conception détaillée

## Couche d’accès aux donnée

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

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

## Couche métier

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

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

Couche de présentation

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

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

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

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

## Conception des Cas d’utilisation

### **Diagramme de cas d’utilisation**

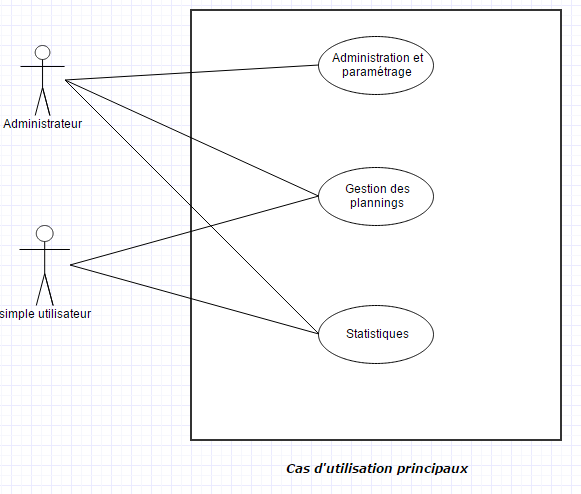
Nous pouvons distinguer ici trois grands domaines de l’application en termes d’acteurs et des moyens mis en œuvre.

* L’administration et le paramétrage : L’administration est effectuée à chaque niveau par le responsable de ce niveau (Directeur) ou un autre personne tierce désignée à cet effet via une interface web. Cet administrateur devra créer les utilisateurs de l’application et éventuellement pouvoir attribuer les droits à ces derniers. Un administrateur d’un niveau pourra créer un niveau inférieur à son niveau d’appartenance avec des droits requis. Il devra aussi créer les semaines de travail dans un mois qui seront utilisées dans la gestion des plannings d’intervention
* La gestion des plannings d’intervention : Ce domaine est effectué par un utilisateur qui en a les droits. Il regroupe en son sein la gestion des opérations, des sous opérations et des interventions d’un planning
* Les statistiques : Les statistiques pourront être visualisées par tous les utilisateurs en lecture seulement. Elles porteront principalement sur le taux de réalisation et le taux des opérations non planifiées mais réalisées, elles pourront concerner une base technique ou une région par période (semaine, mois, années).

Les rapports pourront être générés à la fin d’une semaine pour un planning ou à la fin d’un mois pour plusieurs plannings.

Les acteurs principaux sont :

* Les administrateurs
* Les simples utilisateurs

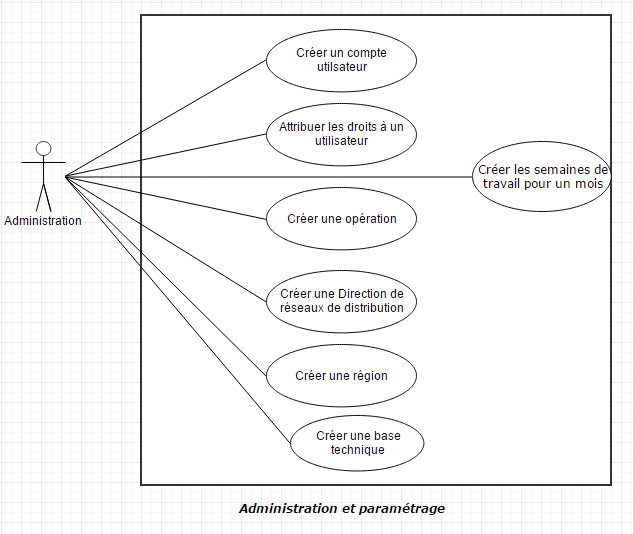


1. Description des acteurs

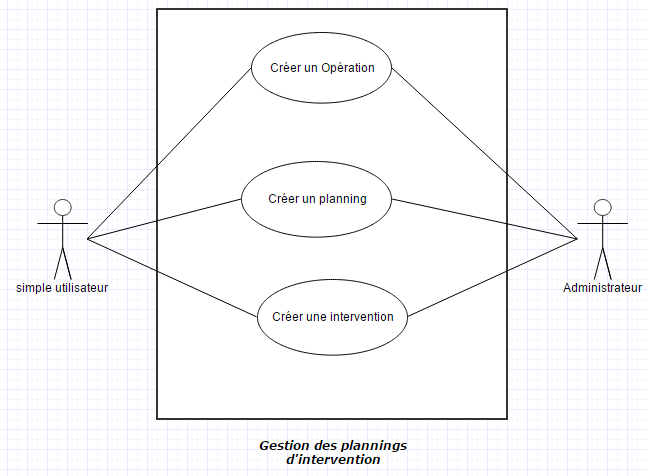
|  |  |
| --- | --- |
| Acteur | Rôle |
| Administrateur | Responsable de l’administration et du paramétrage. Il créé les opérations nécessaire à la création des plannings.  Visualiser les statistiques |
| Simple utilisateur | Responsable de la création des plannings et des interventions qui sont associées à sa base technique |

1. Description des cas d’utilisation

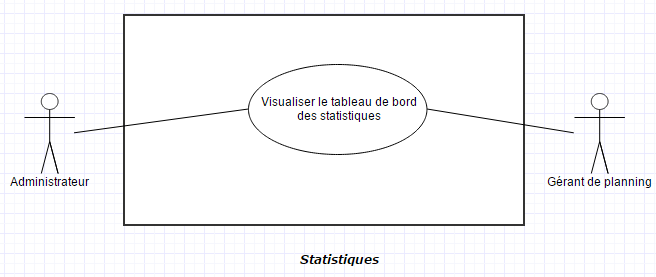
* Description des Cas d’utilisation par des diagrammes
* Le cas d’utilisation : Administration et paramétrage



* Le cas d’utilisation : Gestion des plannings



* Le cas d’utilisation : les statistiques



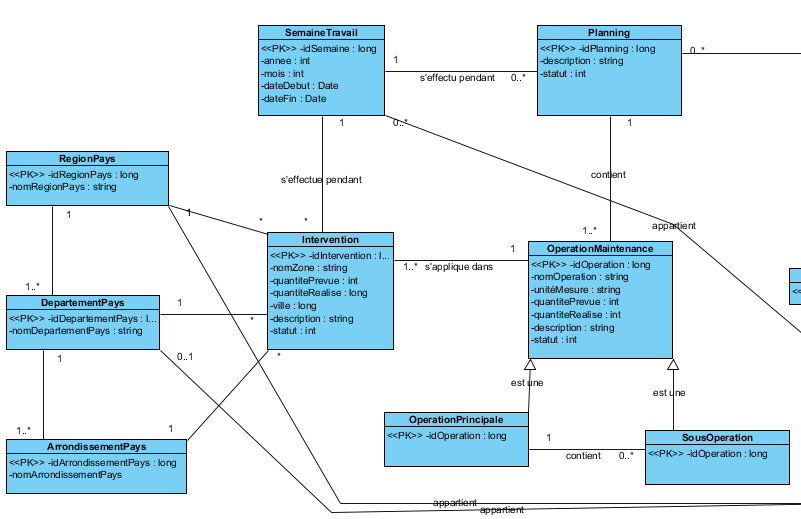
|  |  |
| --- | --- |
| Scénario | Description |
| Créer un compte utilisateur | Création d’un compte pour un utilisateur du système par un administrateur. |
| Attribuer les droits à un utilisateur | Un utilisateur créer par un administrateur aura besoin des droits pour interagir avec le système |
| Créer une opération | L’administrateur dans sa tâche d’administration doit créer les opérations dans le système qui seront utilisés par les gérants de planning. |
| Créer une semaine de travail | Puisqu’ un planning d’interventions est exécuté pendant une semaine, elle devra déjà être connue d’avance donc existée dans le système. Elle nous donne le début et la fin de l’exécution d’un planning dans un mois précis d’une année. |
| Créer une Direction des réseaux de distribution | Une Direction des réseaux de distribution doit être créée par un administrateur pour pouvoir créer une région qui appartient toujours à une Direction |
| Créer une région | Une région doit exister pour nous permettre de créer une base technique |
| Créer une base technique | Création d’une base technique liée une région précise |
| Créer un planning | La liste des taches que l’on va effectuer sur le terrain au minimum pendant une semaine |
| Créer une intervention | Une intervention correspond à l’exécution d’une des opérations existant dans le système sur le terrain. |
| Visualiser le tableau de bord des statistiques | C’est le tableau de bord de contrôle des activités dans les régions et les base techniques |

## Conception des classes métiers

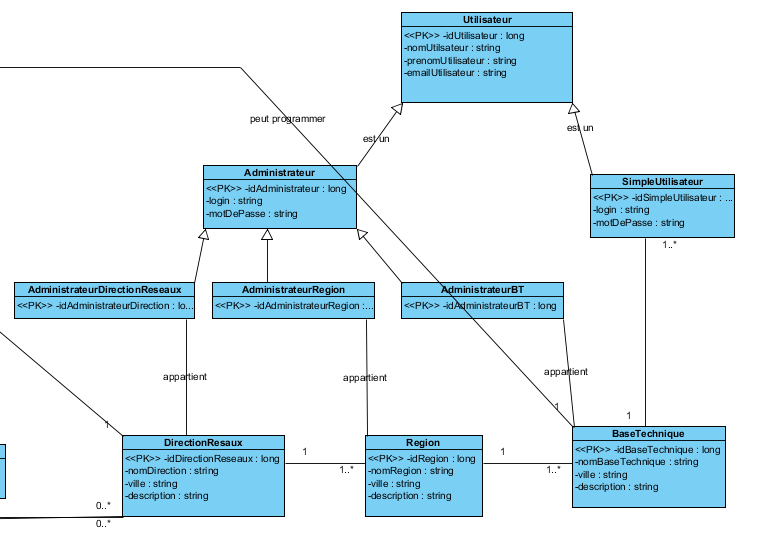
Un diagramme de classe constitue l’un des pivots essentiels dans la modélisation d’un logiciel informatique. Il permet de définir quelles seront les composantes du système final en mettant en  
évidence les différentes classes d’un système et en modélisant les relations qui les associent, les interactions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

Voici donc le diagramme de classe de notre future application :

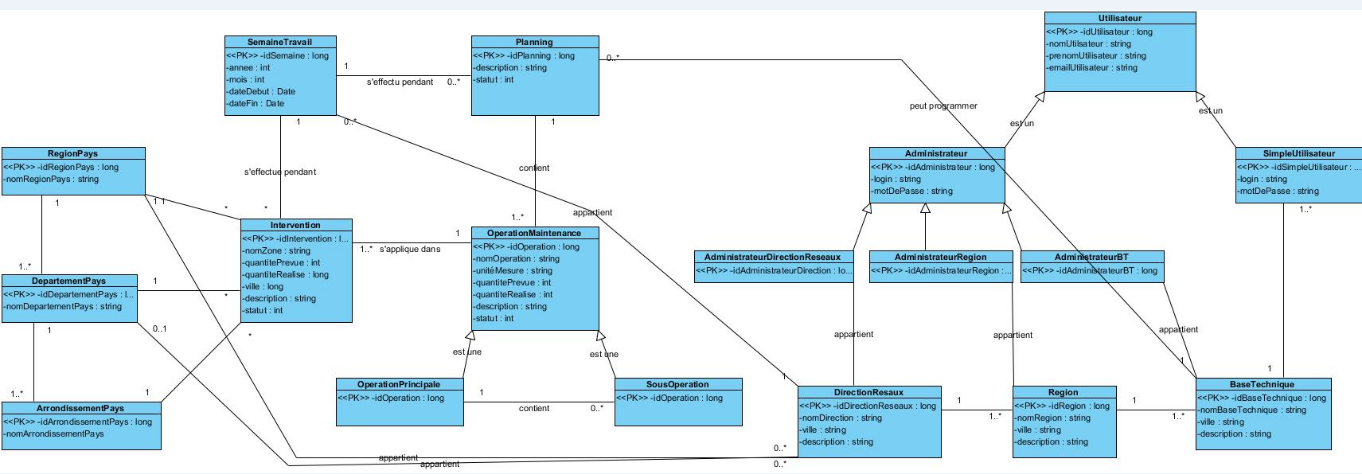
Le diagramme de classe concernant la tache de gestion les plannings des interventions est le suivant :



Le diagramme de classe concernant les cas d’utilisation de l’administration est le suivant :



Voici donc le diagramme de classe global du système :



## Interface Homme machine

Les interfaces conçues ne sont que les fruits d'une longue chaîne de travail ; ce n'est qu'une simple image dont l'utilisateur ignore les arrières plans.  
La page d’authentification : c’est la première page vue par l’utilisateur. L'authentification sert à restreindre l'accès à l'application et à délimiter ce mode d'accès selon les privilèges accordés à l'utilisateur. L'interface ci-dessous présente notre page d'authentification. Il s'agit de saisir l'identifiant et le mot de passe pour pouvoir accéder aux fonctionnalités accordées selon le droit d’accès.

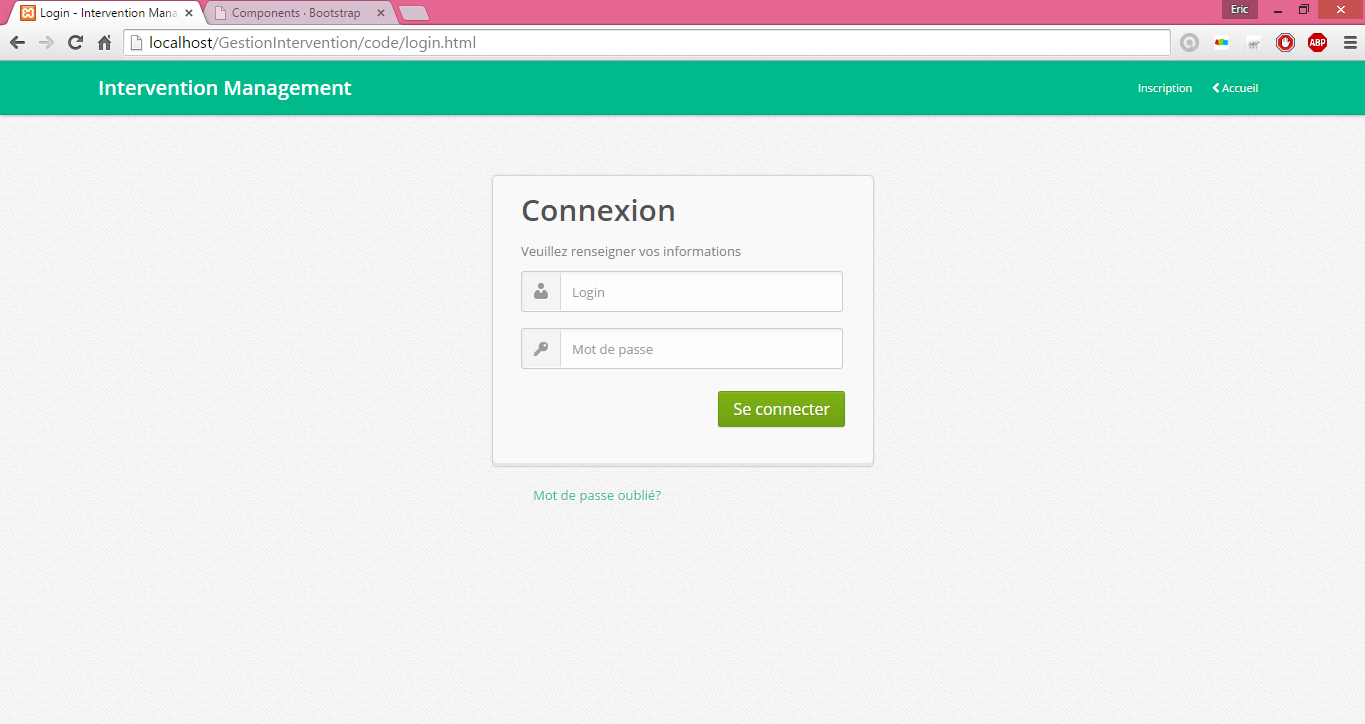
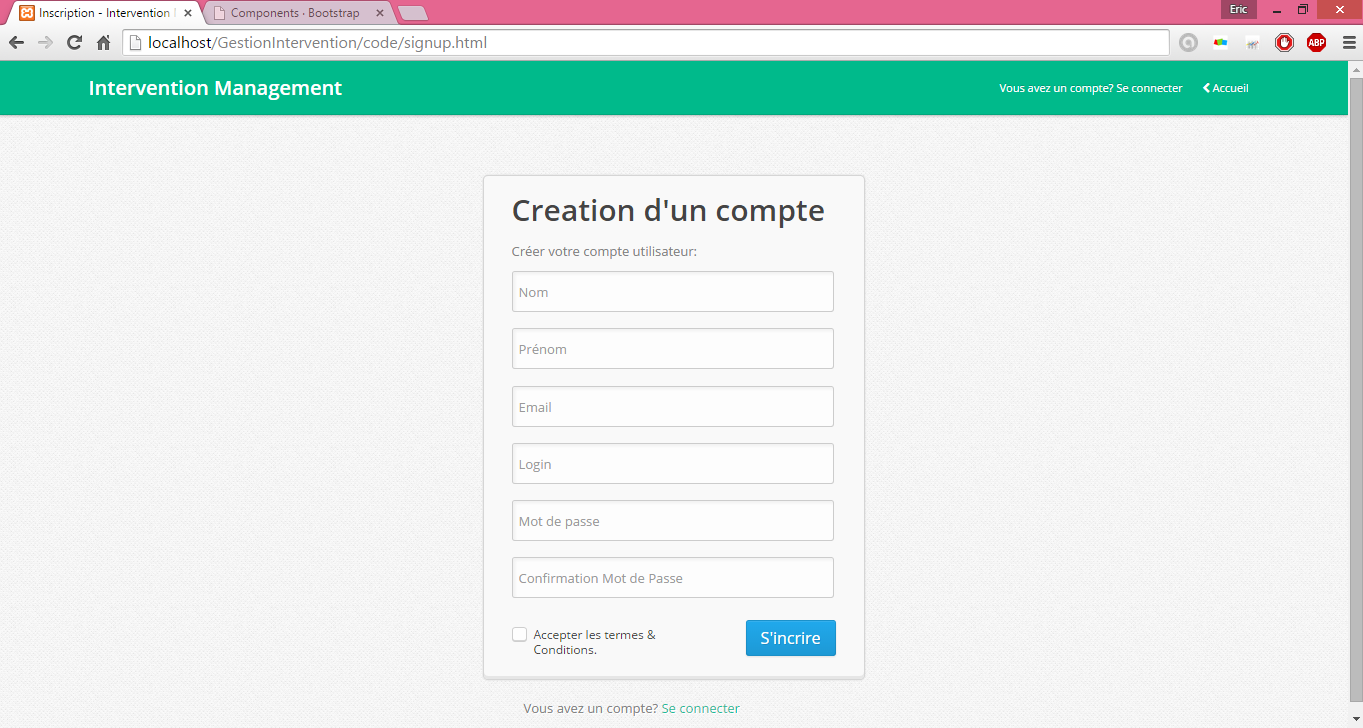
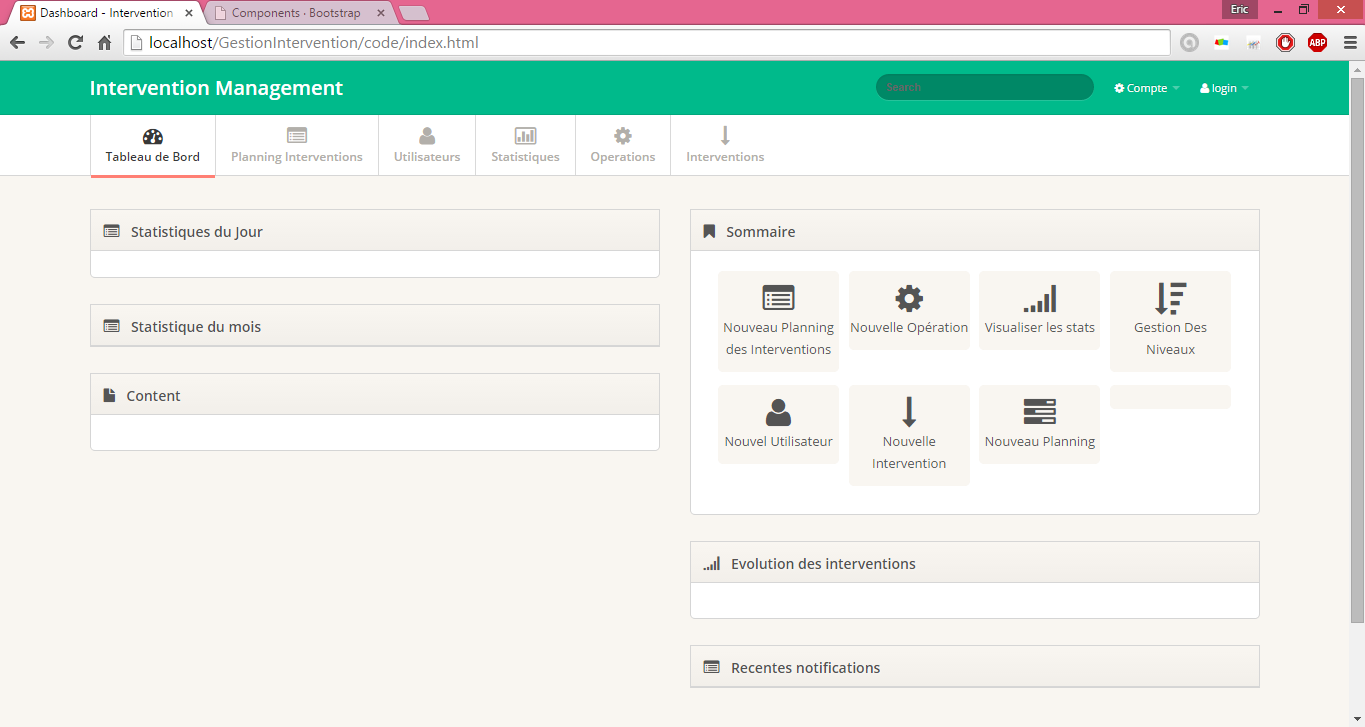


Figure : Interface d’authentification

Si on ne dispose pas encore d’un compte utilisateur nous pouvons le créer en cliquant sur inscription sur la page d’authentification. On nous propose donc cette interface d’inscription :



Apres donc la connexion d’un utilisateur, il arrive à la page d’accueil qui est le tableau de bord :



A partir d’ici nous pouvons donc accéder à toute les fonctionnalités du logiciel.