

# Projet Cimetière

---

Rapport de stage :

Philippe Basuyau

Octobre 2016

Hello everyone,

Thanks to be there,

My name is Philippe and following a skills assessments. I'm currently in retraining, so my choice was focused on IT as you noticed.

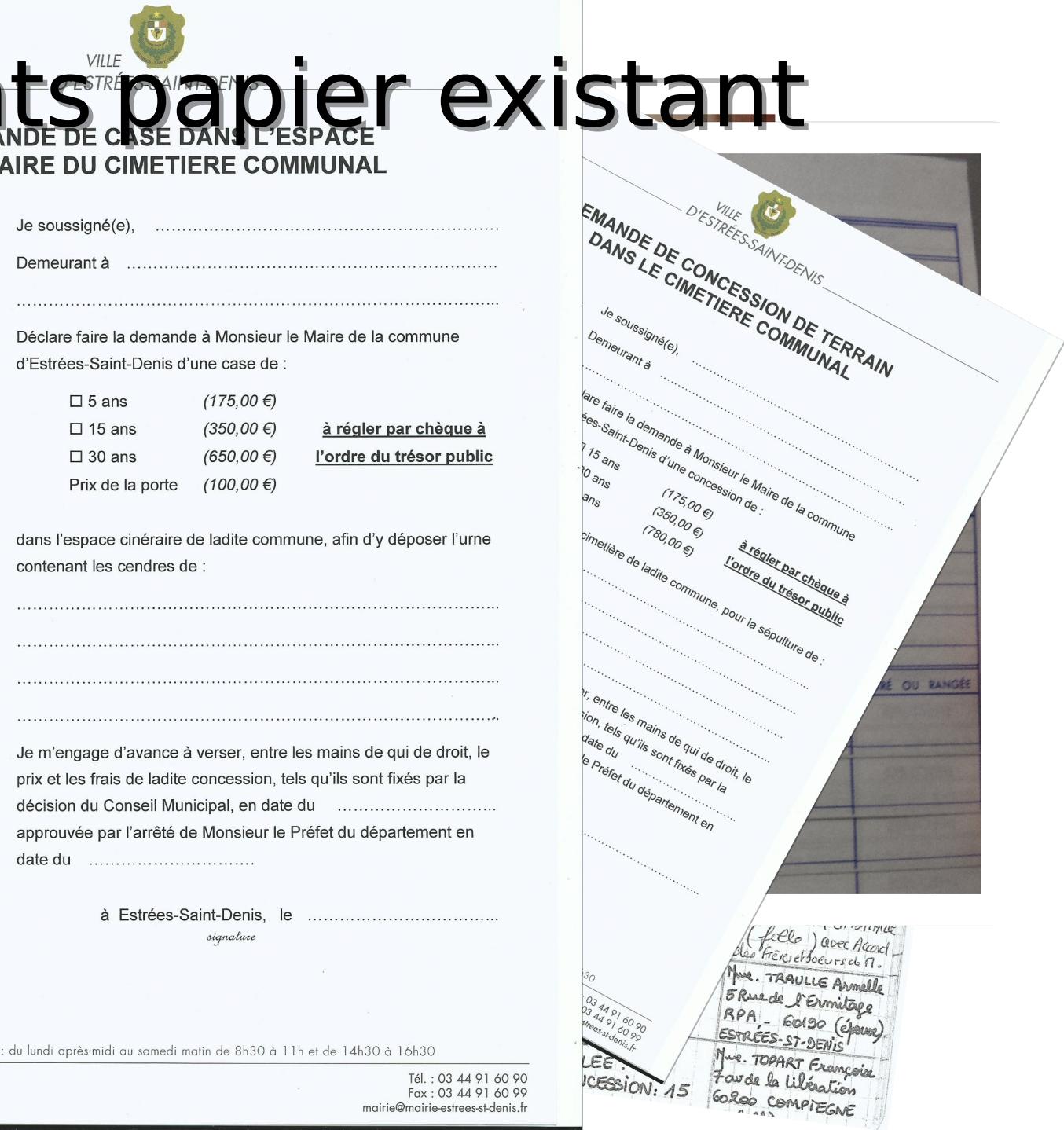
I'm here today for my software developer degree and I'll introduce my training project, the e-cemetery which I conceived, shaped and developed alone.

For that, I had to learn the terms of job and read the legislation governing this subject. Next, I approached territorial agents and I asked them many questions to better understand and apprehend their work habits.

I conceived the specifications with them according to their needs. In this adventure, I acted as the « ScrumMaster », as the "designer" for views and as the "developer" of course for codes.

After a presentation at the town hall, they already requested me future improvements.

# Les Documents papier existant





DONE



DOING



TODO

## La Gestion du Cimetière

Mettre en place le rappel automatique des concessions échues  
Mieux pensé pour et par l'utilisateur

Plus souple et plus intuitif



Avantages et Gains apporter par le Logiciel

Facilité et simplicité de travail

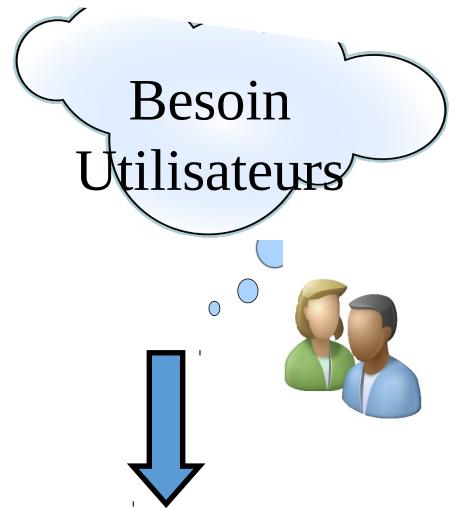
Plus sécurisé (gestion des Agent Territoriaux)

Choix du concessionnaire sur support mobile

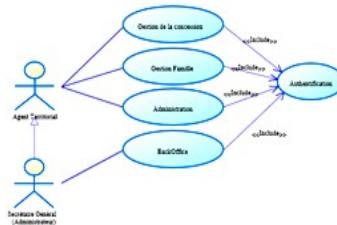
Plus pratique pour les procédures de reprise

Gain de temps, l'équivalent à un mi-temps

# Le modèle de Pascal Roques

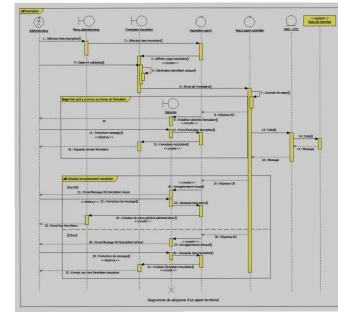


## Chapitre 3



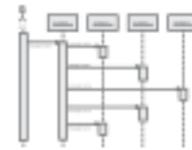
Diagrams  
De cas d'utilisation

## Chapitre 4



Diagrams de  
séquence système

## Chapitre 7



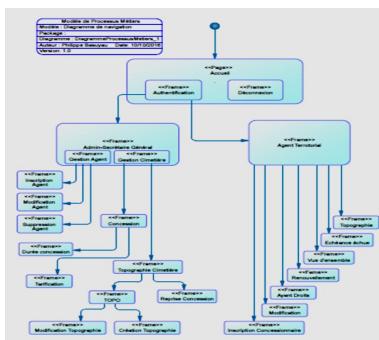
Diagrams d'interaction

## Chapitre 5



Diagrams de  
classes  
participantes

## Chapitre 6



Diagrams  
De navigation



Diagrams de classes  
de conception

## Code

```
void base64_encode(const uint8_t *data, size_t len, char *dst)
{
    size_t src_idx = 0;
    size_t dst_idx = 0;
    for ( ; src_idx < len; src_idx += 3, dst_idx += 4)
    {
        uint8_t s0 = data[src_idx];
        uint8_t s1 = data[src_idx + 1];
        uint8_t s2 = data[src_idx + 2];
        uint8_t s3 = (char) ((s1 & 0xfc) >> 2);
        dst_idx += 0 = charset((s0 & 0xfc) >> 2);
        dst_idx += 1 = charset(((s0 & 0xfc) >> 2) | (s2 & 0x30) >> 6);
        dst_idx += 2 = charset(((s0 & 0xfc) >> 2) | (s2 & 0x30) >> 6);
        dst_idx += 3 = charset((s2 & 0x30));
    }
    if (src_idx < len)
    {
        uint8_t s0 = data[src_idx];
        uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
        dst_idx += 0 = charset(((s0 & 0xfc) >> 2));
        dst_idx += 1 = charset(((s0 & 0xfc) >> 2) | (s1 & 0x30) >> 6);
        dst_idx += 2 = charset(((s0 & 0xfc) >> 2) | (s1 & 0x30) >> 6);
    }
}
```

Figure empruntée à « UML 2, Modéliser une application Web », Eyrolles.

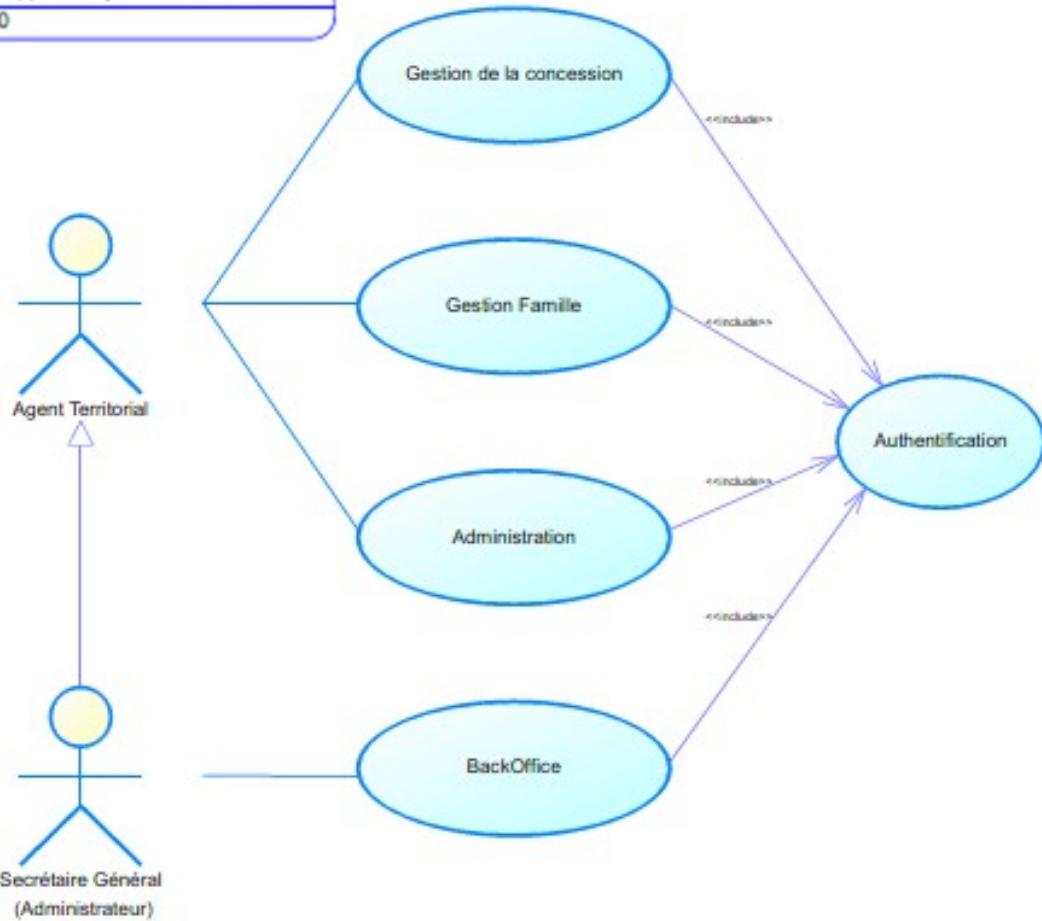
DONE :  
Besoins

DOING :  
DCU

TODO :  
DAC

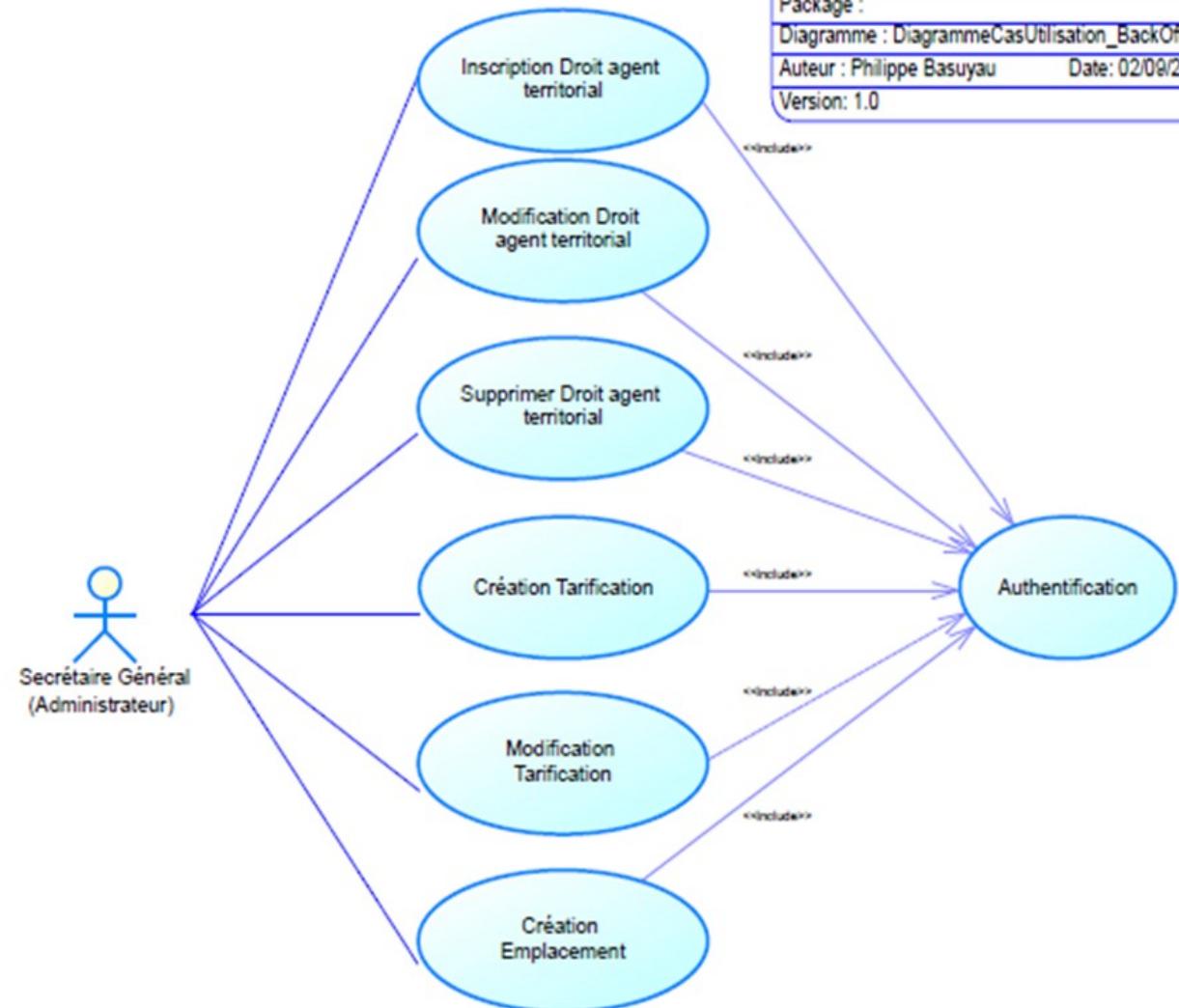
# Diagramme de Cas Utilisation

Modèle orienté objet
Modèle : Projet Cimetière Général
Package :
Diagramme : DiagrammeCasUtilisation
Auteur : Philippe Basuyau Date: 02/09/2016
Version: 1.0



Cas d'utilisation de l'agent territorial

Modèle orienté objet
Modèle : Définition de cas d'utilisation_BackOffice
Package :
Diagramme : DiagrammeCasUtilisation_BackOffice
Auteur : Philippe Basuyau Date: 02/09/2016
Version: 1.0



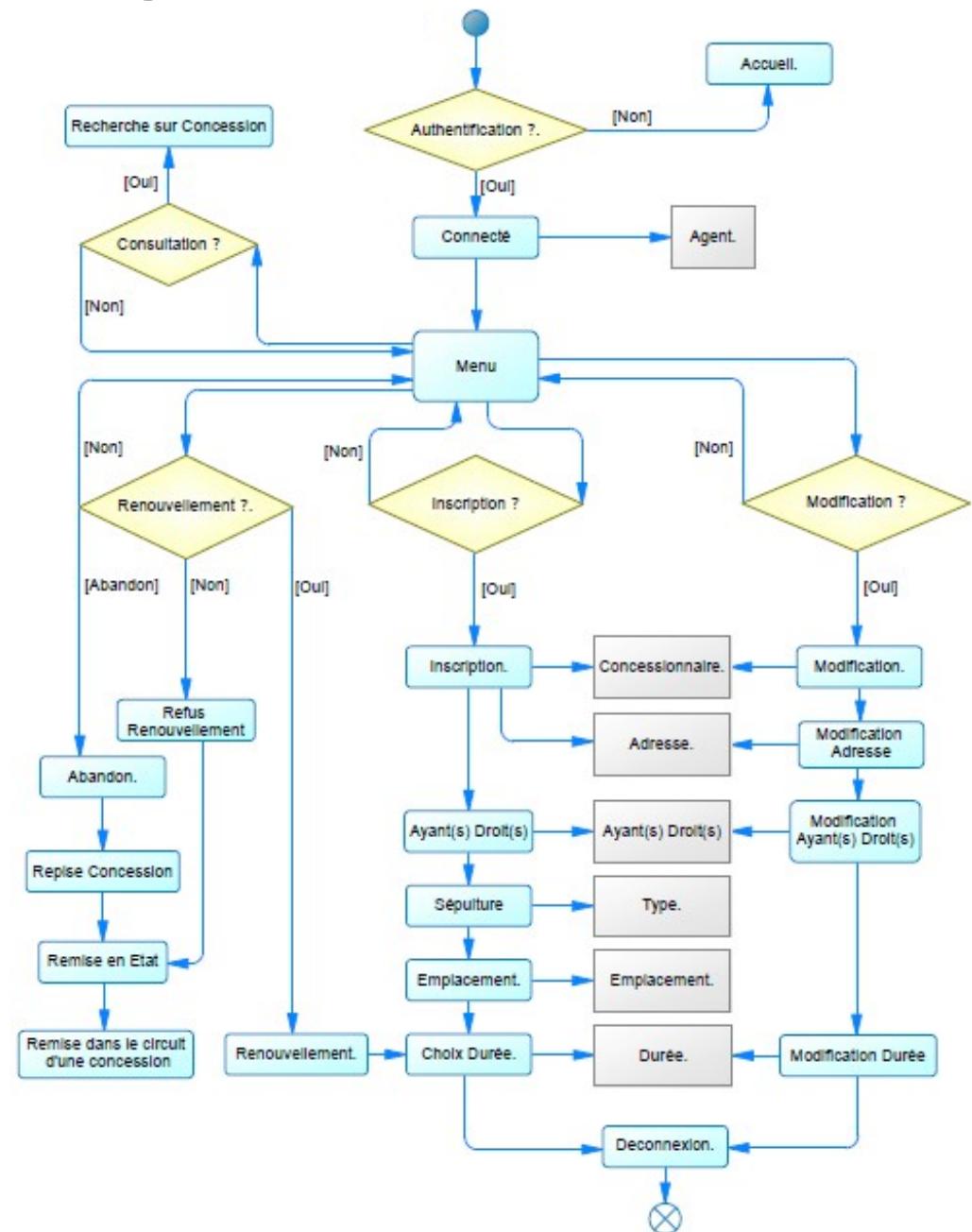
Cas d'utilisation du secrétaire général

**DONE :**  
**DCU**

DODIN  
G:  
DCA

TODO

# Diagramme d'Activités : général agent territorial



## Diagramme d'Activités général du coté de l'agent territorial

## Choix

## Action

# Data

## Concession

## Modification

## Ayant droits

Renouvellement

## Vue d'ensemble

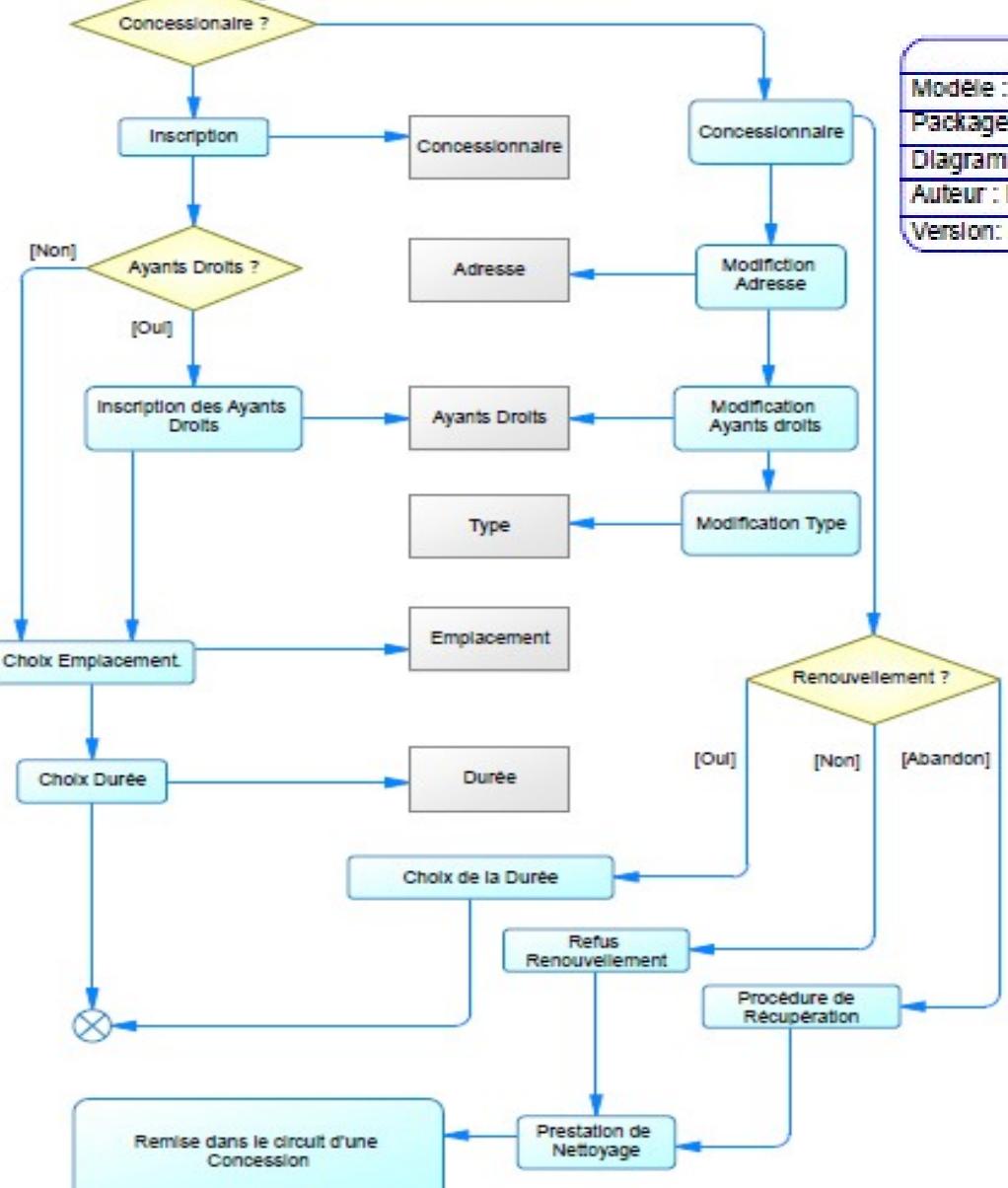
## Échéance

# Topographie

Digitized by srujanika@gmail.com

Modèle orienté objet
Modèle : Diagramme d'activités UML
Package :
Diagramme : DiagrammeActivites_AgentTerritorial
Auteur : Philippe Basuyau Date: 26/08/2016
Version: 1.0

# Diagramme d'Activités : Le Concessionnaire



Modèle orienté objet  
Modèle : Diagramme d'activités UML  
Package :  
Diagramme : DiagrammeActivites\_Inscription  
Auteur : Philippe Basuyau Date: 26/08/2016  
Version: 1.0



## Le Diagramme d'activités : Le concessionnaire

Ce diagramme représente l'activité principale de l'agent territorial.

L'inscription

Modification

Les ayants Droits

Renouvellement

**DONE :**  
DAC

**DOING :**  
Les Maquettes

**TODO :**  
DNAV

Nous sommes le 29 août 2016

# E-Cimetière

Deconnexion

Bonjour, Mme Nat\*\*\*\* T\*\*\*\*\*

Inscription

Ayant Droit

Consultation

Renouvellement

## Inscription du Concessionnaire

### Concessionnaire

Civilité	Mr ▼
Nom	Ici le nom
Nom marital	Ici le nom marital
Prénom	Ici le prénom
Date de Naissance	Ici la date de naissance
Adresse	Ici l'adresse
Ville	Ici le nom de la ville
Code Postal	Ici le code postal
Téléphone	Ici le Numéro de téléphone
Email	Ici l'email

### Choix de la Sépulture

Sépulture	Choix Type Sépulture ▼
Emplacement	Section ▼ Allée ▼ N° ▼
Durée	Choix de la durée ▼
Type concession	Type de concession (familiale) ▼
Date du jour	Ici la date du jour

### Tarification

Tarif Sépulture Tarif de la sépulture

### Conjoint(e)

Civilité	Mr ▼
Nom	Ici le nom
Nom marital	Ici le nom marital
Prénom	Ici le prénom
Date de Naissance	Ici la date de naissance
Adresse	Ici l'adresse
Ville	Ici le nom de la ville
Code Postal	Ici le code postal
Téléphone	Ici le Numéro de téléphone
Email	Ici l'email

Valider

**DONE :**  
DAC

**DOING :**  
Les Maquettes

**TODO :**  
DNAV

Nous sommes le 29 aôut 2016

# E-Cimetière

[Deconnexion](#)

Bonjour, Mr Muss\*\*\*

[Accès Agent](#)

[Inscription Agent Territorial](#) ▼

## Inscription de droit d'Agent Territorial

Mr ▼

Nom

Mot de passe

Prénom

Confirmer le mot de passe

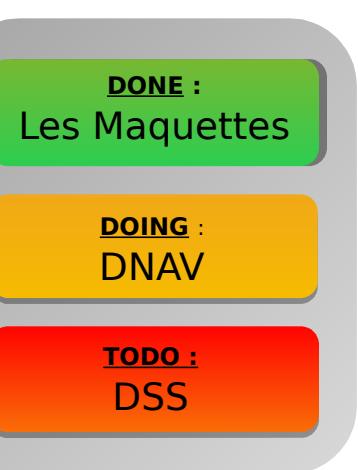
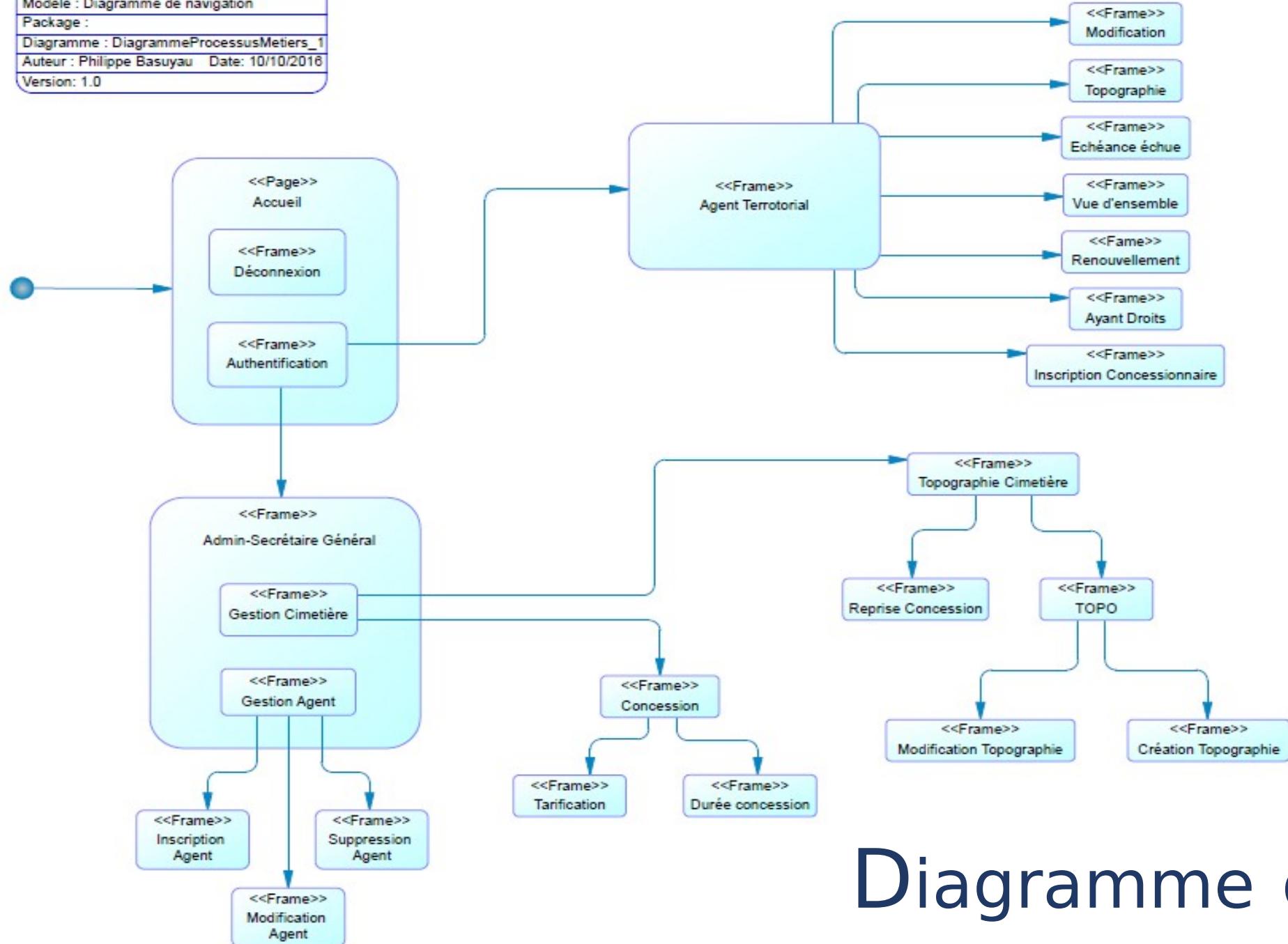
Email

Fonction de l'agent  ▼

Autorisation de droit  Cocher la case pour l'Autoriser l'accès

**Valider**

Modèle de Processus Métiers
Modèle : Diagramme de navigation
Package :
Diagramme : DiagrammeProcessusMetiers_1
Auteur : Philippe Basuyau Date: 10/10/2016
Version: 1.0



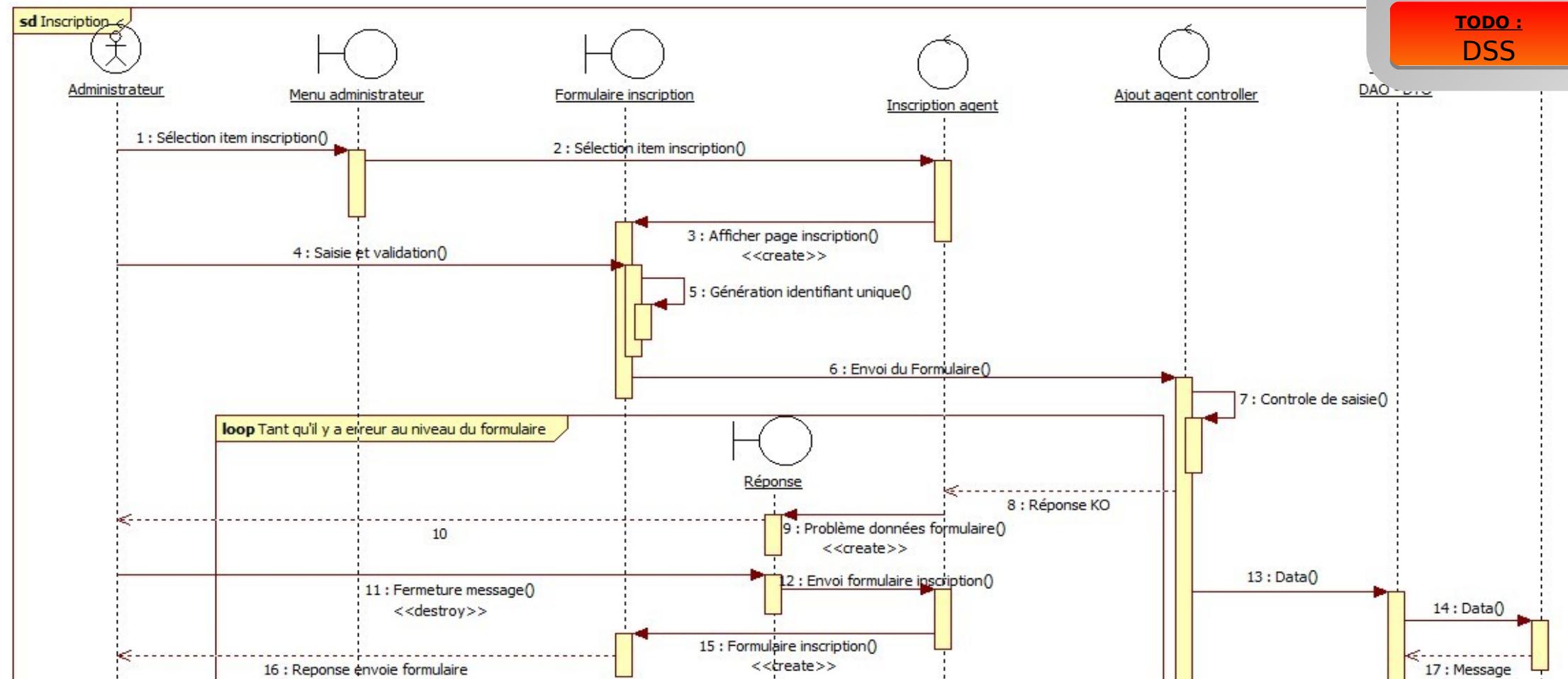
# Diagramme de Navigation



Diagram  
me  
d  
e  
Séquen  
ce  
Systè  
me

# Scénario

C

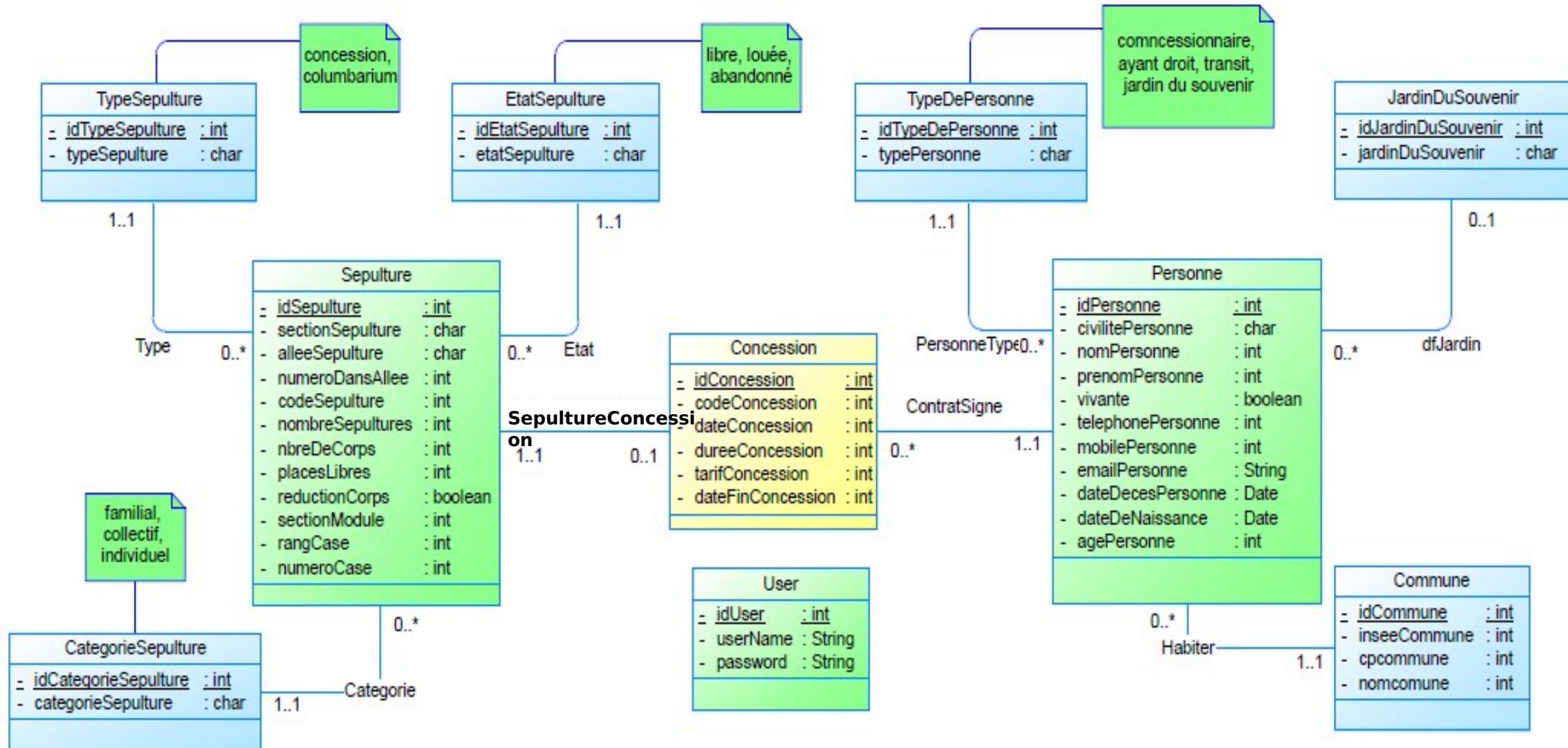


DONE :  
DSD

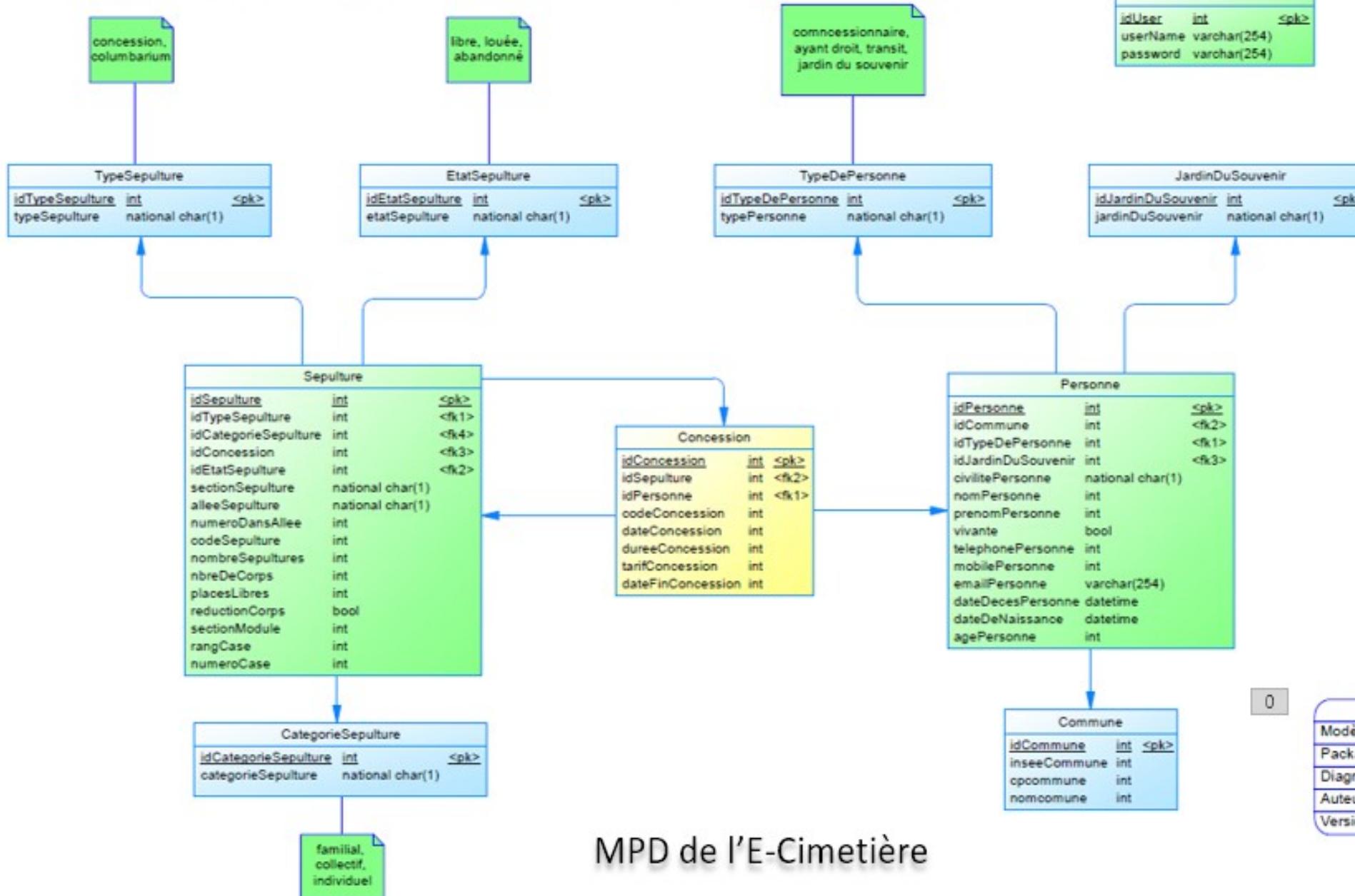
DOING :  
DCL

TODO :  
MPD

# Le Diagramme de Classes



# Modèle Physique de Données



DONE :  
DCL

DOING :  
MPD

TODO :  
MySQL

Modèle Physique de Données		
Modèle : mdpcimetiere		
Package :		
Diagramme : DiagrammeClassesCimetiere		
Auteur : Philippe Basuyau Date: 05/09/2016		
Version: 2.0		

MPD de l'E-Cimetière

# MySQL : La BD Cimetière

DONE :  
MPD

DOING :  
MySQL

TODO :  
MVC

```
/*=====
/* Nom de SGBD : MySQL 5.0
/* Date de création : 01/09/2016 12:04:28
/*=====
```

```
DROP DATABASE cimetiere;
CREATE DATABASE IF NOT EXISTS cimetiere
DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci;
```

```
USE cimetiere;
```

```
SET FOREIGN_KEY_CHECKS = 0;
```

- J'efface la BD Cimetière si elle existait  
n'existe pas

- J'utilise DEFAULT CHARACTER SET pour le tri j'utilise  
utf\_general\_ci

J'utilise la BD  
cimetière

Je désactive les clés  
étrangères

```
/*=====
/* Table : Commune
/*=====

create table Commune
(
    idCommune          int not null auto_increment,
    inseeCommune       char(5),
    cpCommune          char(5),
    nomCommune         varchar(50),
    primary key (idCommune)
)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

## Table sans FK

Char : 5 caractères  
varchar : 50 caractères ne prend pas en compte les vides donc gain de place

ENGINE=InnoDB : il gère les FK et les transaction plus l'indexation

# MySQL : La BD Cimetière

```
/*
 * Table : Concession
 */
create table Concession
(
    idConcession      int  not null auto_increment,
    idSepulture       int  not null,
    idPersonne        int  not null,
    codeConcession    int(5),
    dateConcession   date,
    dureeConcession  int(3),
    tarifConcession  int(5),
    dateFinConcession date,
    primary key (idConcession)
)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```



## Table avec FK

Pk : chiffre entier non nul et auto incrémenté

FK : chiffre entier et non nul

Utilisation: de l'utf8,  
COLLATE  
utf8\_unicode\_ci

```
alter table Sepulture add constraint FK_Etat foreign key (idEtatSepulture)
    references EtatSepulture (idEtatSepulture) on delete restrict on update restrict;
```

Prend la clé primaire (idEtatSepulture) de la table Etat et met la comme contrainte sur la table Sepulture comme clé étrangère, et je ne peux pas effacer cette clé primaire du parent ou la modifier tant qu'elle sera en clé étrangère dans l'enfant

```
SET FOREIGN_KEY_CHECKS = 1;
```

DONE :  
MPD

DOING :  
MySQL

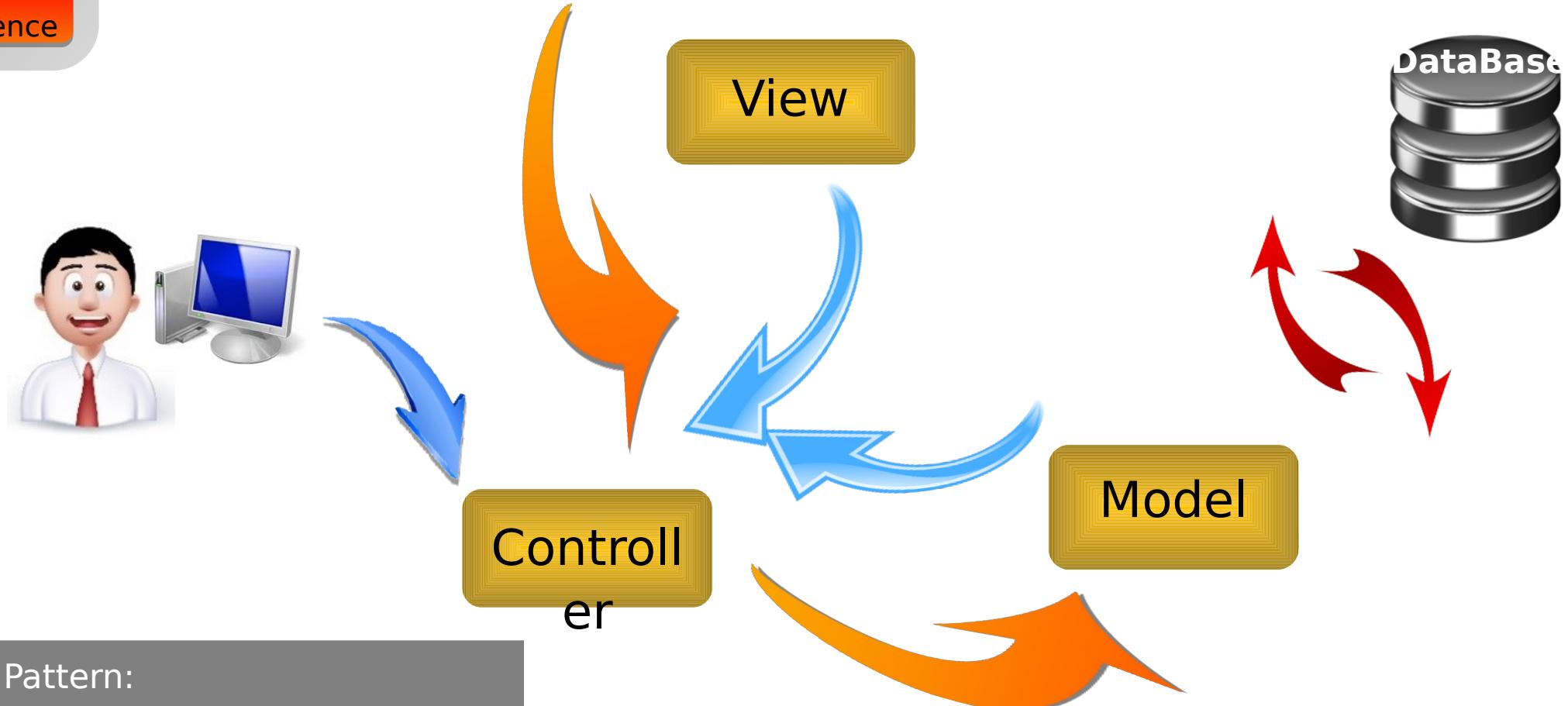
TODO :  
MVC

**DONE :**  
MySQL

**DOING :**  
MVC

**TODO :**  
Arborescence

# Modèle MVC



Le Pattern:

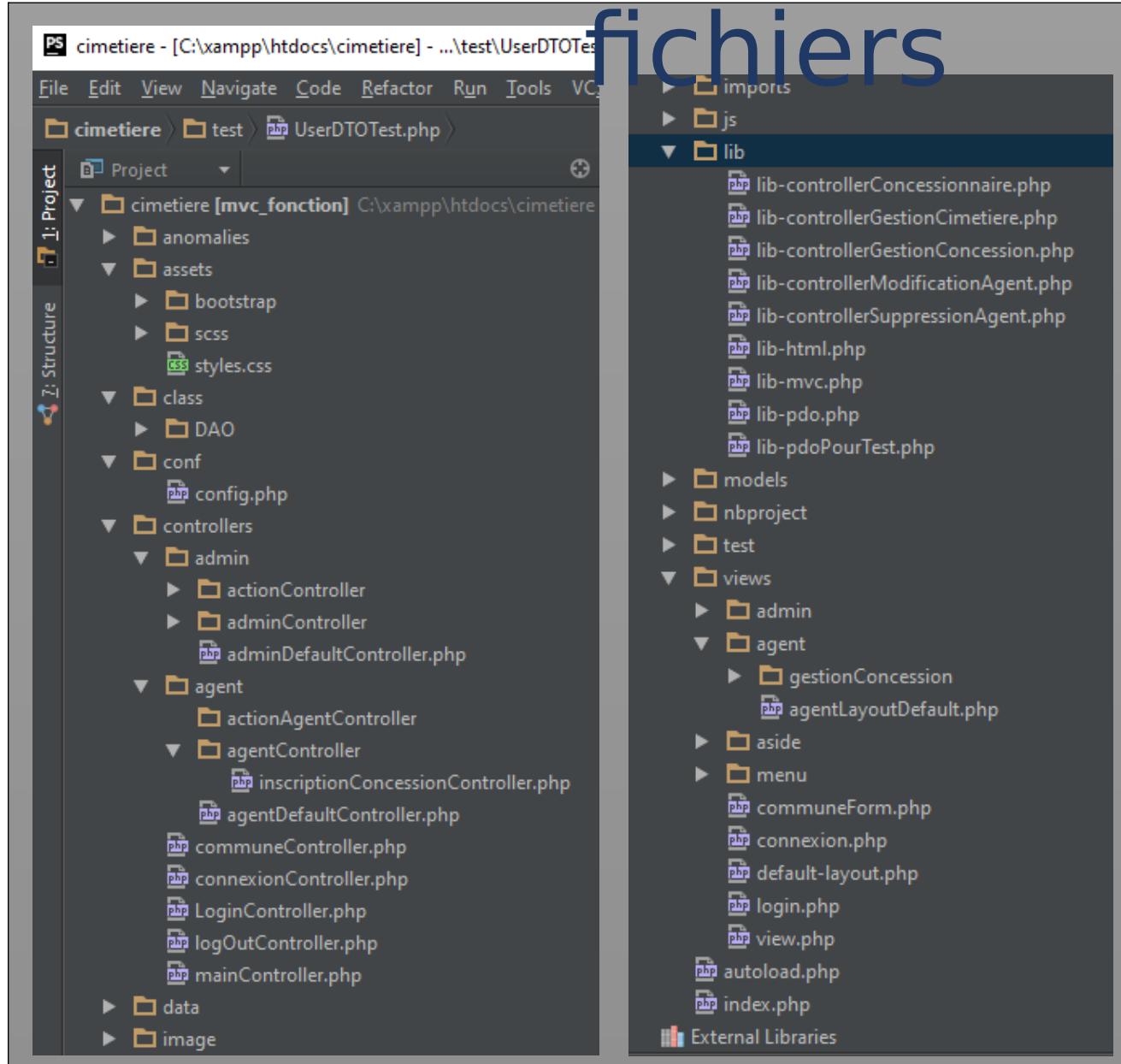
**MVC (Model - View - Controller)**

**DONE :**  
MVC

**DOING :**  
Arborescence

**TODO :**  
Sécurité

# PHPStrom l'Arborescence des fichiers



# La Sécurité

DONE :  
MVC

DOING :  
Sécurité

TODO :  
DAO - DTO

```
<?php index
```

// A la place require je sécurise contre les doublons de code avec require\_once

```
require_once 'conf/config.php';
```

```
require_once 'lib/lib-html.php';
```

```
require_once 'lib/lib-mvc.php';
```

```
require_once 'lib/lib-pdo.php';
```

//Récupération des données

```
require_once 'models/user.php';
```

```
require_once 'autoload.php';
```

//pour éviter les injections XSS je place "FILTER\_SANITIZE\_STRING" sur chaque post récupéré

Dans le contrôleur qui récupère les informations du formulaire

```
$idUser = filter_input(INPUT_POST, 'idUser', FILTER_VALIDATE_INT);  
$identifiant = filter_input(INPUT_POST, 'identifiant',  
    FILTER_SANITIZE_STRING);  
$userName = filter_input(INPUT_POST, 'userName',  
    FILTER_SANITIZE_STRING);  
$userNameConfirm = filter_input(INPUT_POST, 'userNameConfirm',  
    FILTER_SANITIZE_STRING);  
$password = filter_input(INPUT_POST, 'password',  
    FILTER_SANITIZE_STRING);
```

# La Sécurité

//Validation des données

```
if ($userName != $userNameConfirm) {  
    $message = "Le nom n'est pas conforme à sa confirmation";  
    $_SESSION['flash'] = $message;  
    header('location:index.php?controller=inscriptionController');  
} elseif ($password == null || mb_strlen($password) < 6) {  
    $message = "Votre mot de passe doit comporter au moins 6 caractères";  
    $_SESSION['flash'] = $message;  
    header('location:index.php?controller=inscriptionController');  
} elseif ($password != $passwordConfirm) {  
    $message = "Le mot de passe n'est pas conforme à sa confirmation";  
    $_SESSION['flash'] = $message;  
    header('location:index.php?controller=inscriptionController');  
} elseif ($identifiant == "") {  
    $message = "L'identifiant est vide, veuillez remplir le champs";  
    $_SESSION['flash'] = $message;  
    header('location:index.php?controller=inscriptionController');  
} elseif (isset($submitAjoutInscription) && isset($identifiant) && isset($userName)  
&& isset($password)) {  
    ...  
}
```

Le contrôleur

Sur la

Concordance

Des champs

**DONE :**  
MVC

**DOING :**  
Sécurité

**TODO :**  
DAO - DTO

# Le générateur d'identifiant - JavaScript

## L'Inscription d'un agent territorial

Générer un Identifiant

Votre identifiant généré:

8wDa5e

```
<HEAD>
<!-- //le JavaScript dans le head-->
<SCRIPT LANGUAGE="JavaScript">
```

```
function getIdentifier() {
    // Définir tous les caractères possibles dans le mot de passe
    var cars =
        "CDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz";
    var long = cars.length;
    wpas = "";
    taille = 6;//je veux un identifiant de 6 caractères

    // je vais utiliser la fonction de Math.random pour prendre au
    // hasard
    for (i = 0; i < taille; i++) {
        wpos = Math.round(Math.random() * long);
        wpas += cars.substring(wpos, wpos + 1);
    } // fin du for
    return wpas; // affiche moi le résultat créé
} //fin de la fonction getIdentifier
```

```
</script>
</HEAD>
```

**DONE :**  
MVC

**DOING :**  
Sécurité

**TODO :**  
DAO - DTO

# Transfert de données

```
<?php
class UserDAO implements IUserDAO {
    /**
     * @var \PDO
     */
    private $pdo;

    /**
     * DAOClient constructor.
     * @param PDO $pdo
     */
    public function __construct(PDO $pdo) {
        $this->pdo = $pdo;
    }

    public function findAll() {
        // La requete SQL
        $sql = "SELECT * FROM user";
        $rs = $this->pdo->query($sql)
            ->fetchAll();
        return $rs;
    }

    public function findOneById(array $idUser) {
        // requete SQL
        $sql = "SELECT * FROM user WHERE idUser=? ";
        // je stocke ma requete SQL
        $statement = $this->pdo->prepare($sql);
        // je l'execute
        $statement->execute($pk);
        // fletch est associé à l'objet PDOStatement
        // il sert à récupérer une ligne depuis un jeu de résultats
        // je mets le résultat dans la variables rs
        $rs = $statement->fetch();
    }
}
```

```
1 <?php
2
3 interface IUserDAO {
4
5     public function findAll();
6
7     public function findOrDie($id);
8
9     public function find($id);
10
11    public function delete($id);
12
13    public function save();
14
15 }
```

```
// implode : rassemble les éléments du tableau en une chaîne de caractères
$sql .= implode(" AND ", $array);
}

$statement = $this->pdo->prepare($sql);
$statement->execute($search);
```

```
private function insert(UserDTO $user) {  
    $sql = "INSERT INTO user (identifiant, userName, p  
VALUES (?, ?, ?, ?)"
```

Les Libs - config

```
$user) {  
    t=?, userName=?, password=?,  
  
    etUserName(), $user->getPassword(),
```

(\$sql);  
le **CUD**  
**(Create Update**  
**Delete)**

```
$statement = $this->pdo->prepare($sql);
return $statement->execute([$user->getIdUser()]);
else{
    return "Problème d'ajout de l'utilisateur";
}
```

# Le tout forme le CRUD

# Transfert des Données

```
<?php  
class UserDTO{  
  
    private $idUser;  
    private  
    $identifiant;  
    private  
    $userName;  
    private  
    $password;  
    private  
    $accessLevel;  
  
    public function __construct($idUser="",  
                                $identifiant="",  
                                $userName="",  
                                $password="",  
                                $accessLevel="") {  
        $this->idUser = $idUser;  
        $this->identifiant = $identifiant;  
        $this->userName = $userName;  
        $this->password = $password;  
        $this->accessLevel = $accessLevel;  
    }  
}
```

```
public function setIdUser($idUser){  
    $this->idUser = $idUser;  
}  
  
public function getIdUser(){  
    return $this->idUser;  
}  
  
public function  
setIdentifiant($identifiant){  
    $this->identifiant =  
    $identifiant;  
}  
  
public function getIdentifiant(){  
    return $this->identifiant;  
}
```

**DONE :**  
Sécurité

**DOING :**  
DAO - DTO

**TODO :**  
Les Libs -  
config

# Les Librairies nécessaires

<?php **lib-PDO.php**

```
// les bibliothèques dont je vais
avoir besoin

require_once 'conf/config.php';

// fonction qui va utiliser le
// rappeler le code de connexion
function getPDO() {
    $options = [
        PDO::ATTR_ERRMODE =>
        PDO::ERRMODE_EXCEPTION
    ];

    return new PDO(DSN,
DB_USER, DB_PASS, $options);
}
```

Pour La connexion a la base de données

```
/*
 * Configuration pour la connexion BD
 */

define('SERVER', '127.0.0.1');
define('PORT_MYSQL', '3306');
define('DB', 'cimetiere');
define('DSN', 'mysql:host=' . SERVER . ':' . PORT_MYSQL .
':3306' . '; dbname=' . DB . ';charset=utf8'); // 
;dbname=cimetiere;charset=utf8');
define('DB_USER', 'root');
define('DB_PASS', "");
```

**DONE :**  
DAO - DTO

**DOING :**  
Les Libs - config

**TODO :**  
Code  
« Inscriv. »

# Les Librairies nécessaires

```
<?php
function getViewContent($view, array $params = []){
    //extrait le nom de ma variable: la cle ; la valeur : la valeur
    extract($params);
    ob_start();
    require ROOT_FOLDER."/views/$view.php";
    $content = ob_get_clean();
    return $content;
}
/**
 * @param $view
 * @param array $params
 * @param string $layout
 * @return string
 */
function getResponse($view, array $params = [], $layout =
'default-layout'){
    //Récupération du contenu de la vue
    $viewContent = getViewContent($view, $params);
    $params['viewContent']= $viewContent;

    //Application de la vue sur le gabarit
    return getViewContent($layout, $params);
}
```



**DONE :**  
DAO - DTO

**DOING :**  
Les Libs - config

**TODO :**  
Code  
« Inscript. »

# Le Code

```
<form method="POST" action="index.php?  
controller=admin/actionController/ajoutInscriptionController">
```

```
    <div class="form-group">  
        <label>Recopier l'identifiant:  
            <input type="text"
```

```
                name="identifiant"  
                placeholder="Recopiez l'identifiant"  
                class="form-control"  
                value="ffffff">
```

```
        </label>  
    </div>
```

```
    <div CLASS="form-group col-md-1">  
    </div>
```

```
    <div class="form-group col-md-4">  
        <label>Votre Nom  
            <input type="text"
```

```
                name="userName"  
                placeholder="votre nom"  
                class="form-control"  
                value="bbb">
```

```
        </label>  
    </div>
```

The screenshot shows a web interface titled "Gestion Agent Territorial". A red oval highlights the URL in the browser's address bar: "index.php?controller=admin/actionController/ajoutInscriptionController". The main content area is titled "L'Inscription d'un agent territorial". It contains a "Générateur d'Identifiant" section with a "Génération" button and a field labeled "Votre identifiant généré:" containing "ffffff". Below this is a "Recopier l'identifiant:" field containing "ffffff". The form fields for user input are: "Votre Nom" (value "bbb"), "Confirmation le Nom" (value "bbb"), "Mot de passe" (value "....."), and "Confirmation du mot de passe" (value "....."). A large "Inscription" button is at the bottom right.

**DONE :**  
Les Libs -

config  
**DOING :**  
Code

«View Inscript. »

**TODO :**  
Code controller

2

# Controller: ajoutAgentController.php

DONE :

Code

«View Inscript. »

DOING :

Code controller 2

TODO :

Code Modification

```
// Enregistrement du message en variable  
de session
```

```
$_SESSION['message']= $message;
```

```
// initialisation
```

```
$pdo = getPDO();
```

```
$dao = new UserDAO($pdo);
```

```
$dto = new UserDTO();
```

```
// Hydratation
```

```
$dto->setIdentifiant($identifiant);
```

```
$dto-
```

```
>setUserName($userName);
```

```
$dto->setPassword($password);
```

```
// Enregistrement  
$dao->save($dto);
```

```
// Enregistrement du message en  
variable de session
```

```
$message = "L'agent territorial ".  
$userName . ' a été correctement enregistré dans  
la base de données';
```

```
$_SESSION['flash'] = $message;
```

```
// Redirection
```

```
header('location:index.php?  
controller=admin/adminController/inscriptionContr  
oller');
```

```
}//fin
```

# Modification

Gestion Agent Territorial

La modification d'un agent territorial

| id | Identifiant | Nom Agent    | Password |          |
|----|-------------|--------------|----------|----------|
| 1  | philou      | geronimo8080 | caimen   | Modifier |
| 9  | fffff       | pascal       | fffff    | Modifier |
| 12 | XFhuqh      | Nathalie     | chanel   | Modifier |
| 13 | hC3ImK      | Basuyau      | philippe | Modifier |
| 14 | 0hic7       |              |          | Modifier |
| 15 |             |              |          | Modifier |

**contrôleur 1:**

```
// Ma bibliothèque
require 'lib/lib-
controllerModificationAgent.
php';
```

**//Initialisation message  
Flash**

```
$flash = '';
if (isset($_SESSION['flash']))
{
    $flash
```

**//--- pour le tableau -----  
//j'ai besoin pour l'initialisation**

```
$pdo = getPDO();
$dao = new UserDao($pdo);
$dto = new UserDTO();
```

**DONE :**  
Code controller 2

**DOING :**  
Code Modification

**TODO :**  
Code Suppression

**// je vais générer un tableau pour y  
mettre mes données**

**// avec findAll je veux toutes les données**

```
$listUserName = $dao->findAll();
```

**// puis le formater**

```
$ContentTabUserName =
tableauUserName($listUserName);
```

**//--- fin du tableau -----**

**//affichage de la vue**

```
echo
getResponse('admin/gestionAgent/modificationAgent',
[
```

```
    'message' => $message,
    'titre' => $titre,
    'flash' => $flash,
    'ContentTabUserName'=>
$ContentTabUserName
```

# Modification

**Le Contrôleur 1 renvoie la vue**

```
<?php if ($flash != '') ?>
  <div class="alert alert-danger">
    <?php echo $flash; ?>
  </div>
<?php endif ?>
<?php
include(TEMPLATE_DIR .
'aside/asideAdmin.php'); ?>
<!--<hr> pour mettre une ligne
-->

<section class="col-md-9">
  <h3><?php echo $titre ?>
</h3>
<!-- <hr>-->
<br>
```

```
<div class="jumbotron">
  <h4>La modification d'un agent territorial</h4>
  <br>
  <fieldset class="row">
    <form action='index.php?
controller=admin&actionController/
modifAgentController'
method='POST'>
      <div>
        <?php echo $ContentTabUserName; ?>
      </div>
      <form action="" method="">
        <table>
          </table>
        </form>
    </fieldset>
  </div>
```



# Modification

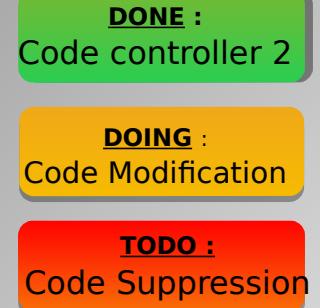
**La vue ModificationAgent envoie vers le contrôleur qui va traiter l'information**

```
$idUser = filter_input(INPUT_POST, 'idUser', FILTER_VALIDATE_INT);
$identifiant = filter_input(INPUT_POST, 'identifiant', FILTER_SANITIZE_STRING);
$userName = filter_input(INPUT_POST, 'userName', FILTER_SANITIZE_STRING);
$password = filter_input(INPUT_POST, 'password', FILTER_SANITIZE_STRING);
$btModifierAgent = filter_input(INPUT_POST, 'btModifierAgent',
FILTER_SANITIZE_STRING);
```

```
// initialisation
$pdo = getPDO();
$dao = new UserDAO($pdo);
$dto = new UserDTO();

/*
 * Modifier un agent territorial
 */
if ($btModifierAgent == "Modifier") {

    //on vérifie que le champs Nom de l'agent n'est pas vide au cas où
    if (isset($btModifierAgent) && ($userName != "") && ($password != "")) {
```



# Modification

**Le Contrôleur 2 récupère les informations les contrôles et les traitent ci-dessous  
Et renvoie une fois fini sur le 1<sup>er</sup> contrôleur pour afficher le tableau mis à jour**

```
// hydratation
    $dto->setidUser($idUser);
    $dto->setIdentifiant($identifiant);
    $dto->setUserName($userName);
    $dto->setPassword($password);
// Enregistrement de la modification
    $dao->save($dto);
// Message Flash en session
    $message = "L'agent territorial " . $userName . ' a été correctement
modification';
// Message flash
    $_SESSION['flash'] = $message;
// Redirection header('location:index.php?
controller=admin/adminController/modificationAgentController');
}
}
```



# La Librairie qui me permet de mettre Un tableau à la place d'un papier par rapport à la BD

```
/*
 * @param array $data
 */
function
tableauUserName( $data ) {

    $x = count($data);

    $html ='';
    $html .='<table class="table
table-striped">';

    //--- Les Entêtes
    $html .='<thead>';
    $html .='<tr>';
    $html .='<th>id</th>';
    $html .='<th>Identifiant
</th>';
    $html .='<th>Nom
Agent</th>';
    $html
    .='<th>Password</th>';
    $html .='<th></th>';
    $html .='</tr>';

    $x = count($data);

    for ($i = 0; $i < $x; $i++) {
        $html .= "<tr>";
        $html .='<td>' . $data[$i][0] . '</td>';
        $html .='<td>' . $data[$i][1] . '</td>';
        $html .='<td>' . $data[$i][2] . '</td>';
        $html .='<td>' . $data[$i][3] . '</td>';
        $html .='<td>' . $data[$i][4] . '</td>';
        $html .='</tr>';

    }
}
```

## //--- Le Body

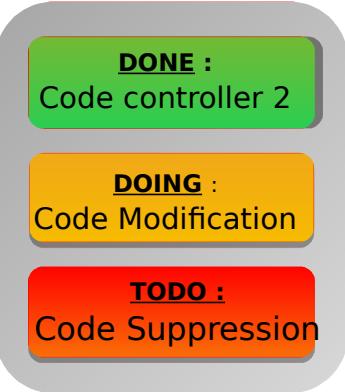
```
$html .='<tbody>';

for ($i = 0; $i < $x; $i++) {
    $html .= "<tr>";
    $html .='<td>' . $data[$i][0] . '</td>';
    $html .='<td>' . $data[$i][1] . '</td>';
    $html .='<td>' . $data[$i][2] . '</td>';
    $html .='<td>' . $data[$i][3] . '</td>';
    $html .='<td>' . $data[$i][4] . '</td>';
    $html .='</tr>';

}
```



# La Librairie qui me permet de mettre Un tableau à plat par rapport à la BD



**//--- pour l'identifiant**

```
$html .='<td>';
$html .= $data[$i]['identifiant'];
// je cache le champs suivant car je veux pas
// pouvoir le modifier avec :type="hidden"
$html .= '<input type="hidden" value="'. $data[$i]
["identifiant"].'" name="identifiant" />';
$html .='</td>';
```

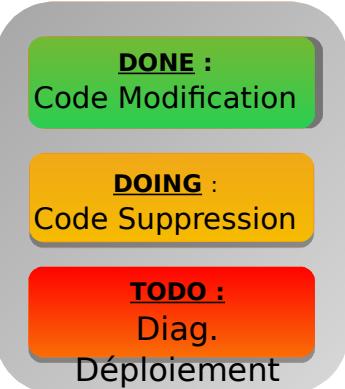
**//--- pour les Nom d'Agent**

```
$html .='<td>';
// si je ne veux pas afficher le nom en plus dans
// les champs pour la modification
// je commande la ligne suivante
```

```
//$html .= $data[$i]['userName'],
$html .= '<input type="text" value="'. $data[$i]
["userName"].'" name="userName" />';
$html .='</td>';
```

	id	Identifiant	Nom Agent	Password	
1	1	philou	geronimo8080	caimen	Modifier
9	9	fffff	pascal	fffff	Modifier
12	12	XPhuqh	Nathalie	chanel	Modifier
13	13	nc3ImK	Basuyau	philippe	Modifier
14	14	0hic7J	fifi	brindacier	Modifier
15	15	a3dW54	Mussart	wxcvbn	Modifier

# Même principe pour la vue suppression

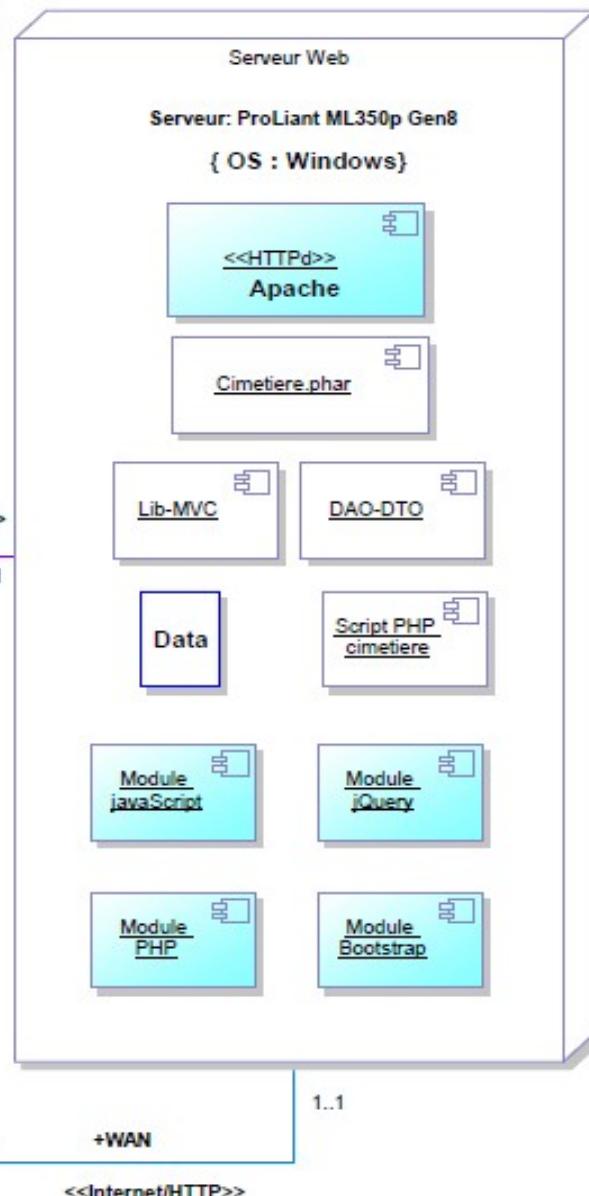
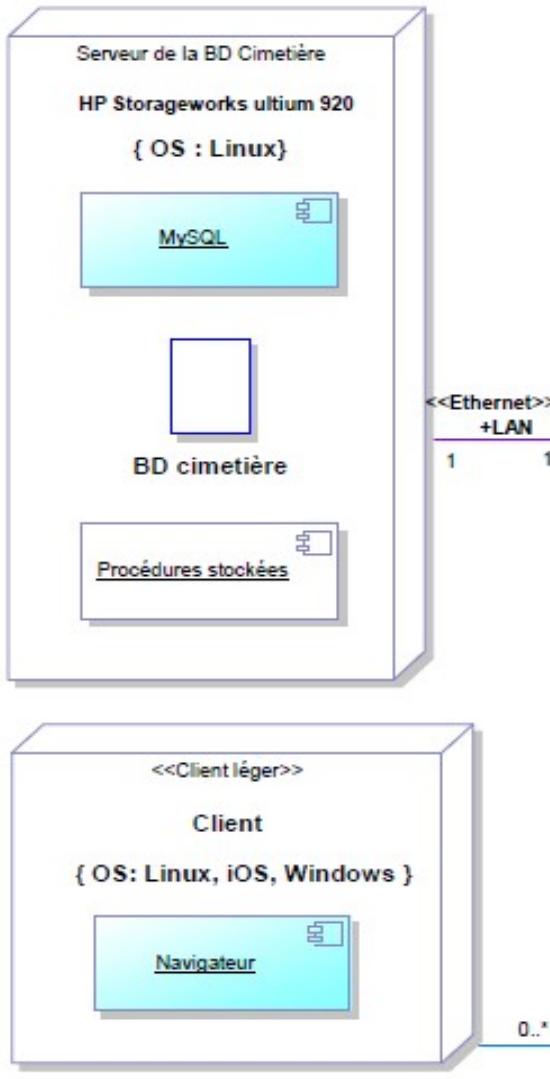


La suppression d'un agent territorial				
<b>id</b>	<b>Identifiant</b>	<b>Nom Agent</b>	<b>Password</b>	
1	philou	geronimo8080	caimen	<a href="#">Supprimer</a>
9	fffffff	pascal	fffffff	<a href="#">Supprimer</a>
12	XFhuqh	Nathalie	chanel	<a href="#">Supprimer</a>
13	hC3ImK	Basuyau	philippe	<a href="#">Supprimer</a>
14	0hic7J	fifi	brindacier	<a href="#">Supprimer</a>
15	a3dW54	Mussart	wxcvbn	<a href="#">Supprimer</a>

# Diagramme de Déploiement

Légende

- Acheté sur étagère
- Produit ad hoc



Se diagramme sera fournis au technicien réseau qui procèdera à l'installation de la totalité des éléments

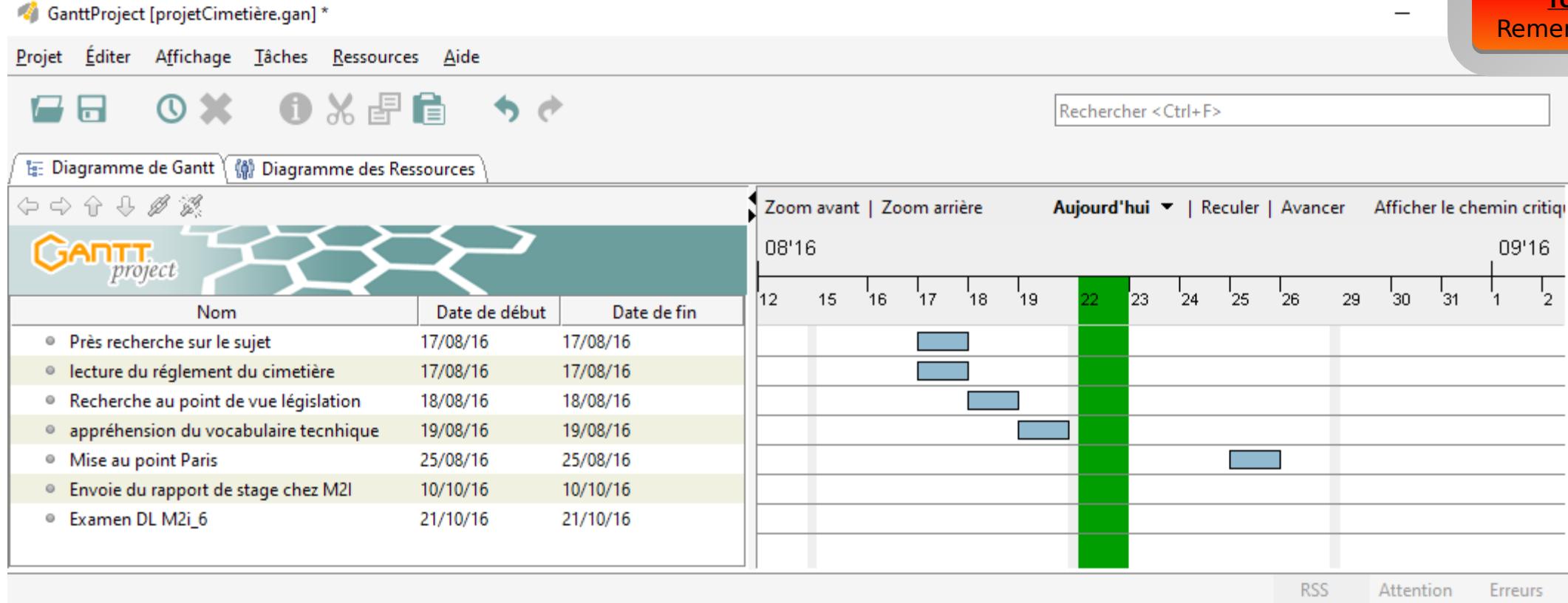
**DONE :**  
Code Suppression

**DOING :**  
Diag. Déploiement

**TODO :**  
Diag. Gantt

Modèle orienté objet	
Modèle : Diagramme de Déploiement Projet Cimetiere	
Package :	
Diagramme : DiagrammeDeploiement_1	Date: 10/10/2016
Auteur : Philippe Basuyau	
Version: 1.0	

# Diagramme de Gantt sur l'évolution du projet



Remerciement à mes formateurs :

**Mr Pascal Buguet**  
**Mr Sébastien**  
**Maloron**

Remerciement à la Mairie d'Estrées Saint Denis pour le projet  
Et **Mr Richard Chaigneau**

Remerciement à **mon épouse Nathalie** pour son soutien de tous les jours

© Philippe Basuyau 2016  
Projet Cimetière