

# **Développement d'un site web pour les demandes CEDR**

(Rapport de stage)

## **Typologie du rapport**

(Rapport de stage dans le cadre de l'obtention d'un DUT  
Informatique)

**Auteur : MINBIELLE Morgan**

**Deuxième année de DUT Informatique**

Formation initiale

Soutenance prévue le 20 juin 2019

# SOMMAIRE

<b>Remerciements</b>	<b>2</b>
<b>Introduction</b>	<b>3</b>
<b>1/ Présentation de la collectivité</b>	<b>4</b>
<b>2/ Contexte du projet</b>	<b>6</b>
<b>2.1 Description de la direction</b>	<b>6</b>
<b>2.2 Description du processus des demandes CEDR</b>	<b>7</b>
<b>3/ Projet et déroulement</b>	<b>9</b>
<b>3.1 Description du projet</b>	<b>9</b>
<b>3.2 Déroulement du projet</b>	<b>10</b>
<b>3.2.1 Prise de poste</b>	<b>10</b>
<b>3.2.2 Conception des tables</b>	<b>11</b>
<b>3.2.3 Structure du site</b>	<b>13</b>
<b>3.2.4 Conception des fonctionnalités</b>	<b>15</b>
<b>3.2.5 Interfaces</b>	<b>16</b>
<b>3.2.6 Contrôleurs et vues</b>	<b>18</b>
<b>3.2.7 Authentification</b>	<b>18</b>
<b>3.2.8 Paramètres</b>	<b>19</b>
<b>3.2.9 CEDR</b>	<b>20</b>
<b>3.2.10 Demandes et pièces-jointes</b>	<b>21</b>
<b>Conclusion</b>	<b>28</b>
<b>Glossaire</b>	<b>29</b>
<b>Annexes</b>	<b>31</b>

## **REMERCIEMENTS**

Je tiens à témoigner toute ma gratitude à madame Jacqueline JAES, ma tutrice de stage qui est chef de développement logiciel au sein de la direction de l'informatique et des technologies de l'information, pour m'avoir permis de découvrir les étapes nécessaires au développement Web dans une collectivité, ainsi que d'avoir partagé ses connaissances afin que je puisse mener à bien mon stage.

Je remercie également tous les membres de la direction pour l'accueil et la bonne ambiance de travail qu'ils ont apportés tout au long de mon stage.

# INTRODUCTION

Durant la deuxième année du Diplôme Universitaire de Technologie (DUT), formation suivie à l'Institut Universitaire de Technologie de Villetaneuse (Université Paris XIII), un stage en entreprise est prévu. Il s'agit d'un stage d'analyste programmeur eb d'une durée de 10 à 12 semaines. J'ai effectué ce stage à la DITI (Direction Informatique et Technologie de l'Information) de la mairie d'Aubervilliers, sous la tutelle de madame Jacqueline JAES.

La mairie d'Aubervilliers est une collectivité, compartimentée en quatre pôles regroupant l'ensemble des services municipaux par domaine de compétence, qui a pour but de répondre aux besoins des habitants de la commune grâce à ses différents services.

La DITI est un service qui permet de mettre en place des solutions logiciels, matérielles et la gestion des infrastructures informatiques au sein de la mairie, afin d'assurer le bon fonctionnement des autres services.

Dans un premier temps, je vous présenterai l'organisation de la collectivité puis le contexte dans lequel s'inscrit mon projet. Ensuite, je vous exposerai en quoi consiste mon projet et son déroulement.

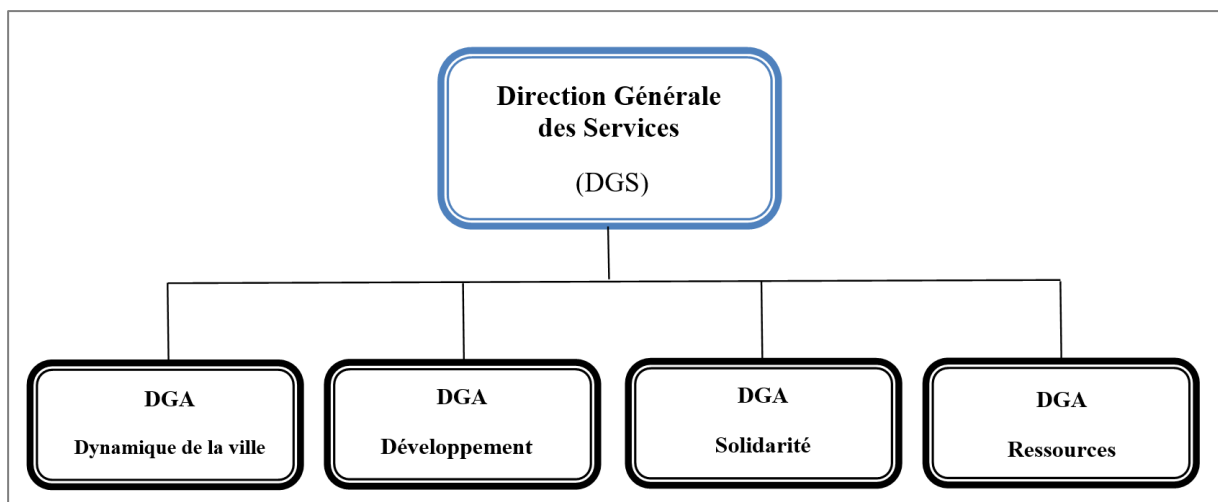
# 1/ PRÉSENTATION DE LA COLLECTIVITÉ

L'administration communale dans laquelle j'effectue mon stage se situe dans la ville d'Aubervilliers, en Seine-Saint-Denis. La mairie compte 1700 agents pour appliquer les politiques municipales. Elle est placée sous une double autorité :

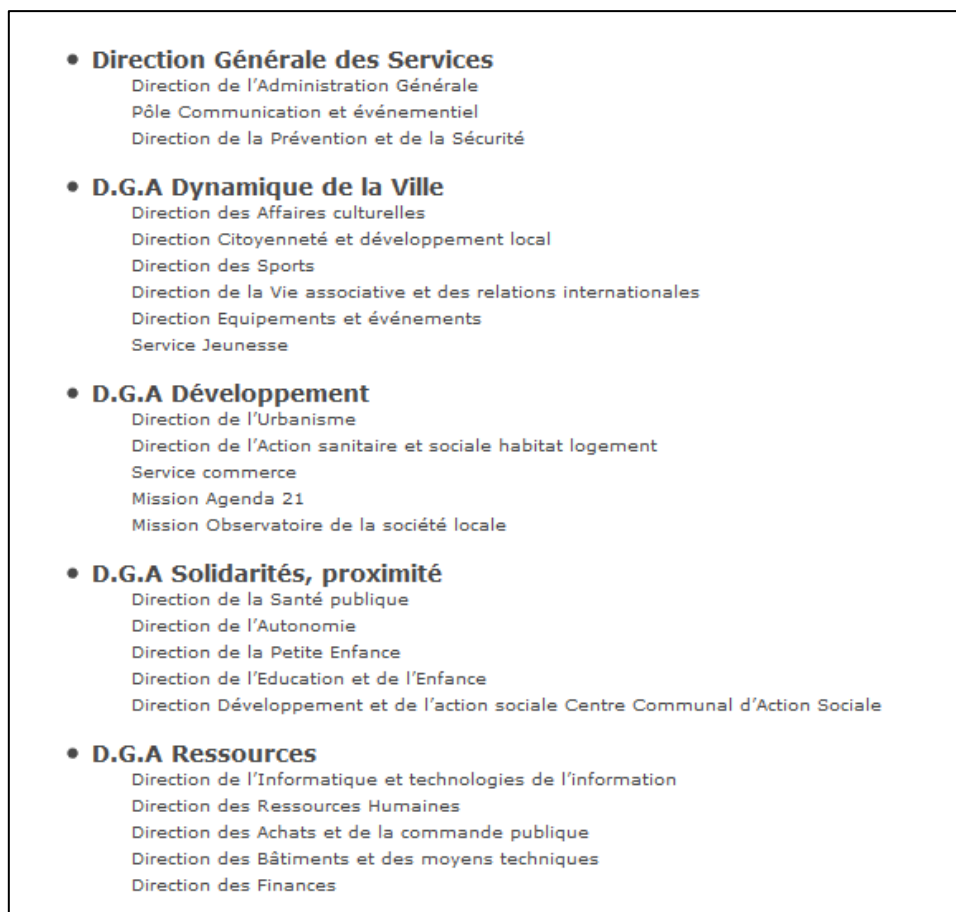
- La maire (en sa qualité d'autorité territoriale) ;
- Le directeur général des services.

Cette administration est organisée en Directions Générales Adjointes (DGA) liées à la Direction Générale des Services (DGS) et en directions, services et secteurs dépendant des DGA.

La structure contient quatre pôles regroupant l'ensemble des services municipaux par domaine de compétence.



*Figure 1 - Structure de l'organisme*



*Figure 2 – Organigramme détaillé de la mairie*

## **2/ CONTEXTE DU PROJET**

### **2.1 DESCRIPTION DE LA DIRECTION**

La DITI est en relation directe avec la DGA ressources, mais aussi avec tous les autres services de la mairie (pour effectuer la maintenance et le suivi logiciel, pour mettre en place des solutions logiciels ou faire l'intermédiaire avec des prestataires). Ses clients sont internes à la municipalité, mais il peut faire appel à des ressources ou prestataires externes.

Le service peut être consulté afin d'aider à améliorer et simplifier les procédures administratives internes, c'est dans ce cadre précis que mon projet s'établit.

Les ressources utilisées dans le cadre de mon projet sont essentiellement des outils logiciels open sources (comme Talend Open Studio), propriétaires ou des outils logiciels internes (intranet, sites web internes, logiciels de gestion, etc.). Niveau matériel, le service dispose de serveurs et d'ordinateurs pour le développement et la gestion du parc de la collectivité.

La DITI compte un peu plus d'une dizaine de membres qui possèdent différentes fonctions (chef de projet, gestion des serveurs, gestion des parcs, de la téléphonie, développement, etc.). Mais pour ce projet, l'équipe est composée de ma tutrice et de moi-même.

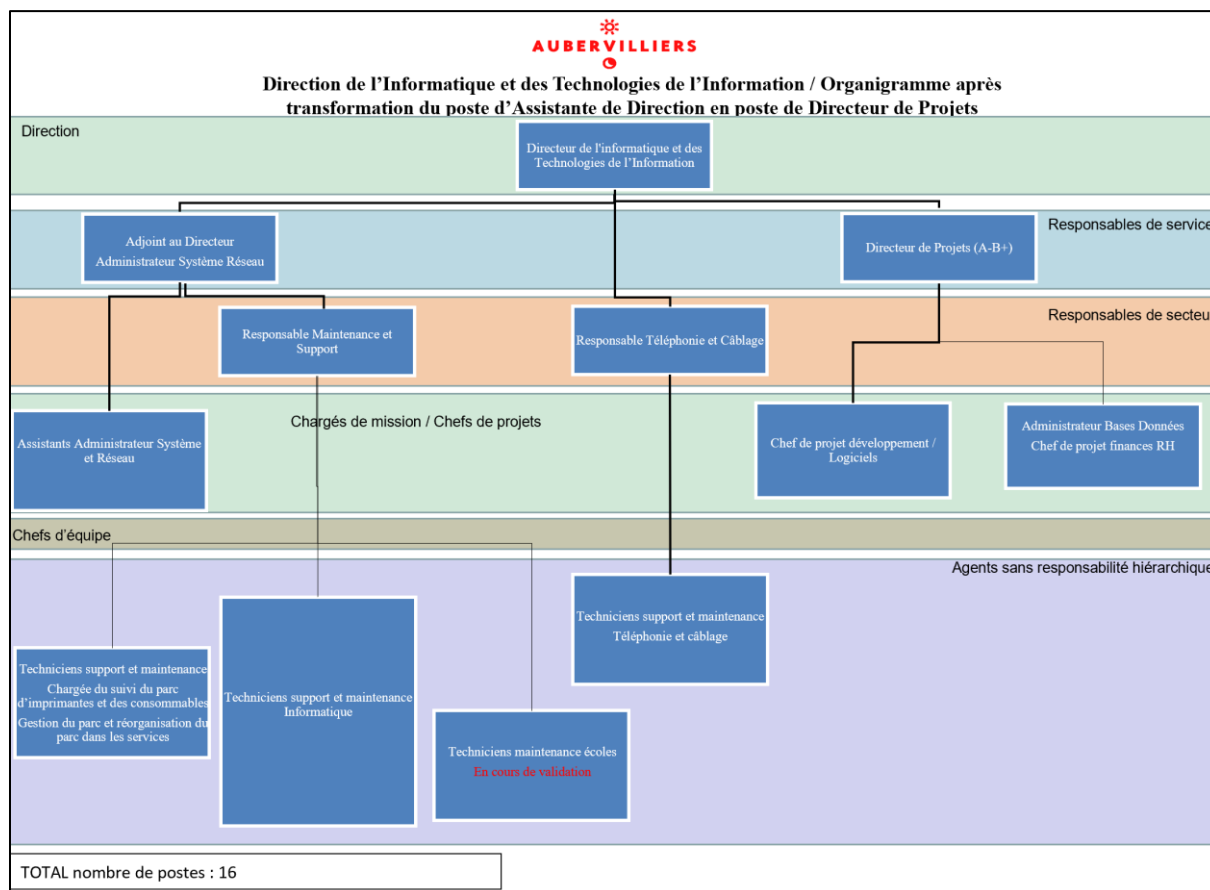


Figure 3 - Organigramme de la D.I.T.I.

## 2.2 DESCRIPTION DU PROCESSUS DES DEMANDES CEDR

Les procédures de recrutement ou de remplacement sont actuellement saisies par une direction ou un service demandeur et transmises au service DC (Développement des Compétences) après avoir reçu l'accord de la hiérarchie. Les informations concernant les demandes sont renseignées dans un document Word, puis copiées dans un tableau Excel commun au DC. Après l'analyse du service DC et de la CEDR (Commission d'Évaluation des Demandes de Recrutement), le service DC saisi les décisions dans le tableau Excel et notifie



le demandeur par mail. On remarque donc ici qu'il s'agit d'une procédure lourde pouvant générer des problèmes de partage dus à la non-standardisation de la saisie des informations.

C'est donc à la suite de cette problématique que le service a contacté le DITI afin de créer une application en extranet (site web accessible en interne et en externe). De plus, un cahier des charges fut établi en début d'année pour que le projet puisse débuter lors de ma période de stage.

## **3/ PROJET ET DÉROULEMENT**

### **3.1 Description du projet**

Le projet a donc pour but de mettre en place un site web en extranet, c'est à dire qu'il puisse être accessible en interne et en externe (par exemple par une école hors réseau de la ville qui souhaite commencer une procédure de recrutement). Ce site propose donc de consulter les différentes séances CEDR disponibles pour déposer une demande et les traiter, tout cela à l'aide d'une seule plateforme dans le but de simplifier le suivi d'une demande.

On distingue ici deux types de demandes :

- Plénière : demande de validation de poste permanent ;
- Restreinte : demande de remplacement.

Le type de demande influence donc le formulaire que devra saisir le demandeur, car les informations nécessaires pour un remplacement ne sont pas les mêmes que pour une prise de fonction (un remplacement nécessite d'indiquer qui est l'agent à remplacer par exemple).

De plus, on identifie également deux types d'utilisateurs qui auront accès à ce site :

- Le demandeur : un compte composé de son adresse mail Zimbra et d'un mot de passe unique (qui sera le même pour tous les demandeurs) sera utilisé pour déposer les demandes ;
- Le gestionnaire : le personnel du service DC aura un accès nominatif pour gérer les demandes.

En fonction de l'utilisateur, des fonctionnalités peuvent différer. Par exemple, un demandeur ne peut voir que les séances de CEDR non archivées ainsi que ses demandes (qu'il pourra modifier avant la clôture de la séance liée à la demande). En revanche, les gestionnaires peuvent voir toutes les demandes et accéder aux séances CEDR archivées. De plus, il existe un unique administrateur (ma tutrice) peut également paramétrer les tables des directions, grades, catégories ou autres (qui correspondent à des champs à renseigner durant une demande) et la liste des gestionnaires.

Les fonctionnalités ont été définies avec le cahier des charges validé par les clients en février. Le site devra donc, entre autres, permettre à un demandeur de se connecter, d'accéder à la liste des différentes séances de CEDR non clôturées, de déposer une demande pour une séance choisie (de la compléter avec des pièces jointes, de pouvoir la modifier ou la supprimer) et de voir la liste de ses demandes effectuées pour consulter l'état de celles-ci (si elles ont été annulées ou acceptées). Quant au gestionnaire, il peut gérer les séances CEDR et les demandes (ajout, modification, suppression), saisir les avis des ressources humaines et les décisions des commissions (favorables ou non), gérer les tables des différents services, grades et secteurs. De plus, le site permettra l'extraction des demandes afin de disposer d'un document Excel contenant l'essentiel des informations nécessaires aux directions.

Les langages de développement utilisés durant ce projet sont le HTML, CSS, PHP et JavaScript. L'utilisation de Bootstrap, templates et de logiciels open sources comme Talend Open Studio, WampServer ou MySQL Workbench a également été nécessaire.

## **3.2 Déroulement du projet**

### **3.2.1 Prise de poste**

Conformément à ma convention, mon stage a débuté au début du mois d'avril au sein du service DITI, dans lequel j'ai fait équipe avec madame JAES, ma tutrice, pour ce projet. De la documentation m'avait déjà été transmis à la suite de mon entretien fin février car le projet avait déjà été en phase d'étude et de planification. J'ai donc eu un briefing avec ma tutrice qui m'a expliqué le projet en détails pour commencer la phase de conception de la base de données qui servira pour le site. Ma tutrice m'a donc fourni toute la documentation (cahier des charges complet, documentation sur certains logiciels, code de base d'anciens projets similaires) et les outils nécessaires (comme MySQL Workbench ou Talend Open Studio) pour débiter convenablement le projet.

1	<b>Planning de développement de l'application de la gestion des demandes en CEDR</b>						
2							
3	<b>Tâches</b>	<b>Service</b>	<b>Durée (en jours)</b>	<b>Date début</b>	<b>Date fin</b>	<b>Statut</b>	<b>Priorité</b>
4	<b>1. Base de données</b>						
5	1.1. Conception de la base	DITI (JJ)	2	02/04/2019		0%	1
6	1.2. Création de la base sur le serveur	DITI (JJ)	2			0%	1
7	1.3. Import des données de sedit RH dans la base (Filière, Cadre, Grade...)	DITI (JJ)	3			0%	1
8	1.4. Import des données excel fournit par le service DC (DGA/DGS, Direction, Service...)	DITI (JJ) + DC	3			0%	1
9	1.5. Saisie manuelle de certaines données (utilisateur, motifs...)	DITI (JJ) + DC	2			0%	1
10	<b>2. Structure de l'application</b>						
11	2.1. Mise en place du template (CSS)	DITI (JJ)	5			0%	1
12	2.2. Paramétrage de l'application	DITI (JJ)	5			0%	1
13	2.3. Création de tous les contrôleurs (MVC)	DITI (JJ)	2			0%	1
14	2.4. Création de toutes les fonctions (PHP + JS)	DITI (JJ)	10			0%	1
15	2.5. Mise en place des plugins nécessaires	DITI (JJ)	1			0%	1
16	<b>3. Développement de chaque cas d'utilisation</b>						
17	3.1. Compte > Connexion	DITI (JJ)	0,5			0%	1
18	3.2. Compte > Déconnexion	DITI (JJ)	0,5			0%	1
19	3.3. Compte > Modifier mon mot de passe	DITI (JJ)	1			0%	1
20	3.4. CEDR > Planning des séances (Filtre + Actions)	DITI (JJ)	2			0%	1
21	3.5. CEDR > Ajouter une séance	DITI (JJ)	1			0%	1
22	3.6. CEDR > Modifier une séance	DITI (JJ)	1			0%	1
23	3.7. CEDR > Supprimer une séance	DITI (JJ)	0,5			0%	1
24	3.8. CEDR > Archiver une séance	DITI (JJ)	1			0%	1
25	3.9. Demande > Consulter les demandes d'une séance (Filtre + Actions)	DITI (JJ)	2			0%	1
26	3.10. Demande > Ajouter une demande à une CEDR (Brouillon + Envoie)	DITI (JJ)	4			0%	1
27	3.11. Demande > Modifier une demande saisie	DITI (JJ)	2			0%	1
28	3.12. Demande > Supprimer / Annuler une demande saisie	DITI (JJ)	2			0%	1
29	3.13. Demande > Transmettre une demande saisie aux interlocuteurs	DITI (JJ)	1			0%	1
30	3.14. Demande > Extraire les demandes d'une séance en xls	DITI (JJ)	7			0%	1

Figure 4 – Extrait du planning du projet

### 3.2.2 Conception des tables

Dans le but de développer les fonctionnalités du site web, il est nécessaire de créer en amont les tables afin d'avoir les interactions entre les données et l'utilisateur. Pour cela, j'ai dû apprendre à mettre en place un modèle de données, à partir du cahier des charges et des indications de ma tutrice, en utilisant l'application MySQL Workbench. Ainsi j'ai donc élaboré une base de données avec un peu plus de dix tables (dont celles contenant les données des divers services, secteurs ou directions au sein de la mairie, celles servant à l'indexation des demandes de recrutement, de leur type ou statut, etc.), qui constituent une base pour les fonctionnalités importantes du site (identification des demandeurs et des gestionnaires, gestion des demandes pour une procédure de recrutement, etc.).

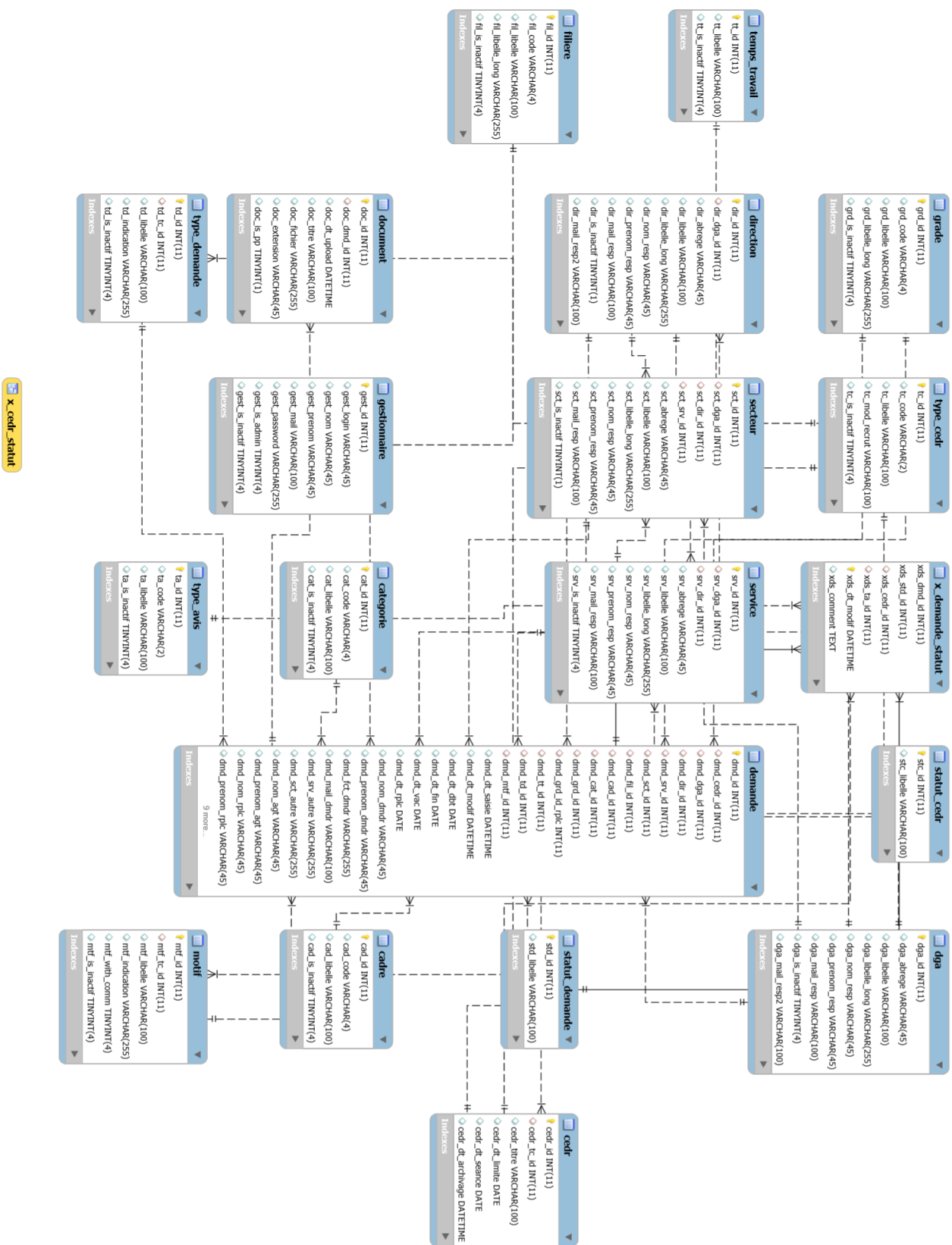


Figure 5 - Base de données CEDR

Afin de remplir les tables créées sur le réseau interne de la mairie, j'ai également dû utiliser un intégrateur de données (Talend Open Studio). Pour cela, j'ai utilisé de légers codes et des fichiers Excel pour compléter les différentes tables des services et directions de la mairie selon des paramètres bien définis concernant les différentes clés. La fin de la première semaine marque la fin du développement principal de la base de données (qui a subi de légères modifications sur certaines clés de tables lors du développement).

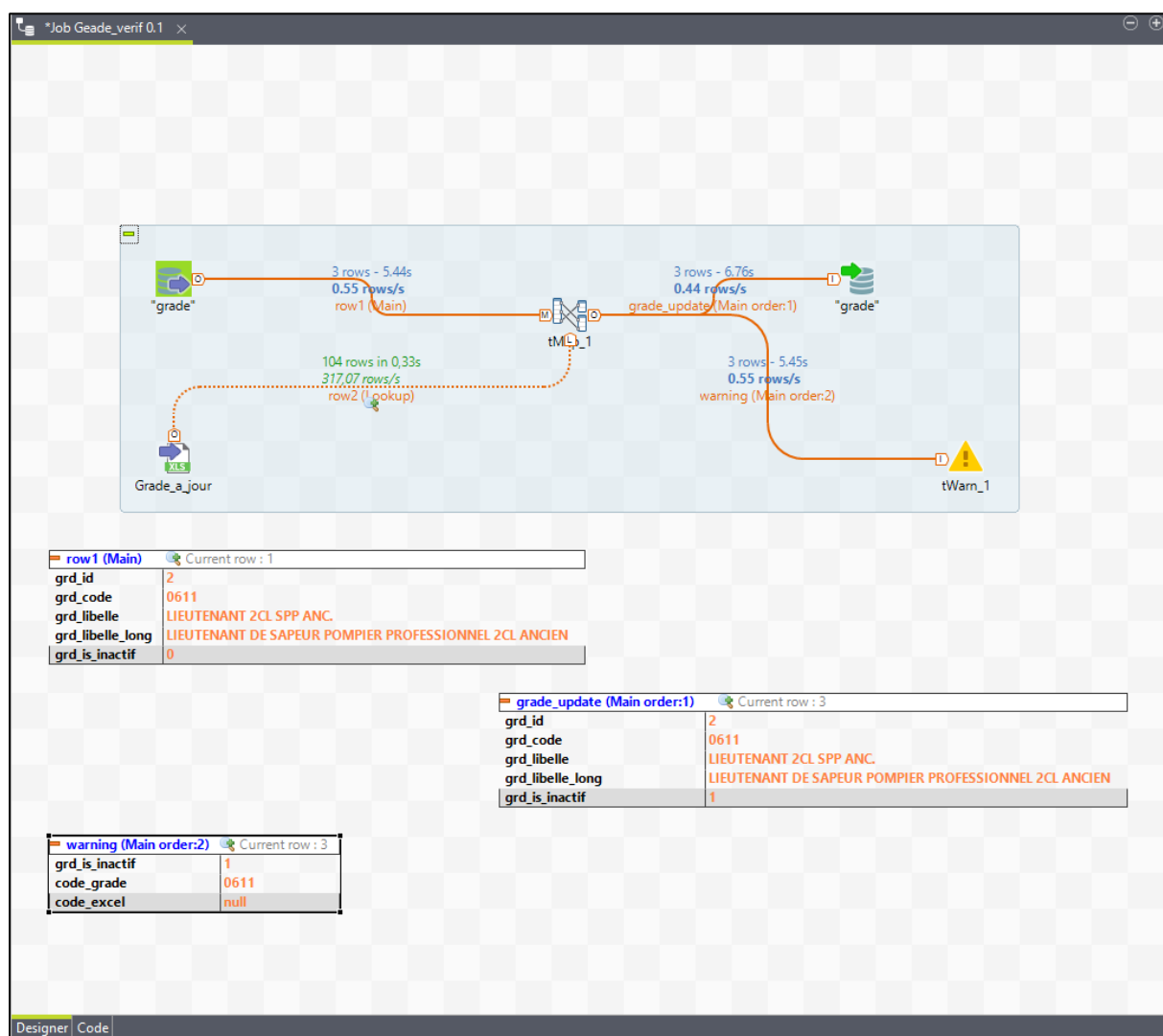


Figure 6 - Intégration des données

### 3.2.3 Structure du site

Le site utilise le principe d'architecture MVC (Modèle Vue Contrôleur) avec une fragmentation des différents éléments propre à la façon de coder de ma tutrice, cela m'a demandé un temps d'adaptation afin de comprendre comment coder de manière efficace et pouvoir faire une intégration optimale de mon code.

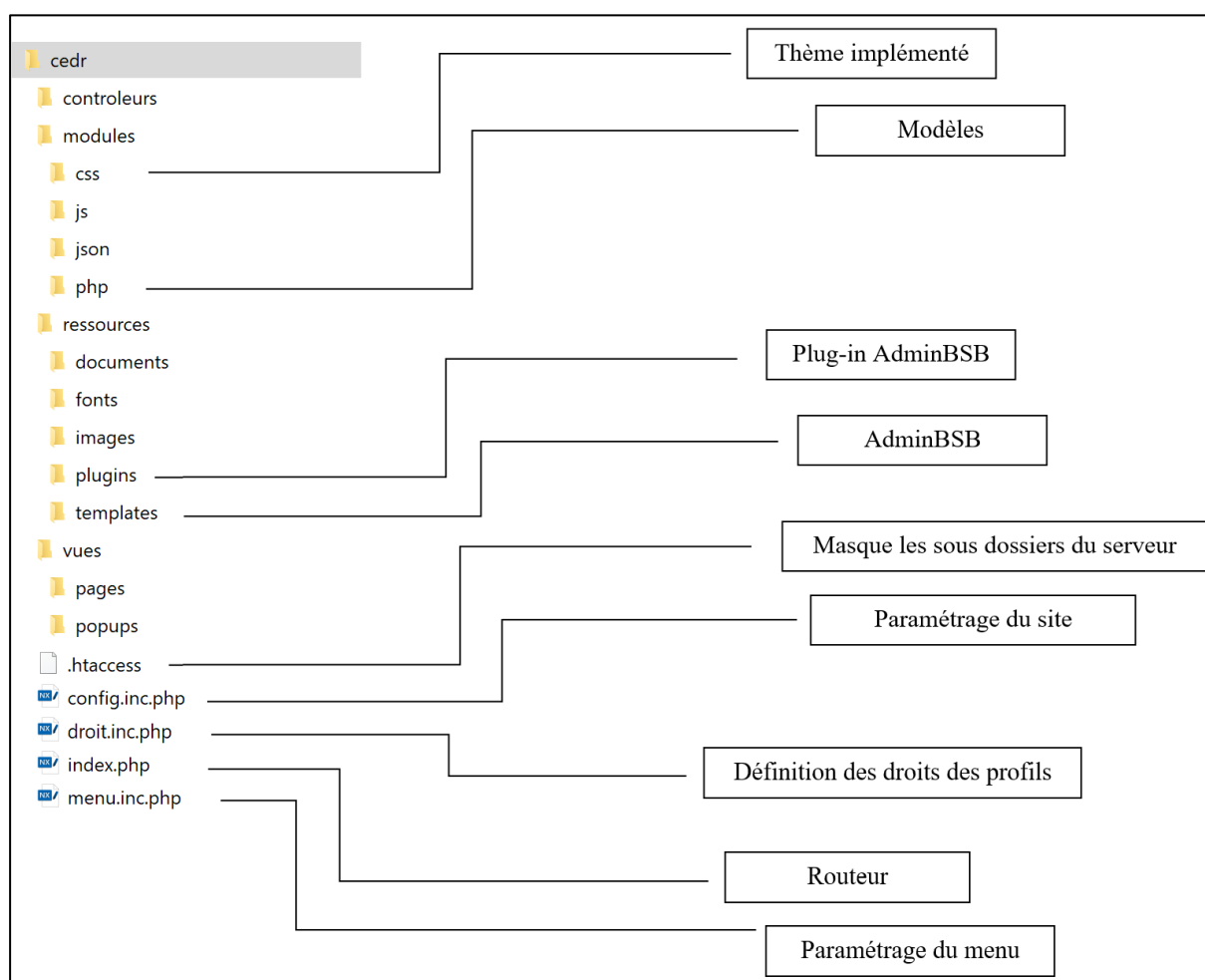


Figure 7 - Structure du site MVC

Ici, les modèles se retrouvent dans le dossier “modules”, qui contient les fichiers comportant les données à afficher. Par exemple, le sous-dossier « php » regroupe les différents fichiers servant à définir les actions avec la base de données en fonction de la table à utiliser. Les contrôleurs, qui servent à actualiser les vues en fonction des données des

modèles, sont dans le dossier “controleurs”, il y en a un pour la gestion des CEDR, des demandes, des comptes, des paramétrages, contact et des documents. Les vues, qui sont les pages d’affichage du site, elles, sont dans le dossier “pages” du dossier “vues”.

### **3.2.4 Conception des fonctionnalités**

À la suite de la finalisation de la base, nous avons pu passer au développement des fonctionnalités, tout en continuant à étudier si d’autres fonctions devaient être ajoutées ou modifiées. Ainsi, les fonctions principales que l’on retrouve dans la majorité des fichiers PHP dédiés aux tables sont l’ajout (d’un service ou d’une demande), la modification, la suppression et la recherche (avec filtre ou non, par exemple si on veut rechercher une séance plénière ou restreinte, qui débute ou finit à une date précise, etc.). Je suivais principalement le cahier des charges et le planning, tout deux établis antérieurement, mais j’échangeais également souvent avec ma tutrice afin d’avoir toutes les clés en main. Elle m’indiquait s’il y avait un nouvel élément à intégrer et quand il fallait le faire. L’ensemble des fonctions liées aux tables m’a pris 2 semaines de développement afin de m’adapter au code d’un autre projet similaire. La figure ci-dessous est un exemple du code contenu dans le fichier « `cedr.inc.php` » qui est dans le dossier « `modules/php/` ». On y voit le déroulé de l’insertion d’une CEDR qui nécessite un type (plénière ou restreinte), un titre, une date limite et une date de séance. Si l’une de ces informations est manquante, la fonction d’insertion retourne un tableau affichant l’erreur qui correspond, sinon les données sont traitées pour éviter l’ajout de caractères inadaptés ou de script d’une personne tierce et l’insertion s’effectue.



```

89 //Ajout d'une CEDR
90 function cedr_insert($idCEDR, $idTypCEDR, $titre, $dtLimite, $dtSeance)
91 {
92     //Test des erreurs
93     $lesErreurs = array();
94     if(!$idTypCEDR) $lesErreurs[] = "Le type de la CEDR doit être défini.";
95     if(!$dtSeance) $lesErreurs[] = "La date de la séance est à renseigner.";
96     if($dtSeance < date_CompareTo(date("d/m/Y"), $dtSeance, ">")) $lesErreurs[] = "La date de la CEDR doit être supérieur à la date du jour!";
97     if($dtSeance < date_ChrFormat($dtSeance) $lesErreurs[] = "La date de la séance doit être au format JJ/MM/AAAA!";
98     if(!$dtLimite) $lesErreurs[] = "La date de limite de dépôt est à renseigner.";
99     if($dtLimite < date_CompareTo($dtLimite, $dtSeance, ">")) $lesErreurs[] = "La date limite doit être inférieur à la date de la CEDR!";
100    if($dtLimite < date_ChrFormat($dtLimite) $lesErreurs[] = "La date limite doit être au format JJ/MM/AAAA!";
101
102    //Traitement des variables
103    $idTypCEDR = mysql_EscapeString($idTypCEDR);
104    $titre = mysql_EscapeString(string_SetHtmlEntities($titre));
105    $dtLimite = mysql_EscapeString($dtLimite);
106    $dtSeance = mysql_EscapeString($dtSeance);
107
108    //Ajout dans la base
109    if(!$lesErreurs)
110    {
111        $req = "INSERT INTO cedr (cedr_id, cedr_titre, cedr_dt_limite, cedr_dt_seance)
112        VALUES ($idTypCEDR, '$titre', STR_TO_DATE('$dtLimite', '%d/%m/%Y'), STR_TO_DATE('$dtSeance', '%d/%m/%Y'))";
113        $conx = mysql_Connection();
114        mysql_query($conx, $req) or $lesErreurs[] = "<u>Erreur SQL (".__METHOD__."</u>: ".mysql_error($conx);
115        $idCEDR = mysql_insert_id($conx);
116        mysql_close($conx);
117    }
118
119    return $lesErreurs;
120 }

```

Figure 8 - Exemple de code d'insertion d'une séance de CEDR

Après avoir fini de développer ces fonctions, ma tutrice vérifiait que mon code correspondait à ce qui était prévu et l'optimisait par rapport à la base (de code) existante. Ensuite, pendant que ma tutrice prenait le relais en ajustant la base en fonction des nouvelles fonctionnalités que l'on avait ajoutées au péalable et des échanges avec les services qui utiliseront cet outil, elle m'a demandé de trouver des frameworks et des templates appropriés pour élaborer un nouveau type d'interface adaptatif à tout type d'écran.

### 3.2.5 Interface

Afin de définir une interface intuitive et différente de ce qui avait déjà été fait pour d'autres projets antérieurs, ma tutrice m'a demandé de trouver des templates adaptés et qui étaient « responsive » (c'est-à-dire, que le site puisse s'afficher sur tout type d'écrans). Ne connaissant pas les templates, ni Bootstrap, je me suis donc documenté sur internet et renseigné auprès de ma tutrice dans un premier temps, puis j'ai recherché des templates qui correspondaient aux critères mentionnés plus tôt. Deux templates ont été retenus, Dashgum et DB Admin. Une fois les templates sélectionnés, j'ai commencé par coder d'autres fonctionnalités et des vues afin d'avoir une première idée sur le rendu du travail (comme par exemple le contrôleur et les vues servant à l'authentification). Avec ma tutrice, nous avons décidé des éléments à enlever ou améliorer en fonction de la structure du code du template d'origine, de notre propre structure et de l'expérience utilisateur. À la suite de ce premier

rendu, certains critères manquants (comme des tableaux avec tris intégrés, pas d'options d'exportation ou peu d'options pour avoir un formulaire complet) ont mené à la recherche d'un autre template, AdminBSB. De cette manière, nous avons basé tout notre développement à partir de ce template en adaptant le code existant. Nous avons également compartimenté la structure de chaque page en plusieurs fichiers PHP complémentaires, un pour la barre de navigation horizontale (cf. figure 9), un pour le menu vertical situé sur la gauche (cf. figure 9) et un dernier intégrant le contenu de la page à afficher. La structure de chaque page utilise donc les mêmes fichiers, qui sont identifiable par leur suffixe : « v\_ », puis le contenu de la page est un fichier avec le suffixe « p\_ » qui est appelé en fonction du cas.

The screenshot shows the 'Gestion des Demandes de CEDR' application. The sidebar on the left contains navigation links: 'Profil Gestionnaire', 'CEDR', 'Planning des CEDR', 'Recherche de demandes', 'Support', and 'Contact'. The main content area displays the 'Planning des CEDR' section, which includes a table of CEDR sessions. The table has columns for 'Numéro', 'Date limite de dépôt', 'Date de la CEDR', 'Intitulé de la CEDR', 'Nombre de demandes', 'Statut', and 'Actions'. The table lists five sessions, with the first one highlighted. Below the table, there is a pagination bar showing 'Affichage de l'élément 1 à 5 sur 5 éléments' and 'Précédent' and 'Suivant' buttons. Red lines point from the sidebar and the table area to labels below the image.

**Fichiers avec préfixe "v\_"**

**Fichier contenu "p\_lstCEDR.php"**

Figure 9 – Page d'affichage de la liste des séances CEDR

### 3.2.6 Contrôleurs et vues

La suite du projet consistait à finir le contrôleur dédié à l'authentification et coder les autres qui sont respectivement dédiés à la gestion des CEDR, des demandes et au paramétrage de certaines tables. En effet, pour l'authentification, ma tutrice a réarrangé la disposition du code afin d'ajouter des vérifications de droits qui sont établies dans le fichier « droit.inc.php » (par exemple, un utilisateur qui n'est pas un demandeur n'a pas le droit d'accéder aux demandes des autres utilisateurs) et la vérification des e-mails (utilisant Zimbra) renseignés dans une autre base de données.

### 3.2.7 Authentification

J'ai donc commencé par développer le contrôleur `c_gérerCompte`, qui permet à un utilisateur de se connecter en fonction de son statut (demandeur ou gestionnaire). S'il est demandeur, il faut s'assurer que le mail correspond à un mail Zimbra et que le mot de passe corresponde au mot de passe unique, tandis que pour un gestionnaire il faut vérifier si le nom d'utilisateur correspond, ainsi que le mot de passe saisi est le même que le mot de passe stocké sur la base (qui est crypté selon le protocole sha1). Ce contrôleur permet également la modification des informations du compte et du mot de passe, ainsi que de permettre à l'utilisateur de se déconnecter. Ma tutrice a restructuré le code pour qu'il ait la bonne syntaxe et que la vérification de mail se fasse bien avec des fichiers qu'elle avait déjà utilisés auparavant.

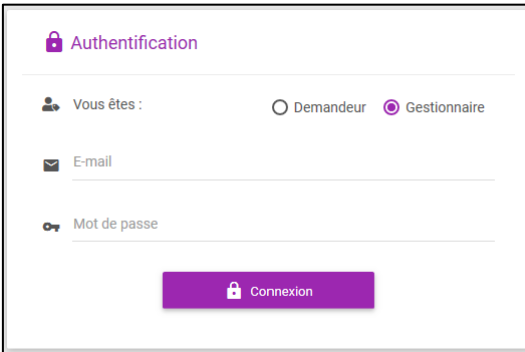
The image shows a web form titled "Authentification" with a purple lock icon. It contains three input fields: "Vous êtes :" with radio buttons for "Demandeur" and "Gestionnaire" (the latter is selected), "E-mail", and "Mot de passe" with a password icon. A purple "Connexion" button with a lock icon is at the bottom.

Figure 10 - Formulaire de connexion

### 3.2.8 Paramètres

Après cela, j'ai transmis mon code à ma tutrice comme à chaque fin d'étape et celle-ci m'a fourni une nouvelle version de l'ensemble du code avec quelques corrections et ajustements. Ensuite, j'ai dû m'occuper des paramétrages, dans lesquelles on peut ajouter, modifier ou supprimer des éléments dans les tables direction, DGA/ DGS, services, cadre, catégorie, etc. Pour ce contrôleur, il n'y a qu'une vue pour la modification et l'ajout de chaque table (donc le contrôleur est relié à une vue `p_frmModifGestionnaire.php`, `p_frmModifDGA.php`, etc.) en plus des vues de confirmation et des erreurs. Ce contrôleur avait la particularité d'avoir pour chaque cas (ajout, modification, suppression, liste), une condition supplémentaire qui permet de définir directement quel paramètre doit être traité sans utiliser de sous-contrôleurs pour chaque table. Ce choix est justifié par le fait que chaque table des paramètres utilise les mêmes actions (définies dans l'URL afin d'identifier quel contrôleur doit être exécuté et quel cas doit être traité).

```
1 <?php
2 //Récupération du nom de la table
3 $smPrm = isset($_MENU[$_uc]['SUB_MENU']) ? $_MENU[$_uc]['SUB_MENU'] : null;
4 $mnTb = isset($_smPrm[$_action]) ? $_smPrm[$_action] : null;
5 if(!$mnTb) { foreach($_smPrm as $sm) { if($_sm['SUFFIX'] && strpos($_action, strtolower($_sm['SUFFIX'])) $mnTb = $sm; } } //Pour récupérer les options si pas lst
6 $tblbl = isset($_mnTb['LABEL']) ? $_mnTb['LABEL'] : null;
7 $tbSfx = isset($_mnTb['SUFFIX']) ? $_mnTb['SUFFIX'] : null; //récupération du cas d'utilisation
8 $sfx = strtolower($tbSfx); //récupération du nom de la table
```

Figure 11 - Entête du contrôleur des paramètres

```

158 case 'add_'.$sfx:
159 {
160     //-----
161     // Variables
162     //-----
163
164     $id = null;
165     $msgErreurs = null;
166     $titre = $tblbl." : Création";
167     $mdlIcon = "add_circle";
168     $urlFrm = "index.php?uc=prm&action=add_".$sfx;
169
170     //-----
171     // Traitement
172     //-----
173
174     switch($tbSfx)
175     {
176         case 'Gestionnaire':
177         {
178             if(!$DROIT['PRM_Gestionnaire'])
179             {
180                 $msgErreurs[] = "Vous n'êtes pas autorisé à accéder à cette page!";
181                 include $_CONFIG['DIR_View']."p_msgErreurs.php";
182                 break;
183             }
184             // Valeurs saisies
185             $gstNom = http_GetRequest('gest_nom');
186             $gstPrenom = http_GetRequest('gest_prenom');
187             $gstLogin = http_GetRequest('gest_login');
188             $gstMail = http_GetRequest('gest_mail');
189             $gstIsAdmin = http_GetRequest('gest_is_admin');
190             $gstIsOff = 0;
191
192             // Insertion
193             if(isset($_POST['frm_submit'])) $msgErreurs = gestionnaire_Insert($id, $gstLogin, $gstNom, $gstPrenom, $gstMail, $gstIsAdmin, $gstIsOff);
194
195             break;
196         }
197         case 'DGA':
198         {
199             if(!$DROIT['PRM_DGA'])
200             {
201                 $msgErreurs[] = "Vous n'êtes pas autorisé à accéder à cette page!";
202                 include $_CONFIG['DIR_View']."p_msgErreurs.php";
203                 break;
204             }
205         }
206     }

```

Figure 12 - Extrait de code du contrôleur des paramètres

### 3.2.9 CEDR

J'ai ensuite développé le contrôleur `c_gererCEDR`, qui par défaut, affiche le planning des séances CEDR et qui permet également à un gestionnaire d'ajouter, de supprimer ou encore d'archiver une séance choisie. Ce contrôleur est accompagné de deux vues propres à celui-ci, une pour l'affichage de la liste des CEDR sous forme d'un tableau (avec la page `p_lstCEDR.php`) et une pour le formulaire d'ajout et de modification d'une CEDR (avec la page `p_frmModifCEDR.php`). Pour la confirmation ou l'affichage d'une erreur de chaque cas de tous les contrôleurs, les pages `p_msgErreurs.php` ou `p_msgConfirmation.php` sont alors appelées pour notifier l'utilisateur.



Figure 133 - Exemple d'un message de confirmation

### 3.2.10 Demandes et pièces-jointes

Enfin, j'ai débuté la programmation du contrôleur de la gestion des demandes, dont la particularité est d'inclure un formulaire complet qui s'adapte en fonction du type de demande sélectionnée par l'utilisateur. De plus ce formulaire est basé sur le « Form-Wizard » du template AdminBSB, qui contient un formulaire compartimenté en plusieurs onglets s'affichant au fur et à mesure que l'on remplit les divers champs.

Figure 144 – premier formulaire type "Form-Wizard" des demandes

Le développement de ce contrôleur m'a pris beaucoup plus de temps à développer que les autres car je n'avais pas réussi à intégrer correctement les ressources (plug-in) nécessaires au déploiement de ce formulaire et à son bon fonctionnement. J'ai donc dû, en plus de développer le contrôleur et les vues correspondantes, coder une page de JavaScript afin d'avoir les interactions que proposait initialement ce « Form-Wizard » (comme le changement d'onglet lorsque l'on appuie sur le bouton « suivant » et que les champs obligatoires sont remplis). Ce fichier JavaScript permet également de masquer certains champs qui ne correspondent pas au type de demande sélectionnée par le demandeur.

Après le développement du formulaire, j'ai commencé à coder un des cas propres à ce contrôleur, qui est la réaffectation, c'est-à-dire que seul le gestionnaire peut réaffecter une demande tant que celle-ci n'a pas le statut « annulée ». Pour cela, le gestionnaire doit appuyer

sur le bouton situé en haut à droite du tableau qui affiche les demandes et de sélectionner l'option « réaffectation ». Une fois l'option choisie, la liste réapparaît avec des cases à cocher pour sélectionner l'ensemble des demandes à réaffecter. Un message de confirmation apparaît, sinon ce sera un message d'erreur en cas d'échec.

Ce contrôleur permet aussi à l'utilisateur d'afficher sa demande une fois celle-ci complétée dans le formulaire. De plus, le demandeur peut ajouter un document via la fonctionnalité « upload ». Cette option ne pourra s'effectuer qu'à la suite de la saisie du formulaire ou lors de sa modification. Cette fonction utilise également un serveur ftp en plus d'une table dédiée pour stocker le document. Cependant, cette fonction n'a pas fonctionné durant les tests.

Enfin, j'ai fini le développement de ce contrôleur avec le cas d'annulation d'une demande. Pendant le développement de ces derniers cas, ma tutrice a récupéré le code du formulaire afin de le faire fonctionner avec les ressources initiales issues du template. Elle m'a également transmis une nouvelle version du code de l'ensemble du site et m'a confié la tâche de développer un moyen de convertir un tableau d'une liste de demandes du site en un tableau Excel selon une disposition précise, chose que la fonction intégrée du template ne peut pas faire.

31.

REPLACEMENTS TEMPORAIRES							
Service ou Direction	Intitulé du poste	Nbre	Cadre d'emploi	Remplacement de	Argumentaires Demandeur et DGA	Avis RH	Décision CEDR
CATEGORIE A							
CATEGORIE B							
CATEGORIE C							
Total							

Figure 15 - Disposition des données de demandes lors de l'extraction

D'autres moyens pour programmer cette extraction ont été envisagés, comme l'utilisation d'un plug-in nommé PHPExcel. Cependant, ce plug-in est obsolète et il aurait pour inconvénient d'alourdir la taille des ressources nécessaires pour le site. Ainsi, j'ai opté pour du JavaScript par rapport à cette fonction. D'autres solutions seront sûrement envisagées une fois l'ensemble terminé car l'actuelle ne permet pas d'avoir une mise en forme poussée pour l'instant (ajout de bordures, couleurs).

	A	B	C	D	E	F	G
1	CEDR Pleniére du 03/06/2019 n°32 - Recrutement sur postes permanents vacants						
2	Numero	Direction / Service	Intitule du poste	Cadre d'emplois	Type de demande	Argumentaire	Statut
3	CATEGORIE B						
4	P32.1	direction	intituleM	ADJOINTS TERRITORIAUX D'ANIMATION	AVEC CREATION	motivation	En cours de saisie
5	P32.2	direction	TEST2	ADJOINTS TECH.TERRITOR ETABLIS.ENSEIG.	AVEC CREATION	Motivation2	En cours de saisie
6	CATEGORIE C						
7	P32.8	direction	ezty	AGENTS SOCIAUX TERRITORIAUX	AVEC MODIFICATION	test	Avis des RH : Favorable
8							

Figure 15 - Tableau Excel généré par la fonctionnalité d'extraction

Ensuite, ma tutrice a vérifié mon contrôleur pour les demandes dans le but de voir comment j'avais traité le cas de la réaffectation et de l'extraction de données sous un tableau Excel. Elle m'a conseillé de réduire le nombre de pages pour la réaffectation afin de faciliter la procédure pour les utilisateurs futurs. En effet, la procédure, telle que je l'avais établi initialement, affichait d'abord une page pour sélectionner une ou plusieurs demandes puis chargeait une deuxième affichant un tableau contenant toutes les CEDR dans lesquelles on pouvait les réaffecter. Je savais que cette procédure était lourde, mais je pensais qu'il s'agissait d'un bon moyen d'aider l'utilisateur à effectuer une réaffectation.





Réaffectation d'une demande							
Afficher 10 éléments		Rechercher :					
<input type="checkbox"/> Tout sélectionner	Intitule	cedr id	Nom du demandeur	Prénom du demandeur	Mail du demandeur	Actions	
<input type="checkbox"/> Sélectionner	intituleM	32	NMDMR	pdmr	m.minbielle@outlook.fr	 	
<input type="checkbox"/> Sélectionner	TEST2	32	MINI	Momo	m.minbielle@outlook.fr	 	
Affichage de l'élément 1 à 2 sur 2 éléments						Précédent	Suivant
<input type="button" value="réaffecter"/>							

Figure 16 - Partie 1 de la réaffectation



Choix de la CEDR pour réaffectation des demandes

Afficher 10 éléments

Rechercher :

Sélectionner	Numéro	Date limite de dépôt	Date de la CEDR	Intitulé de la CEDR	Nombre de demandes	Statut
<input type="radio"/> Sélectionner	P1	10/04/2019	20/04/2019	N/A	0	Archivée
<input type="radio"/> Sélectionner	P13	13/02/2019	28/02/2019	Février PASSÉ	0	Archivée
<input type="radio"/> Sélectionner	P2	05/03/2019	01/04/2019	AUCUN	0	Archivée
<input type="radio"/> Sélectionner	P28	14/05/2019	17/07/2019	N/A	0	En cours d'analyse
<input type="radio"/> Sélectionner	P31	14/06/2019	24/06/2019	CEDR plénière juin 2019	1	Ouverte
<input type="radio"/> Sélectionner	P32	03/06/2019	03/06/2019	CEDR JJ	2	Clôturée
<input type="radio"/> Sélectionner	P33	13/08/2019	20/08/2019	Coucou	1	Ouverte
<input type="radio"/> Sélectionner	P34	10/09/2019	17/09/2019	HELLO IT'S ME	1	Ouverte
<input type="radio"/> Sélectionner	P6	01/05/2018	15/05/2018	CEDR Mai 2018 OK	0	Archivée
<input type="radio"/> Sélectionner	P7	05/05/2019	07/05/2019	N/A	0	Archivée

Affichage de l'élément 1 à 10 sur 29 éléments

Précédent 1 2 3 Suivant

réaffecter





Figure 17 - Partie 2 de la réaffectation

En suivant les conseils de ma tutrice et en prenant du recul, j'ai simplifié la procédure. Le cas de la réaffectation est désormais traité en une seule page dans laquelle l'utilisateur choisi ses demandes, puis sélectionne plus bas la CEDR dans laquelle il souhaite les réaffecter. Le choix de la CEDR se fait via une liste déroulante.

Réaffectation d'une demande

Show 10 entries

Search:

	Intitulé	cedr id	Nom du demandeur	Prénom du demandeur	Mail du demandeur	Actions
<input type="checkbox"/> Tout sélectionner						
<input type="checkbox"/> Sélectionner	intituleM	32	NOMDR	pdmr	m.minbielle@outlook.fr	 
<input type="checkbox"/> Sélectionner	TEST2	32	MINI	Momo	m.minbielle@outlook.fr	 

Showing 1 to 2 of 2 entries

Previous 1 Next

☐ Réaffecter à la CEDR: CEDR Mai 2018 OK - P6

Figure 19 - Nouvelle interface de réaffectation

Une fois cette correction apportée, le problème concernant l'ajout d'un document sur le serveur ftp devait être résolu pour s'assurer de son bon fonctionnement. Ma tutrice m'a donc indiqué qu'il fallait que j'intègre le dossier du site directement sur le serveur WebApp qui permettra aux utilisateurs d'accéder au site ultérieurement. J'ai donc dû utiliser le logiciel WinSCP pour accéder à ce serveur et tester le site.

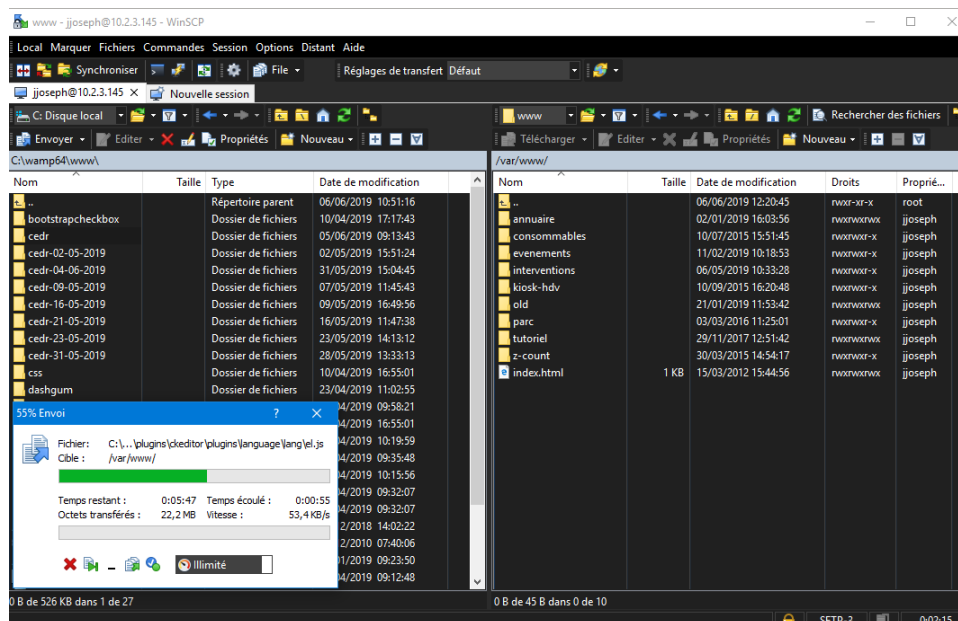


Figure 20 - Copie du dossier du site via WinSCP


Après avoir eu accès au serveur, il m’a suffi de copier mon dossier contenant tous les fichiers du site dans le dossier /var/www/ du serveur afin de pouvoir y accéder depuis le navigateur en saisissant l’URL “webapp/cedr/” qui lancera automatiquement la page dynamique “index.php” faisant office de routeur du site (c’est cette page qui, en fonction des cas, choisit le contrôleur et le cas qui doit être exécuté pour répondre à la requête émise).



Figure 21 - URL d’accès au site via WebApp

Quelques erreurs de syntaxe du contrôleur de gestion des demandes empêchaient l’affichage des demandes sur le navigateur. Pour identifier ces erreurs, j’ai donc dû utiliser l’outil PuTTY (qui permet d’accéder au serveur en ligne de commandes Shell) afin d’avoir les droits de consulter les logs (qui sont des rapports d’erreurs stockés dans un fichier). Une fois ce problème résolu, il fallait réexécuter la tentative d’upload d’un document pour vérifier que le problème n’était que local (mon poste n’ayant pas certains accès). La procédure a fonctionné car le document s’est bien retrouvé sur le serveur ftp et donc dans la base de données.

Index de ftp://cedr@10.2.3.127/CEDR/

 Vers un rép. de plus haut niveau

Nom	Taille	Dernière modification
Fichier: 1559573264_test.txt		03/06/2019 16:47:00
Fichier: 1559573385_test.txt		03/06/2019 16:49:00
Fichier: 1559632872_test.txt		04/06/2019 09:21:00
Fichier: 1559658676_test.txt		04/06/2019 16:31:00
Fichier: 1559658937_test.txt		04/06/2019 16:35:00
Fichier: 1559659315_test.txt		04/06/2019 16:41:00
Fichier: 1559659671_test.txt		04/06/2019 16:47:00
Fichier: 1559660832_test.txt		04/06/2019 17:07:00
Fichier: 1559661105_test.txt		04/06/2019 17:11:00
Fichier: 1559723843_test.txt		05/06/2019 10:37:00
Fichier: 1559738327_test.txt		05/06/2019 14:38:00
Fichier: 1559745213_test.txt		05/06/2019 16:33:00
Fichier: 1559745711_test.txt		05/06/2019 16:41:00
Fichier: 1559748555_test.txt		05/06/2019 17:29:00
Fichier: 1560243413_test.txt		11/06/2019 10:56:00
Fichier: 1560246181_test.txt		11/06/2019 11:43:00
Fichier: 1560262763_test.txt	1 KB	11/06/2019 16:19:00

Figure 22 - Résultat des transferts FTP

	doc_id	doc_dmd_id	doc_dt_upload	doc_titre	doc_fichier	doc_extension	doc_is_pp
▶	1	3	2019-06-11 16:19:11	test.txt	1560262763_test.txt	txt	0
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 23 - Résultat du transfert dans la table

J'ai également profité de cette intégration du site sur le serveur pour m'assurer que toutes les fonctions de ma version actuelle du site fonctionnent sans erreurs. Une fois l'ensemble du code vérifié, j'ai transmis ma version à ma tutrice pour qu'elle la version définitive du formulaire de demande. Après avoir reçu la version complète du site reçue, j'ai pu voir les différents changements effectués sur certains fichiers (notamment au niveau des fichiers JavaScript), ainsi que le code concernant ce nouveau formulaire. J'ai intégré au code mes dernières fonctionnalités dans le but d'avoir une vue d'ensemble concernant le rendu du site. De plus, j'ai essayé d'adapter mon contrôleur afin que la façon de coder les différents cas corresponde au mieux à la structure du nouveau code, comme à chaque récupération du code.

Après la fin de la refonte du formulaire de demande de ma tutrice et de son code pour la gestion des documents qu'un utilisateur peut ajouter, supprimer ou voir, j'ai récupéré l'ensemble de son code et fusionné celui-ci avec ma dernière version. Ma dernière version avait pour principale nouveauté d'intégrer une vue supplémentaire qui générerait le tableau HTML de la liste des demandes liées à une CEDR sélectionnée par l'utilisateur qui sera extrait en fichier Excel. Cette vue est la dernière solution trouvée afin d'effectuer l'extraction de données avec mise en forme. Après avoir effectué cette fusion et m'être assuré que l'ensemble fonctionnait correctement, j'ai donc une nouvelle fois intégré cette version du site sur le serveur WebApp.

	A	B	C	D	E	F	G	H
1	NumÃ©r	Direction / Service	IntitulÃ© du poste	CatÃ©gorie	Cadre d'emplois	Type de demande	Argumentaire	Statut
2		CABINET						
3		ACHATS						
4		ACCUEIL - ACHILLE DOMART						
5	P32.1		intitulÃ©M	CATÃ©GORIE B	ADJOINTS TERRITORIAUX D'ANIMATION	AVEC CRÃ©ATION	motivation	En cours de saisie
6		DBMT						
7		ATELIERS MECANIKES ET						
8		ACCUEIL - DOMART PESQUE						
9	P32.2		TEST2	CATÃ©GORIE B	ADJOINTS TECH.TERRITOR ETABLIS.ENSEIG.	AVEC CRÃ©ATION	Motivation2	RÃ©affectÃ©e

Figure 18 - Tableau Excel issu de l'extraction

Modification de la demande de recrutement permanent (P31.3)

Réaffectée - La demande de recrutement doit être obligatoirement accompagnée de la fiche de poste

1. Origine de la demande
2. Détails de la demande
3. Avancement / Annexes (3)

Pièces jointes à la demande

+ Ajouter un fichier

Date d'envoi	Nom du fichier	Type	Actions
Le 11/06/2019 à 16h19	test.txt	Annexe	
Le 14/06/2019 à 15h52	Chiots.jpg	Profil de poste	
Le 14/06/2019 à 16h16	test.txt	Annexe	

Avancement de la demande

Date de mise à jour	Statut	Commentaire
Le 31/05/2019 à 13h57	En cours de saisie	
Le 12/06/2019 à 09h55	Réaffectée	

Précédent
Enregistrer

Figure 19 - Nouveau formulaire

## CONCLUSION

Ce stage était concentré sur la problématique de la simplification des procédures de demandes de recrutement ou de remplacement par les différents services de la commune.

Le but de mon stage était donc d'établir un site pour lequel ces procédures s'effectueraient de manière plus transparente et efficace, à la fois par les directions générales et les employés qui font les demandes. Une fois le site déployé, les ressources humaines ou autres gestionnaires fourniraient un mot de passe pour donner un accès aux utilisateurs qui souhaitent effectuer une demande et s'occuperont de la validation des demandes et de l'édition des séances.

Le site, tel qu'il est aujourd'hui, n'est pas finalisé car, bien qu'il contienne l'ensemble des fonctionnalités prévues, il reste des portions de codes à uniformiser et à optimiser en fonction de la structure établie. De plus, certaines solutions comme l'extraction d'une liste de demandes d'une CEDR, ne permettent pas de remplir complètement leur fonction de manière complète. Néanmoins, le site étant fonctionnel et contenant toutes les fonctions essentielles à la procédure des demandes, il devrait être disponible sous peu pour avoir des retours des utilisateurs.

D'un point de vue personnel, ce projet de stage m'a permis de mettre en pratique les méthodes et connaissances de programmation pour le développement d'un site web. De plus, il m'a appris à travailler en équipe réduite depuis le début du projet (bien que le cahier des charges et la planification furent établis en amont), en ayant continuellement des modifications à apporter suite aux échanges avec les clients ou aux discussions sur la conception d'une fonctionnalité. J'ai également beaucoup appris en observant une nouvelle façon d'organiser la structure du code ou des méthodes appliquées dans cette collectivité afin de travailler en groupe. L'apprentissage autonome de la maîtrise de nouveaux outils informatiques et l'appui de ma tutrice furent également pour moi, une source d'épanouissement.

# GLOSSAIRE

**CEDR** : Commission d'Évaluation des Demandes de Recrutement.

**CSS** : Cascading Style Sheets (ou feuilles de style en cascade en français) est un langage qui définit la mise en forme d'une page web.

**Extranet** : il s'agit d'un mélange du réseau interne d'une organisation et d'Internet.

**Framework** : regroupe plusieurs outils constituant les bases d'un logiciel informatique.

**HTML** : l'HyperText Markup Language est un langage de balisage, créer pour structurer les pages web.

**Intranet** : réseau informatique privé, généralement utilisé au sein d'une organisation.

**Internet** : réseau informatique mondial.

**JavaScript** : il s'agit d'un langage de programmation au niveau client, qui permet aux pages web de devenir interactives grâce à l'exécution de scripts par le navigateur.

**MySQL** : il s'agit d'un système de gestion de base de données.

**PHP** : l'Hypertext Preprocessor est un langage utiliser dans le développement web afin de dynamiser des pages.

**PHPExcel** : il s'agit d'une librairie PHP permettant de faire un tableau Excel à partir de données d'une page.

**Plug-in** : c'est un module d'extension, qui complète un logiciel pour ajouter une fonctionnalité.

**PuTTY** : il s'agit d'un émulateur du terminal de commande présent sous l'environnement Linux.

**SCP (Secure Copy)** : transfert de fichiers entre deux machines via le protocole de communication SSH.

**Serveur FTP (File Transfer Protocol) :** serveur permettant l'envoi de fichier via un réseau informatique ou Internet.

**Shell :** interpréteur de commandes permettant à l'utilisateur d'accéder aux fonctionnalités internes de son système d'exploitation.

**SSH (Secure SHell) :** protocole de communication sécurisé.

**Template :** en développement Web, un template définit l'architecture du design d'une page.

**URL :** Uniform Ressource Locator ou localisateur uniforme de ressource désigne une adresse web qui permet d'identifier quelle ressource afficher.

**WebApp :** application en ligne avec laquelle l'utilisateur interagit directement via son navigateur sans avoir d'installations nécessaires sur son poste.

**WinSCP:** programme permettant le transfert de fichier sécurisé entre deux ordinateurs distants via le protocole SCP.

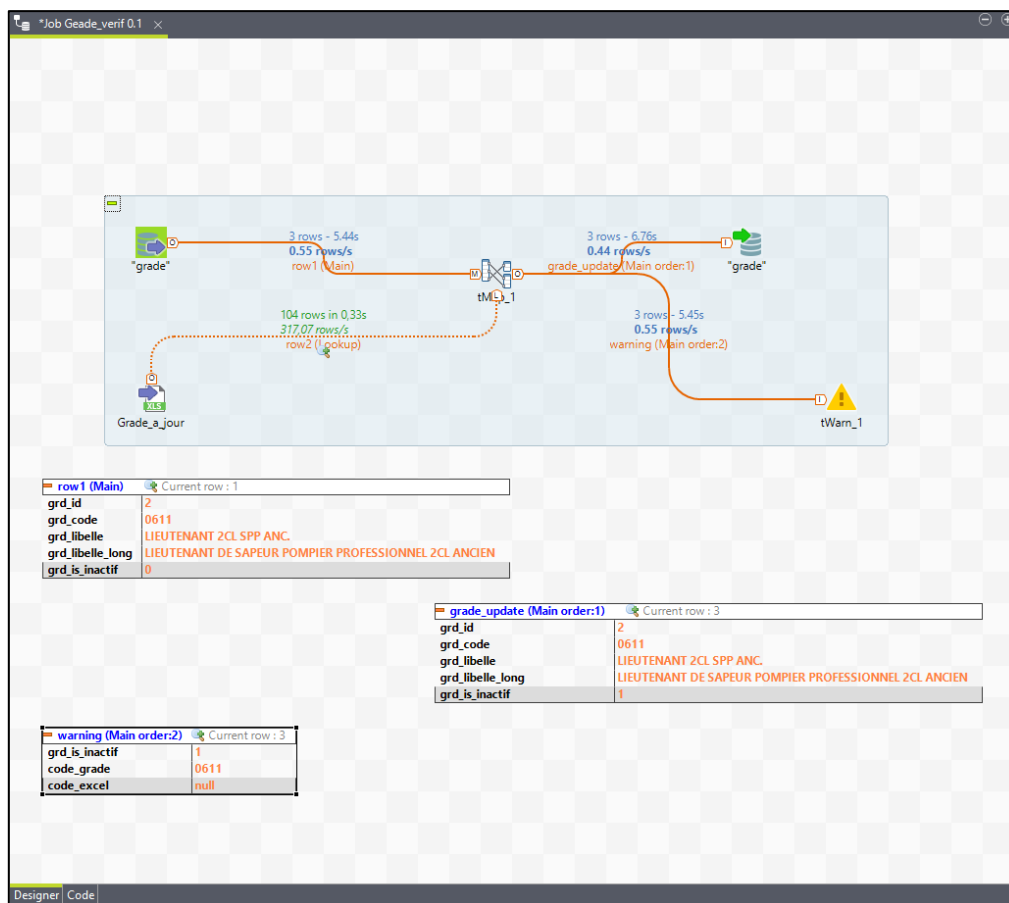
# ANNEXES

GRADES ACTUALISES AU 01/01/2019			
Code grade	Libellé moyen grade	Libellé long grade	trouver
0010	DIRECTEUR DE CABINET	DIRECTEUR DE CABINET	0
0013	ETUDIANT STAGIAIRE	ETUDIANT STAGIAIRE	0
0014	FORMATEUR	Formateur vacataire	0
1001	Administrateur HCI	Administrateur hors classe	0
1005	Administrateur	Administrateur	0
1009	Attaché HCI	Attaché Hors Classe	0
1010	Attaché	Attaché	0
1016	Directeur ter	Directeur territorial	0
1019	Attaché Pal	Attaché principal	0
1036	Rédacteur	Rédacteur	0
1037	Rédacteur Pal 2CI	Rédacteur Territorial Principal 2 CL	0
1038	Rédacteur Pal 1CL	Rédacteur Territorial Principal 1 CL	0
1054	Adjt adm	Adjoint administratif	0
1057	Adjt adm Pal 2CI	Adjoint administratif principal de 2ème classe	0
1058	Adjt adm Pal 1CI	Adjoint administratif principal de 1ère classe	0
1102	D.G. 80 à 150 mille hab.	Directeur Général des Services 80 à 150.000 habitants	0
1202	D.G.A.40 a 150 mille hab	Directeur General Adjoint des Services 40 a 150.000 habitants	0
2001	Technicien	Technicien territorial	0
2008	Technicien princ 2e clas	Technicien principal de 2e classe	0
2012	Technicien princ 1e clas	Technicien principal de 1ere classe	0
2200	Agent maitrise	Agent de maîtrise	0

Annexe 1 – Extrait d'un fichier Excel contenant tous les grades à utiliser pour le formulaire de demande







Annexe 4 - Extraction pas à pas pour vérifier l'intégration des données

	grd_id	grd_code	grd_libelle	grd_libelle_long	grd_is_inactif
▶	-1	ZZZZ	- SANS GRADE -	- POSTES SANS GRADE -	1
	1	5940	MONITEURS EDUCATEURS	MONITEURS EDUCATEURS	0
	2	0611	LIEUTENANT 2CL SPP ANC.	LIEUTENANT DE SAPEUR POMPIER PROFESSIO...	0
	3	0351	SOUS-ARCHIVISTE CHEF	SOUS-ARCHIVISTE CHEF	0
	4	0352	SOUS-ARCHIVISTE PRINCIP.	SOUS-ARCHIVISTE PRINCIPAL	0
	5	0185	OUVRIER ENT. VOIE PUB.EC	OUVRIER ENTRETIEN VOIE PUBLIQUE (EMPLOI ...	0
	6	1030	REDACTEUR TER.CHEF	REDACTEUR TERRITORIAL CHEF	0
	7	3603	ASSIST. CONSERV.HC (ANC)	ASSISTANT DE CONSERVATION HORS CLASSE ...	0
	8	0230	ORDON.POMPES FUNEBRES EC	ORDONNATEUR DE POMPES FUNEBRES EMPLOI...	0
	9	0226	GARDIEN CIMETIERE 1C(EC)	GARDIEN DE CIMETIERE 1ERE CATEGORIE EMP...	0
	10	0707	BRIGADIER (EC)	BRIGADIER EMPLOIS COMMUNAUX	0
	11	0272	CHEF EGOUTIER CAT.INSAL.	CHEF EGOUTIER CATEGORIE INSALUBRE	0
	12	8010	ADJOINT TER.D'ANIMATION	ADJOINT TERRITORIAL D'ANIMATION	0
	13	1020	SECRETAIRE DE MAIRIE	SECRETAIRE DE MAIRIE	0
	14	0162	RECEVEUR PLACIER	RECEVEUR PLACIER (EMPLOI COMMUNAL)	0
	15	1052	AGENT ADMINIS.TER.QUAL	AGENT ADMINISTRATIF TERRITORIAL QUALIFIE	0
	16	7023	CHEF SERVICE PM PRINC 1C	CHEF DE SERVICE DE POLICE MUNICIPALE PRI...	0
	17	3302	ATTACHE CONS.PATR. 1E CL.	ATTACHE TERRITORIAL DE CONSERVATION D...	0
	18	5302	PSYCHOLOGUE HORS CLASSE	PSYCHOLOGUE HORS CLASSE	0
	19	0548	GARCON LABO.ANALYSE CHIM	GARCON DE LABORATOIRE ANALYSES CHIMIQU...	0
	20	0701	GARDE CHAMPETRE (EC)	GARDE CHAMPETRE EMPLOI COMMUNAUX	0

Annexe 5 - Extrait du contenu de la table "Grade" suite à l'extraction de données

**Modification de la demande de recrutement permanent (P33.4)**  
**En cours de saisie** - La demande de recrutement doit être obligatoirement accompagnée de la fiche de poste

1. Origine de la demande    2. Détails de la demande    3. Avancement / Annexes (0)

**Organigramme**

**DGA :** \* DGA DEVELOPPEMENT (DGA DVPT)

**Direction :** \* ACTION SANITAIRE ET SOCIALE HABITAT LOGEMENT (DASSL)

**Service :** \* - AUTRE - (AUTRE)

**Si autre service, préciser :** \* AUTRE

**Secteur :** ACCUEIL - ACHILLE DOMART

**Demandeur**

**Identité :** \* NTEST ptest

**Fonction :** \* fct

**E-mail :** \* m.minbielle@outlook.fr

Précédent    Suivant

Annexe 6 - Onglet 1 du formulaire de demande final

**Modification de la demande de recrutement permanent (P33.4)**  
**En cours de saisie** - La demande de recrutement doit être obligatoirement accompagnée de la fiche de poste

1. Origine de la demande    2. Détails de la demande    3. Avancement / Annexes (0)

**Type de demande**

**Demande de recrutement :**

☐ AVEC CRÉATION (Le poste n'existe pas)  
☐ AVEC MODIFICATION (Le poste n'existe pas)  
☒ SANS CRÉATION (Le poste existe déjà, exemple: Remplacement d'un agent en retraite)

**Profil de poste concerné**

**Intitulé du poste :** Remplacement

**FINIRE :** \* CULTURELLE    **Catégorie :** \* A

**Cadre d'emploi :** ADJOINTS DU PATRIMOINE TERRITORIAUX

**Grade :** \* Choisir un grade

**Temps de travail :** ☒ Temps complet    ☐ Temps non complet    ☐ Vacataire / Horaire

**Nombre d'heures :** \* 35

**Missions, activités :** \*

**Remplacement**

**Identité de l'agent à remplacer :** NTEST ptest

**Motif de la vacance :** \* Disponible

**Date de vacance officielle du poste :** \*

**Date souhaitée de remplacement :** \*

**Motivations**

**Motiver votre demande :** \*

Précédent    Suivant

Annexe 7 - Onglet 2 du formulaire de demande final (les champs suivant le type de demandes deviennent obligatoires s'ils correspondent aux informations nécessaires)

**Modification de la demande de recrutement permanent (P31.3)**  
✖ Réaffectée - La demande de recrutement doit être obligatoirement accompagnée de la fiche de poste

1. Origine de la demande
2. Détails de la demande
3. Avancement / Annexes (3)

**📎 Pièces jointes à la demande**

[+ Ajouter un fichier](#)

Date d'envoi	Nom du fichier	Type	Actions
Le 11/06/2019 à 16h19	test.txt	Annexe	
Le 14/06/2019 à 15h52	Chiots.jpg	Profil de poste	
Le 14/06/2019 à 16h16	test.txt	Annexe	

**🔄 Avancement de la demande**

Date de mise à jour	Statut	Commentaire
Le 31/05/2019 à 13h57	<span style="color: blue;">✍ En cours de saisie</span>	
Le 12/06/2019 à 09h55	<span style="color: red;">✖ Réaffectée</span>	

[Précédent](#)
[Enregistrer](#)

Annexe 8 - Onglet 3 du formulaire de demande final

**Uploader un fichier**

**Fichier (Max. 10M) :** Parcourir... Aucun fichier sélectionné.

**Type profil de poste :** NON ☒ OUI

**⚠ Attention, pensez à enregistrer votre demande avant d'uploader votre fichier!**

[↩ Annuler](#)
[📤 Uploader le fichier](#)

Annexe 9 - Fenêtre d'upload de fichier qui apparaît lorsque l'on appuie sur "Ajouter in fichier" dans le dernière onglet du formulaire

```

1 <?php
2 //Menu utilisateur
3 $_NAV_USER = array(
4     "edit_account"      => array(
5         "LABEL"          => "Modifier mon compte",
6         "URL"             => "index.php?uc=auth&action=edit_account",
7         "ICON"            => "account_box",
8         "RULE_MENU"      => user_IsLogged() && (user_IsGest() || user_IsAdmin())
9     ),
10    "chg_pwd"           => array(
11        "LABEL"           => "Changer de mot de passe",
12        "URL"              => "index.php?uc=auth&action=chg_pwd",
13        "ICON"             => "vpn_key",
14        "RULE_MENU"       => user_IsLogged() && (user_IsGest() || user_IsAdmin())
15    ),
16    "logout"            => array(
17        "LABEL"            => "Déconnexion",
18        "URL"              => "index.php?uc=auth&action=logout",
19        "ICON"             => "power_settings_new",
20        "RULE_MENU"       => user_IsLogged()
21    )
22 );

```

Annexe 10 - Extrait de code de menu.inc.php avec le tableau navbar qui sera utiliser pour les options du compte sur la barre de navigation verticale gauche du menu de la page affichée

```

1 <section>
2
3 <!-- Left Sidebar -->
4 <aside id="leftsidebar" class="sidebar">
5
6 <?php if(user_IsLogged() && $action != "deconnexion") { ?>
7 <!-- User Info -->
8 <div class="user-info" style="background-color: grey;">
9 <div class="image">
10 
11 <span> Profil <?=$SESSION['profil'] == 1 ? "Gestionnaire" : "Demandeur"; ?> </span>
12 </div>
13 <div class="info-container">
14 <div class="name" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"><?=$SESSION['nomcomplet']; ?></div>
15 <div class="email"><?=$SESSION['email']; ?></div>
16 <div class="btn-group user-helper-dropdown">
17 <i class="material-icons" data-toggle="dropdown" aria-haspopup="true" aria-expanded="true">keyboard_arrow_down</i>
18 <ul class="dropdown-menu pull-right" style="min-width: 230px;">
19 <?php
20 //Menu utilisateur
21 foreach($_NAV_USER as $unNav)
22 {
23     if($unNav['RULE_MENU'])
24     {
25         echo "<li>\n";
26         echo "<a href=\"". $unNav['URL']. "\">\n";
27         echo "<i class=\"material-icons\">". $unNav['ICON']. "</i>\n";
28         echo $unNav['LABEL']. "\n";
29         echo "</a>\n";
30         echo "</li>\n";
31     }
32 }
33 ?>
34 </ul>
35 </div>
36 </div>

```

Annexe 11 - affichage des options de compte en fonction de menu.inc.php

```

26 $_CONTENT = function($uc, $action)
27 {
28     //Appel de la variable globale
29     global $_CONFIG;
30     global $_DROIT;
31     global $_MENU;
32
33     //Redirection vers les controleurs
34     switch($uc)
35     {
36         case 'auth':
37         {
38             $action = http_GetRequest('action', "login");
39             include $_CONFIG['DIR_Control']."c_gererCompte.php";
40             break;
41         }
42         case 'cedr':
43         {
44             if(user_IsLogged()) include $_CONFIG['DIR_Control']."c_gererCEDR.php";
45             else include $_CONFIG['DIR_View']."i_retourConnexion.php";
46             break;
47         }
48         case 'dmd':
49         {
50             if(user_IsLogged()) include $_CONFIG['DIR_Control']."c_gererDmd.php";
51             else include $_CONFIG['DIR_View']."i_retourConnexion.php";
52             break;
53         }
54         case 'doc':
55         {
56             if(user_IsLogged()) include $_CONFIG['DIR_Control']."c_gererDoc.php";
57             else include $_CONFIG['DIR_View']."i_retourConnexion.php";
58             break;
59         }
60         case 'prm':
61         {
62             if(user_IsLogged()) include $_CONFIG['DIR_Control']."c_prmTable.php";
63             else include $_CONFIG['DIR_View']."i_retourConnexion.php";
64             break;
65         }
66         case 'help':
67         {
68             $action = http_GetRequest('action', "contact");
69             include $_CONFIG['DIR_Control']."c_voirAide.php";
70             break;
71         }
72         default:
73         {
74             $msgErreurs[] = "Cas d'utilisation inconnu!";
75             include $_CONFIG['DIR_View']."p_msgErreurs.php";
76             break;

```

Annexe 12 - Extrait du code du routeur qui redirige l'utilisateur vers le contrôleur puis la vue adaptée à sa requête