

Licence professionnelle en alternance

Système Informatique et Logiciels – DASI

Développeur et Administrateur de Systèmes d'Information

2015-2016

**PROJET TUTEURE**

GARCIA Nicolas

Développement d'une application de gestion des demandes de droits informatiques



Centre Hospitalier William Morey, DSIN

4 Rue Capitaine Drillien, 71321 Chalon-sur-Saône

Maître d'apprentissage : CLERC Sylvain

Tuteur enseignant : CORRIGAN Anne



## Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon projet et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, j'adresse mes remerciements à la DSI du Centre Hospitalier William Morey de m'avoir accueilli en tant qu'apprenti au sein du service.

Je tiens à remercier vivement mon maître d'apprentissage, Sylvain Clerc pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien.

Je remercie également Nicolas Limoges pour la confiance qu'il m'a accordé tout au long de ce projet.

Enfin, je tiens à remercier ma tutrice, Anne Corrigan pour m'avoir accompagné lors de ce projet.

## Contenu

Remerciements.....	1
1. Introduction.....	3
1.1 Présentation du Centre Hospitalier William Morey.....	3
1.2 Présentation des directions gérant l'informatique.....	4
1.2.1 La Direction des Technologies de Santé et du Numérique.....	4
1.2.2 La Direction du Système d'Information et du Numérique.....	4
1.2.3 Mes missions.....	4
2. Le contexte.....	6
3. Le Projet.....	7
3.1 Environnement technique.....	7
3.2 Les objectifs.....	8
3.3 Réalisation du projet.....	9
3.3.1 Le formulaire.....	9
3.3.2 La gestion.....	10
3.3.3 Quelques points technique.....	15
4. La suite du projet.....	17
5. Bilan.....	18
6. Annexes.....	19

## 1. Introduction

Ce projet tuteuré s'inscrit dans le cadre d'une licence professionnel Développeur d'applications d'entreprise et Administrateur de Systèmes d'Information (DASI).

J'ai effectué cette formation en alternance au Centre Hospitalier William Morey à Chalon sur Saône.

### 1.1 Présentation du Centre Hospitalier William Morey



Depuis 2011, le Centre Hospitalier William Morey remplace l'ancien hôpital St-Laurent à Chalon sur Saône.

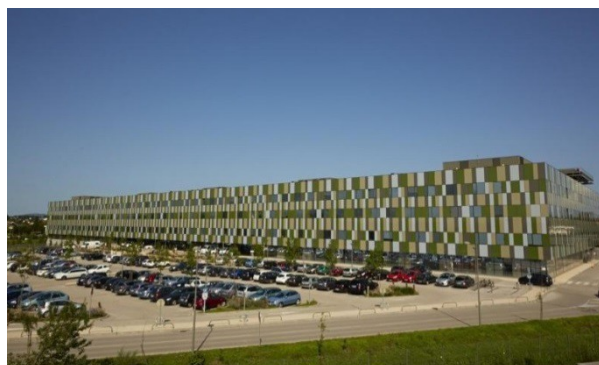
Le Centre Hospitalier William Morey est l'hôpital pivot du territoire Nord Saône et Loire. Il est réparti sur 4 sites : L'hôpital, 2 EHPAD à Chalon et St-Remy et l'IFSI - IFAS (Institut de Formation des Infirmiers et Aides-Soignants).

Le Centre Hospitalier Chalon sur Saône William Morey, c'est:

- 421 lits et places en Médecine, Chirurgie et Obstétrique
- 42 lits de convalescences à orientation gériatrique
- 270 lits d'Hébergement pour Personnes Âgées Dépendantes
- 1946 personnels hospitaliers, dont 260 personnels médicaux

Il dispose également d'un héliSmur, de 9 salles de bloc opératoire, d'un service d'imagerie médicale et d'un niveau d'informatisation très évolué.

Ses missions principales consistent à assurer un accès égal à tous à une offre de soin optimal ainsi qu'une amélioration constante de la prise en charge des patients aussi bien au niveau de la qualité que de la sécurité des soins délivrée au patient.



## 1.2 Présentation des directions gérant l'informatique

### 1.2.1 La Direction des Technologies de Santé et du Numérique

La DSTN (Direction des Technologies de Santé et du Numérique) dirigé par Patrick BASSET se compose de deux services : La DSIN (Direction du Système d'Information et du Numérique) qui gère le système d'information et la Direction de l'Ingénierie Biomédicale qui gère les équipements médicaux.

Ces deux services sont liés du fait que les équipements médicaux utilisent de plus en plus des équipements informatiques (Ordinateurs, imprimante,...).

### 1.2.2 La Direction du Système d'Information et du Numérique

J'ai donc travaillé à la DSIN qui a pour mission principale d'assurer le fonctionnement du système d'informations 7j/7 H24 (Système, réseaux, applicatif, interface et poste de travail).

Tous les services de l'hôpital utilisent le système d'informations, de ce fait, il a un impact direct sur le personnel qui l'utilise ainsi que sur les patients (rapidité de prise en charge, sécurité des données personnel,...).

Le service se compose de 16 personnes et est dirigé par Nicolas Limoge.

### 1.2.3 Mes missions

Mon travail au sein du service consiste en deux missions :

Ma mission principale est de faire évoluer l'application GiDD (Gestion Informatisée des Demandes de Droits) développée en premier lieu pour le CH de Chalon afin de l'adapter aux différents CH du groupement.

Ma mission secondaire est d'assurer le support utilisateur, en général deux jours par semaine. Le support utilisateur s'organise de cette façon :

- Le réceptionniste du service reçoit les appels des utilisateurs et crée un ticket (Demande d'Intervention). L'utilisateur peut également créer un ticket depuis un formulaire appelé GMAO.

**GMAO - Informatique**

Site:

Service:

Contact:

Téléphone:

Fonction:

Equipement:

Si vous saisissez cette demande pour une autre personne, précisez le dans la description.

Debuter par UC, CO, EC, PR, MF ou ETIQ.

Votre demande concerne un patient : ☐ Oui ☐ Non

Description:

Pour un patient, utilisez l'aide à la saisie afin de renseigner les détails utiles au traitement de votre demande.

Fichier:  Aucun fichier choisi

\* : champs obligatoires.

- En salle de Hotline, deux techniciens reçoivent les tickets sur l'application GLPI et les traitent. Une troisième personne s'occupe des tickets applicatifs (helpdesk, demandes de droits,...). Quand c'est possible, les interventions se font par téléphone et en prenant la main sur le poste de l'utilisateur.

Matériel	Description	Statut
83499	Général	MC
83500	Ordinateur UC1850	MC
83497	Général	GP
83495	Général	GP
83445	Ordinateur UC2764	MC
83431	Général	MC
82788	Général	GP
82993	Général	GP
83209	Général	GP

Matériel	Description	Statut
83459	Général	MC
83436	Général	GP
83404	Général	GP
83431	Général	GP
83472	Général	GP

Matériel	Description	Statut
71455	Général	GP
82316	Général	GP

Cette mission m'a permis d'apprendre beaucoup de chose sur des domaines très variés : Active directory, Administration Windows/Linux, Gestion de téléphone Wifi, Paramétrage de switch, utilisation de master, installation et maintenance d'imprimantes, de copieurs et d'imprimantes à étiquettes.

## 2. Le contexte

Afin d'améliorer la sécurité dans l'établissement, il est impératif de maîtriser au mieux les droits d'accès des agents, que ce soit un accès physique ou informatique.

L'objectif premier de GiDD (Gestion informatisée des Demandes de Droits) est de gérer les droits d'accès de manière globale tout en gardant une trace de ce qui a été fait pour chaque agent et ce, de son arrivée jusqu'à son départ de l'établissement.

Une première version de GiDD a été développée pour le CH William Morey. Mais cette version ne peut pas être adaptée sur les différents sites. L'objectif est donc de reprendre l'application de base et de l'améliorer pour qu'elle puisse s'adapter aux différents sites.

Cette nouvelle version comporte également de nouvelles fonctionnalités, une refonte graphique ainsi que l'utilisation de programmation orientée objet afin de faciliter la maintenance de l'application.

L'application est composée de deux grandes parties :

Le formulaire qui est rempli par la direction des ressources humaines lors de l'arrivée d'un nouvel agent. Au moment de la validation du formulaire, un mail contenant les informations de l'agent est envoyé aux différents gérants afin que ceux-ci créent les droits.

Une fois les droits créés, les gérants accèdent à la partie gestion de GiDD afin de valider la création des droits.

La gestion permet également de suivre l'avancement d'une demande de droits, de valider et supprimer les droits et de définir les applications, les gérants, les responsables et les fonctions.

Quand une demande est validée, un mail contenant les identifiants de connexion est envoyé au cadre du nouvel agent afin qu'il lui transmette.

## 3. Le Projet

### 3.1 Environnement technique



GiDD est une application web développée principalement en PHP et JavaScript. Elle utilise de l'AJAX qui permet des chargements asynchrone et donc de ne pas recharger la page à chaque action.

Afin de simplifier l'utilisation du DOM et d'AJAX, j'ai ajouté le framework jQuery qui permet d'avoir un code plus simple, plus lisible et qui gère automatiquement les problèmes de compatibilité entre les différents navigateurs.

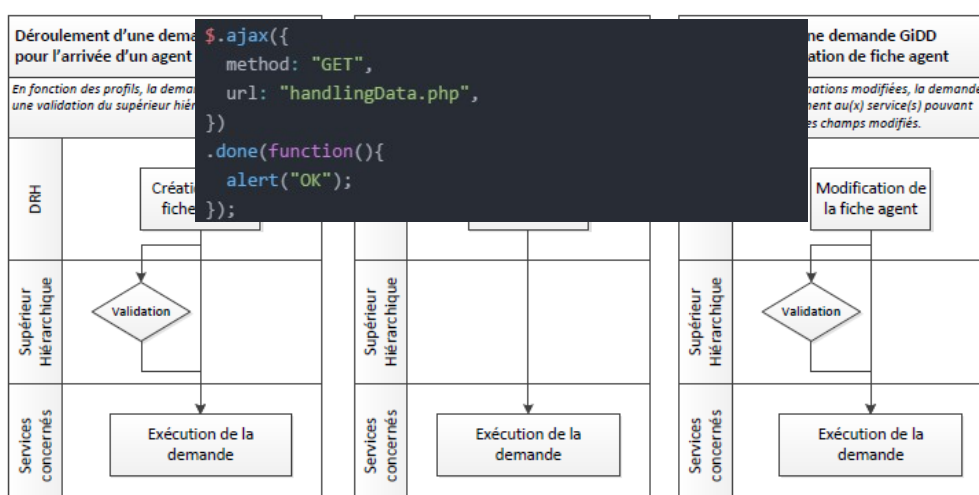
Par exemple une requête Ajax en JavaScript :

```
var xhr = getXMLHttpRequest();

xhr.onreadystatechange = function() {
    if (xhr.readyState == 4 && (xhr.status == 200 || xhr.status == 0)) {
        alert("OK"); // C'est bon \o/
    }
};

xhr.open("GET", "handlingData.php", true);
xhr.send(null);
```

Deviens ceci en jQuery :





La syntaxe jQuery est plus simple aussi bien à lire et à écrire.



Par rapport à l'ancienne version, Mysql a été remplacé par Mysqli, le premier étant obsolète.



Cotés logiciel, j'utilise Atom comme éditeur de texte car très personnalisable. De plus c'est un éditeur de texte récent et souvent mis à jour.



Spark est un logiciel de messagerie instantanée, il me permet de contacter mon maître d'apprentissage quand il travaille sur un autre site.



Enfin, j'ai choisis d'utilisé WinSCP comme client FTP.

### 3.2 Les objectifs

La première version de GiDD a été développée par une seule personne et en peu de temps. Ce qui explique l'absence de commentaire, de fichier css, de POO, de gestionnaire de version, etc...

L'objectif de cette version est de corriger ces différents problèmes afin d'obtenir une application plus facile à maintenir.

Une refonte graphique est également nécessaire afin d'améliorer l'ergonomie de l'application

Egalement l'utilisation de fichier de configuration qui permet de déployer l'application sur différents site.

De plus, l'ajout de nouvelles fonctionnalités ont nécessité des modifications dans la base données, ce qui demande une reprise du code en profondeur. (Voir schéma en annexe).

### 3.3 Réalisation du projet

Je suis le seul développeur sur ce projet, accompagné par Sylvain Clerc qui supervise l'avancée de l'application lors de réunion hebdomadaire. Sur une semaine classique, je travaille trois jours sur ce projet et deux jours en support utilisateur.

Le projet s'est déroulé en deux parties, j'ai commencé par la partie formulaire (de septembre à janvier) et ensuite la partie Gestion (de janvier à fin mai).

#### 3.3.1 Le formulaire

The screenshot displays the 'Fiches Agent' form in the GiDD application. The form is organized into two main columns. The left column contains input fields for various agent details: Matricule (20141113), Badge (badg21), Nom usuel (CLERC), Prénom (MARTIN), Grade (585D - TSH 2eme CLASSE), Fonction (Technicien Supérieur Hospitalier), Motif, Date (Arrivée: 23/10/1975, Départ: ), Service (UF) (5452 - CMP CATTP PIVOT REMPART ST PIE [...]), Arrivée (23/10/1975, Départ: ), 1104 - ACTIVITE CONVENTIONNEE, Arrivée (23/10/1975, Départ: ), Statut (TEST), ADELI (99), RPPS (9988), and Observation (obs 21 lignes). The right column features a vertical stack of buttons: Enregistrer, Dossier, Relance, Historique, and Retour. To the right of these buttons is a 'MODE DEBUG' section with a large empty text area. At the bottom left, a small note indicates '\* champs obligatoires'.

J'ai commencé par cette partie de l'application car elle avait besoin principalement d'une refonte graphique et d'une amélioration de l'ergonomie. Les nouveautés ici sont :

- La possibilité de saisir plusieurs services pour un agent
- Le bouton relance qui permet de renvoyer un mail aux personnes qui n'auraient pas validé tous les droits.
- La partie Historique a été refaite pour un affichage plus clair
- Le bouton Dossier qui permet d'extraire les informations d'un agent sur un fichier PDF

Le plus difficile sur cette partie a été de gérer les différentes erreurs qui pourraient être produite par l'utilisateur comme des parties obligatoires non remplies, des dates incohérentes, etc... Il a donc fallut vérifier que chaque champ contient des données cohérentes.

Les champs obligatoires du formulaire peuvent varier d'un site à l'autre, cette liste est donc gérée par un fichier de configuration.

Si une demande de droits n'a pas été validée au bout de sept jours, un mail de relance est envoyé aux responsables qui doivent les valider. Pour cela j'ai utilisé crontab, le système de planification de tâche de linux, qui permet d'exécuter des scripts PHP à la fréquence choisie.



Pour générer le rapport en fichier PDF, j'ai utilisé le framework FPDF que je ne connaissais pas et qui permet de créer des fichiers PDF en PHP.

**CENTRE HOSPITALIER**  
CHARLES SUD-SUD

**Rapport de demandes de droits**

Agent: **Holmes SHERLOCK**  
Matricule: 256  
Grade: 0422 - AAH  
Fonction: Adjoint des cadres  
Statut: 15  
ADELI: 15  
RPPS: 15

Demands concernant l'agent :

Application: **Windows**  
Etat: **En cours**  
Début droits: 15/12/2017  
Fin droits: pas de date de fin  
Gérant: DSI  
Observation: 15

### 3.3.2 La gestion

La gestion se découpe en plusieurs parties (Profils, Gérants, Responsables, Fonctions, Applications et Demande).

#### 3.3.2.1 Profils, Gérants, Responsables, Fonctions

Les parties profils, gérants, responsables et fonctions ont subi les mêmes modifications.

Chaque partie était composée de trois écrans (affichage, ajout, modification). L'interface a été revue de façon à limiter chaque partie à un seul écran. L'application est ainsi plus facile à utiliser et plus intuitive.

Selon la fonction d'un agent, un profil lui sera lié qui permettra d'établir la liste des applications dont il a besoin. Un profil permet donc de faire le lien entre une liste fonctions et une liste d'applications.

The screenshot shows the 'Profil' management interface in the GIDD application. The top navigation bar includes 'Profil - Gérants - Responsables - Fonctions - Applications - Demandes - Demandes en attente de validation'. The main interface is divided into several sections:

- Left sidebar:** A list of functions including 'Admission', 'Aide soignant', 'AMP - Animateur', 'ASH', 'Assistante sociale', 'Cadre de Santé', 'Diététicien', 'Directeur', 'Infirmier', 'Minimum', 'Praticien', 'Préparateur Pharmacie', 'Psychologue', 'Secrétaire médicale', and 'TIM'.
- Central form:** Fields for 'Intitulé' (set to 'Aide soignant') and 'Description'. Below these is a 'Liste des fonctions' section with checkboxes for 'Aide soignant', 'AMP', and 'Elève aide-soignant'.
- Right panel:** A 'Liste des applications' section with checkboxes for 'DFAG' (Test, Test enfant), 'DRH' (Badge), and 'DSI' (Windows).
- Buttons:** 'Valider', 'Supprimer', and 'Retour' buttons are located on the right side.

La partie profil était intéressante car un profil fait référence à une liste de fonctions et une liste d'applications. Il a donc fallu gérer ces deux listes en AJAX, la difficulté a donc été de transformer une liste JavaScript en une liste PHP.

Pour cela j'ai converti la liste Javascript en une chaîne de caractère contenant du JSON pour simplifier le passage d'un langage à l'autre.

Chaque application dispose d'un gérant, c'est-à-dire le service qui crée, modifie et valide les droits sur cette application. Chaque gérant est lié à une liste personnes. Un profil correspond à une liste de fonctions et une liste d'applications.

The screenshot shows the 'Gérants' management interface in the GIDD application. The top navigation bar includes 'Profil - Gérants - Responsables - Fonctions - Applications - Demandes - Demandes en attente de validation'. The main interface is divided into several sections:

- Left sidebar:** A list of managers including 'DFAG', 'DRH', 'DSI', 'Qualité', and 'Tutelles'.
- Central form:** Fields for 'Intitulé' (set to 'DSI') and 'Liste des gérants:'. Below these is a list of managers with checkboxes for 'Administrateur', 'CLERC Sylvain (DSI)', 'ABDELLAH Linda', 'AGOSTO Marie-Laure', 'AKNIN Morgane', 'ALIX Stephane', 'ALLARD-GRYSKA Marie-Christine', 'AMIENS Jean Bernard', and 'ANDRE Anne-Sophie'.
- Right panel:** A 'Liste des applications' section with checkboxes for 'DFAG' (Test, Test enfant), 'DRH' (Badge), and 'DSI' (Windows).
- Buttons:** 'Valider', 'Supprimer', and 'Retour' buttons are located on the right side.

La partie responsable permet, pour chaque service, de définir un responsable. C'est la personne qui recevra les informations de connexion de l'agent, afin de lui transmettre.

The screenshot shows the 'Responsables' (Managers) management interface. At the top, a navigation bar includes 'Profils - Gérants - Responsables - Fonctions - Applications - Demandes - Demandes en attente de validation'. The main area is divided into two columns. The left column, titled 'Responsables', contains a list of roles with their associated manager: '8200 - ACCUEIL COORDINATION DES HOSPITALISATION' (CLERC Sylvain (DSI) (clercsy)), '1104 - ACTIVITE CONVENTIONNEE' (CLERC Sylvain (DSI) (clercsy)), '4820 - ACTIVITES DE SECT INDIF POLE D' (CLERC Sylvain (DSI) (clercsy)), '6006 - ACTIVITES SPORTIVES' (CLERC Sylvain (DSI) (clercsy)), '1130 - AFFAIRES FINANCIERES', '5481 - AIGUE MARINE H.C.', '5482 - AIGUE MARINE H.J.', '5483 - AIGUE MARINE H.N.', '5480 - AIGUE MARINE INDIFFERENCIE', and '1510 - AMICALE DU PERSONNEL'. The right column contains a search bar, a 'Service' dropdown set to '8200 - ACCUEIL COORDINATION DES HOSPITALISATION', a 'Responsable' dropdown set to 'CLERC Sylvain (DSI)', and three buttons: 'Valider', 'Supprimer', and 'Retour'.

Enfin, la fonction de l'agent est liée à un profil. Plusieurs fonctions peuvent faire référence au même profil. Une fonction peut être soumise à validation par un gérant.

The screenshot shows the 'Fonctions' (Functions) management interface. At the top, a navigation bar includes 'Profils - Gérants - Responsables - Fonctions - Applications - Demandes - Demandes en attente de validation'. The main area is divided into two columns. The left column, titled 'Fonctions', contains a list of roles: 'Adjoint des cadres', 'Admission', 'AE', 'Agent Chef', 'Agent de maitrise', 'Agent SMD', 'Aide de pharmacie', 'Aide soignant', 'AMP', 'Animateur', 'Apprenti', 'ASH', 'Assistant Sociale', 'Attaché d'administration', 'Aumonier', 'Cadre de Santé', 'Cadre Supérieur de Santé', and 'CAE'. The right column contains a search bar, an 'Intitulé' dropdown set to 'Adjoint des cadres', a 'Fonction soumise à validation' checkbox checked, a 'DRH' dropdown, and three buttons: 'Valider', 'Supprimer', and 'Retour'.

Le code PHP permettant la visualisation, l'ajout, la modification et la suppression dans la base de données a été déplacé dans des classe PHP. L'utilisation d'objet permet d'alléger le code principal et de simplifier la maintenance de l'application

Le code CSS a été déplacé vers un fichier à part afin d'alléger le code principal et de rendre le code HTML plus lisible.

### 3.3.2.2 Applications

La partie applications est la plus difficile du projet car c'est celle qui a nécessité le plus de modifications. Les changements effectués sur la base de données pour les fonctionnalités de la gestion des applications a rendu le code de l'ancienne version inutilisable, pour cette partie je suis donc partie de zéro.

Les applications contiennent beaucoup d'informations qui sont réparties sur six tables différentes. Pour une meilleure lisibilité cette partie est séparée en différentes sections :

D'abord les informations générales de l'application : intitulé, description et gestion. La gestion peut prendre trois valeurs (Externe, Interne, CSA). La gestion interne est la seule qui permet de choisir si une confirmation par email est nécessaire.

La gestion CSA est arrivée en cours de projet, elle va permettre de créer certains droits automatiquement mais cette fonctionnalité ne fait pas parti de mon projet.

Chaque application dispose d'un gérant (Un groupe de personnes qui va créer les droits) mais peut aussi dépendre d'une autre application. De plus, d'autres applications peuvent dépendre de celle-ci. Pour le « traitement », j'ai créé une barre de recherche qui permet d'ajouter des applications dépendante et de les paramétrer sans avoir à changer ou recharger la page.

Une application peut disposer de différents fichiers, certains sont destinés aux gérants (Pour la création et la gestion des droits) et d'autres à l'utilisateur (Manuel d'utilisation, etc...). De plus, le chemin vers le fichier peut être local ou externe (URL).

Les champs utilisateur contiennent la liste des champs du formulaire. Pour chaque application, on précise quels champs sont obligatoires, facultatifs ou non utilisés par l'application. J'ai choisi de gérer cette partie de façon graphique afin de simplifier la lecture et la modification de ces champs.

Champs utilisateur			
Intitulé	Obligatoire	Facultatif	Non utilisé
Matricule			
Badge			
Prénom			
Nom			
Grade			
Fonction			
Date d'arrivée			

Les champs gérants sont des champs ajoutés par les gérants et qui sont nécessaire à l'application. Un champ peut être un texte, un texte long, une case à cocher ou une liste.

Dans le cas où c'est une liste, elle est reliée à une liste de contenu et un bouton radio permet de sélectionner le champ affiché par défaut.

### 3.3.3 Quelques points technique

Pour le champ gestion, j'avais besoin d'effectuer un traitement en fonction de la valeur choisie.

```
$(document).on("change","#select_gestion",function(){
    var select = $("#select_gestion");
    if(select.val() == "interne"){
        $("#chk_email").attr("disabled",false);
    } else if(select.val() == "externe"){
        $("#chk_email").prop("checked",true);
        $("#chk_email").attr("disabled",true);
    } else {
        $("#chk_email").attr("disabled",false);
        $("#chk_email").prop("checked",false);
        $("#chk_email").attr("disabled",true);
    }
});
```

J'ai donc créé un listener jQuery qui, à chaque changement sur la liste va activer ou désactiver la checkbox selon le choix effectué.

Sur cette partie j'ai également utilisé des lambdas que j'ai appris en formation lors d'un TP Java et qui m'as permis de gagner du temps.

```
if(Dependances.filter(key => key == uid).length > 0)
{
    alert("Cette dépendance existe déjà!");
}
```

Ici, j'utilise la fonction filter qui me permet de vérifier si une dépendance fait déjà parti de la liste. L'utilisation du lambda permet de réaliser cela simplement et en une seule ligne.

La partie champs utilisateurs a été plus facile à gérer que ce que j'imaginais. JQuery m'a permis de gérer cette partie assez facilement à l'aide de classe CSS.



```
var tableChamps = {};  
function selectChamps(obligatoire,uid)  
{  
  $("#champs_"+obligatoire+"_"+uid).addClass("td_champs_active");  
  $("#champs_"+obligatoire+"_"+uid).removeClass("td_champs_hover");  
  
  if(obligatoire != 'o')  
  {  
    $("#champs_o_"+uid).addClass("td_champs_hover");  
    $("#champs_o_"+uid).removeClass("td_champs_active");  
  }  
  
  if(obligatoire != 'f')  
  {  
    $("#champs_f_"+uid).addClass("td_champs_hover");  
    $("#champs_f_"+uid).removeClass("td_champs_active");  
  }  
  
  if(obligatoire != 'n')  
  {  
    $("#champs_n_"+uid).addClass("td_champs_hover");  
    $("#champs_n_"+uid).removeClass("td_champs_active");  
  }  
  
  tableChamps[uid] = obligatoire;  
}
```

Ici, je mets la case choisie en couleur et je mets les autres en grise.

La partie fichier a été difficile du fait que je n'avais jamais utilisé de formulaire qui enregistre des documents. Il a donc fallu que je fasse quelques recherches pour comprendre le fonctionnement.

## 4. La suite du projet

La première phase consistait à redévelopper l'application, maintenant qu'elle est fonctionnelle, la suite va être de déployer l'application. De nombreux tests vont être réalisés pendant le mois de juin sur le site de Sevrey.

La gestion des droits diffère beaucoup d'un cas à l'autre, selon l'agent et le type d'applications qu'il utilise. Cette période de test aura donc pour objectif de tester le plus de cas possibles afin d'éliminer les bugs et autres soucis rencontrés. Certaines demandes de droits sont gérées en plusieurs étapes et parfois sur plusieurs jours, ce qui va fortement compliquer ces tests.

Cette période permettra également d'avoir un retour des utilisateurs ce qui permettra de s'assurer que l'application correspond bien aux besoins initiaux.

Une fois cette étape de tests terminée, l'application sera déployée sur les autres sites du territoire ce qui permettra de vérifier que les fichiers de configuration permettent de la déplacer.

Cette phase est importante car GiDD va être amené à évoluer par la suite en fonction des différents sites et des besoins futurs du CH. Il faut s'assurer qu'un développement futur n'aura pas pour conséquence de redéployer l'application sur chaque site mais bien d'appliquer une mise à jour.

## 5. Bilan

GiDD est le plus gros projet sur lequel j'ai travaillé. Il m'a permis de développer mes compétences en développement Web. J'ai pu me familiariser avec le développement en AJAX ainsi que l'utilisation de jQuery qui sont des technologies que je connaissais peu. Cela m'a appris à créer des pages dynamiques et ergonomiques.

J'ai également découvert le framework FPDF qui permet de générer des fichiers PDF en PHP.

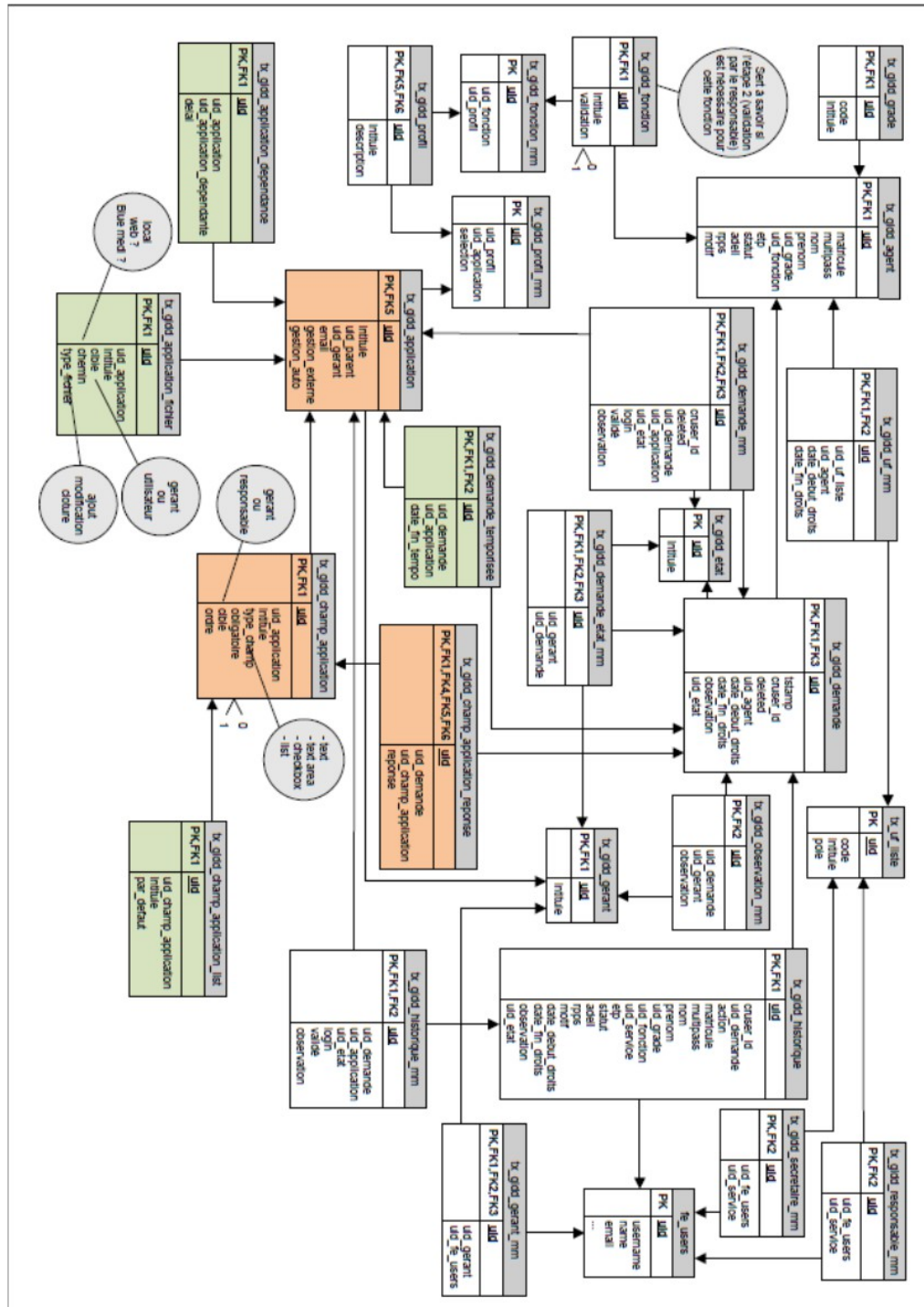
Les soucis présents sur la première version de l'application m'ont permis de comprendre en quoi l'organisation du code et sa documentation est importante pour la maintenance.

Le fait que je sois le seul développeur m'a forcé à être le plus autonome possible. De nombreuses difficultés sont apparues sur ce projet dû au fait que j'utilise des technologies que je ne maîtrise pas forcément. J'ai donc effectué beaucoup de recherches qui m'ont appris à mieux utiliser les documentations notamment celles de JavaScript, PHP et jQuery.

Tout au long du projet j'ai acquis des compétences qui m'ont servis lors de la formation et inversement. Travailler sur GiDD m'a permis de me rendre compte de la réalité du métier de développeur.

## 6. Annexes

Schéma modifié de la base de données. Les tables rouge ont été modifiées et les tables vertes ajoutés.



## Logigramme du traitement d'une demande

