



**AP1 – SEM3 Gestion des visites**

**Alexis DUCHEMANN  
SIO2B**

# Sommaire

<i>Sommaire</i> .....	2
<i>Introduction</i> .....	3
1- Contexte.....	3
2- Cas d'utilisation .....	4
3- Technologies utilisées.....	4
4- Documents (MCD, MLR...) .....	5
5- Arborescence.....	5
<i>Application</i> .....	6
1- Connexion.....	6
2- Barre de navigation.....	7
3- Déconnexion .....	7
4- Vérification de connexion.....	8
5- Création de rapport.....	9
6- Tableau des rapports et médecins.....	13
7- Modification rapport et médecins.....	15

# Introduction

## 1- Contexte :

L'entreprise GSB (Galaxy Swiss Bourdin) est un laboratoire pharmaceutique. Son activité commerciale est principalement réalisée par les visiteurs médicaux. En effet, un médicament remboursé par la sécurité sociale n'est jamais directement vendu au consommateur mais prescrit au patient par son médecin. Toute communication publicitaire sur les médicaments remboursés est d'ailleurs interdite par la loi. Il est important pour l'industrie pharmaceutique de promouvoir ses produits directement aux médecins à l'aide de leurs visiteurs médicaux.

Les visiteurs médicaux ont pour travail de régulièrement visiter les médecins généralistes, spécialistes, services hospitaliers... pour leur tenir au courant de l'intérêt de leur produits et nouveautés du laboratoire. Chaque visiteur dispose d'un portefeuille de praticiens, de sorte que le même médecin ne reçoit jamais deux visites différentes du même laboratoire. Comme tous les commerciaux, ils travaillent par objectifs définis par la hiérarchie et reçoivent en conséquence diverses primes et avantages. Pour affiner la définition des objectifs et l'attribution des budgets, il est nécessaire d'informatiser les comptes rendus de visites.

Les visiteurs ont pour principale activité les visites. Elles sont réalisées auprès de praticien (médecin dans son cabinet, à l'hôpital, pharmacien, chef de clinique...), on souhaite en connaître la date, le motif, et savoir pour chaque visite les médicaments présentés et le nombre offerts. Le bilan fourni par le visiteur devra aussi être enregistré.

Les produits distribués par le laboratoire sont des médicaments : ils sont identifiés par un numéro de produit qui correspond à un nom commercial. Un médicament à des effets thérapeutiques et des contre indications, on connaît sa composition. Lors d'une visite auprès d'un médecin, un visiteur présentera un ou plusieurs produits pour lesquels il pourra laisser des échantillons.

Les médecins sont le cœur de la cible des laboratoires. Ils font l'objet d'une attention toute particulière. Pour tenir à jour leurs informations, les laboratoires achètent des fichiers à des organismes spécialisés qui donnent, les diverses informations d'état civil et la spécialité complémentaire.

## **2- Cas d'utilisation :**

Le site doit répondre à plusieurs cas d'utilisations :

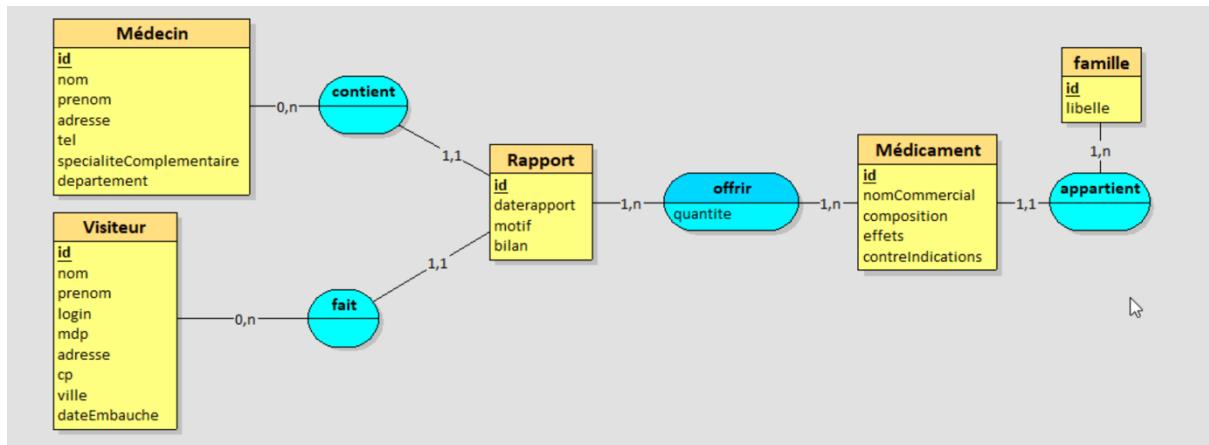
- Création de rapport :
  - o Créer le rapport
  - o Retourner un formulaire avec la liste des médecins et des champs de saisies
  - o Sélectionner un médecin à partir du début de son nom, sélectionner une date, remplir les différents champs et sélectionner les médicaments et les quantités offertes
  - o Enregistrer le rapport
- Modification de rapport :
  - o Le visiteur demande à modifier un rapport
  - o Le système affiche un formulaire pour choisir la date du rapport
  - o Le visiteur sélectionne ensuite la date
  - o Le système affiche tous les rapports effectués à cette date
  - o Le visiteur sélectionne un rapport
  - o Le système renvoie un formulaire pour modifier le motif et le bilan
  - o Le visiteur modifie les informations
  - o Le système enregistre les modifications
- Gérer les médecins
  - o Voir les informations d'un médecin
  - o Formulaire de recherche de médecins
  - o Le système renvoie les informations personnelles du médecin
  - o Le visiteur clique sur le numéro de téléphone
  - o Le système compose le numéro
  - o Le visiteur souhaite voir les anciens rapports de visites de ce médecin
  - o Le système renvoie tous les rapports
  - o Le visiteur demande à modifier les informations d'un médecin
  - o Le système enregistre ces informations

## **3- Technologies utilisées :**

- HTLM : structure du site
- CSS : mise en forme du site
- Javascript : bouton..
- Jquery : permettre d'utiliser DataTables
- DataTables : formateur de tableau
- MAMP : localhost pour le site avec BDD mysql

#### 4- Documents (MCD, MLR...) :

MCD :

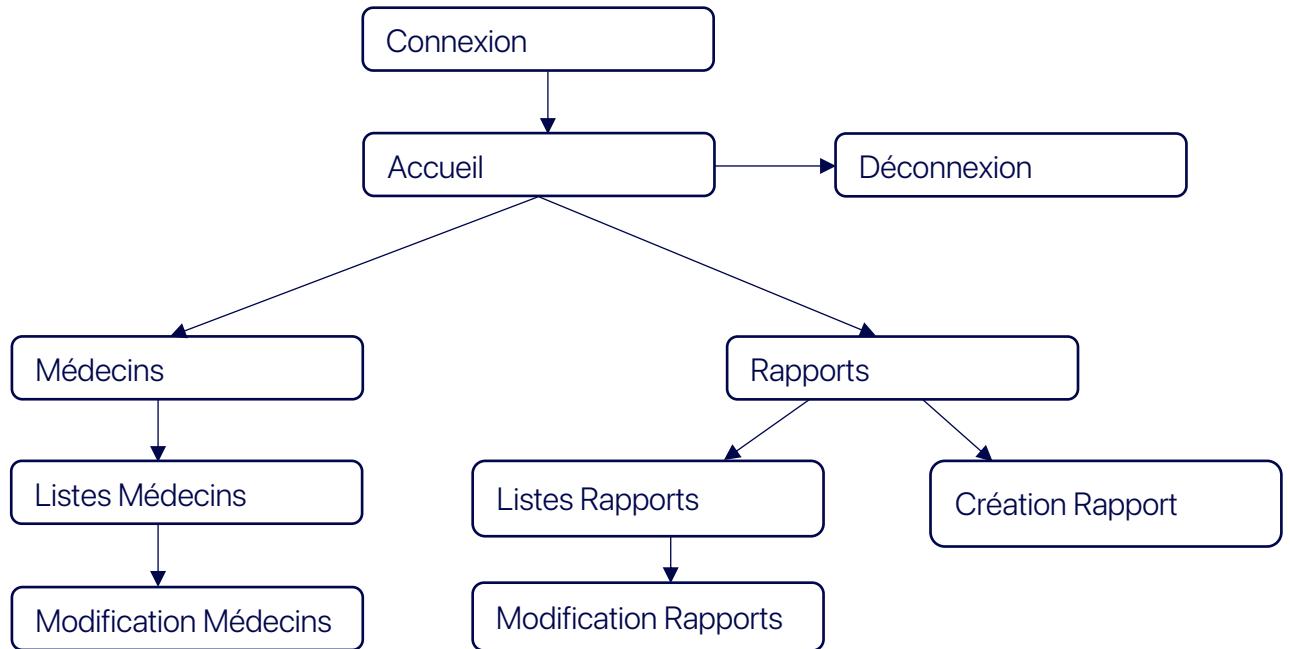


MLR :

```

Médecin = (idMédecin INT, nom VARCHAR(50), prenom VARCHAR(50), adresse VARCHAR(100), tel BIGINT, specialiteComplementaire VARCHAR(50), departement INT);
Visiteur = (idVisiteur INT, nom VARCHAR(50), prenom VARCHAR(50), login VARCHAR(50), mdp VARCHAR(50), adresse VARCHAR(50), cp INT, ville VARCHAR(50), dateEmbauche VARCHAR(50));
famille = (idFamille INT, libelle VARCHAR(50));
Rapport = (idRapport INT, daterapport DATE, motif VARCHAR(50), bilan VARCHAR(100), #idVisiteur, #idMédecin);
Médicament = (idMédicament INT, nomCommercial VARCHAR(50), composition VARCHAR(50), effets VARCHAR(100), contreindications VARCHAR(50), #idFamille);
offrir = (#idRapport, #idMédicament, quantite VARCHAR(50));
  
```

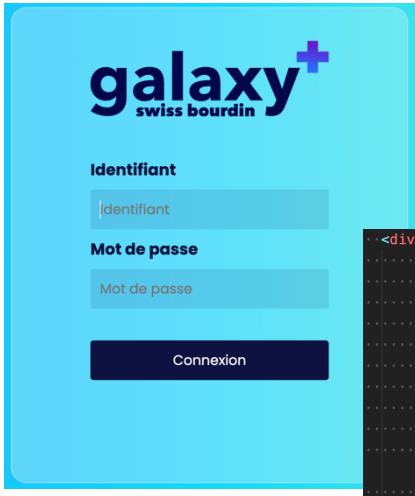
#### 5- Arborescence :



# Application

## 1- Connexion :

La page de connexion sera la page qui s'affichera toujours quand le visiteur n'est pas connecté, elle se présente ainsi :



Ici nous avons le formulaire de connexion qui récupère un identifiant et un mot de passe et qui l'envoie ensuite vers le contrôleur.

```
<div id="formlogin">
    <div id="loginchild">
        <a href=".?action=connexion"></a>
        <form action=".?action=connexion" method="post" id="formloginpage">
            <a>Identifiant</a>
            <input type="text" name="login" autofocus placeholder="Identifiant" required /><br />
            <a>Mot de passe</a>
            <br>
            <input type="password" name="mdp" autofocus placeholder="Mot de passe" required id="myInput" /><br />
            <input type="submit" value="Connexion">
        </form>
    </div>
</div>
```

Grâce au contrôleur ci-dessous nous récupérons les données qui ont été préalablement insérés grâce au \$POST\_. Ensuite nous allons donner aux variables \$loginvisiteur et \$mdpvisiteur leur \$POST\_ respectif.

```
10 | $loginvisiteur=null;
11 | $mdpvisiteur=null;
12 |
13 | if (isset($_POST["login"]) && isset($_POST["mdp"])){
14 |     $loginvisiteur=$_POST["login"];
15 |     $mdpvisiteur=$_POST["mdp"];
16 |
17 |
18 |     $requestconnexion = visiteurDAO::login($loginvisiteur, $mdpvisiteur);
```

Le contrôleur va ensuite lancer la requête de connexion à l'aide de la fonction disponible dans la classe visiteurDAO qui est : login(\$login, \$mdp).

```
35 |     public static function login($login, $mdp){
36 |         if (!isset($_SESSION)){
37 |             session_start();
38 |         }
39 |         $ret=false;
40 |
41 |         $visiteur=visiteurDAO::getvisiteurbyloginDAO($login);
42 |         $mdpbd=$visiteur->getmdpvisiteur();
43 |         $loginbd=$visiteur->getloginvisiteur();
44 |
45 |         if (trim($mdpbd) == trim($mdp) && $login == $loginbd) {
46 |             $_SESSION["login"] = $loginbd;
47 |             $_SESSION["mdp"] = $mdpbd;
48 |             $ret=true;
49 |
50 |         }
51 |     return $ret;
52 | }
```

Cette fonction va permettre de vérifier si les éléments insérés sont bien dans la base de données. Si alors ils sont bien dans la BDD alors on va avoir une session qui s'ouvre et le succès de la connexion.

## 2- Barre de navigation



La barre de navigation se situe en haut du site web, elle permet un accès rapide aux différentes pages comme nous pouvons le voir ici avec un lien vers la page médecins et un lien vers la page rapport.

## 3- Déconnexion

Comme nous pouvons sur l'image du haut dans la barre de navigation, nous avons un bouton de déconnexion :



Ce bouton va permettre d'exécuter l'action deconnexion,.php qui se trouve dans le dossier contrôleur.

Le contrôleur se présente ainsi, il fait appel à une fonction de la classe visiteurDAO qui va donc permettre la déconnexion.

```
1  <?php
2  if(·$_SERVER["SCRIPT_FILENAME"] ·==· __FILE__ ·){
3  |···$racine="..";
4  }
5
6  include_once "$racine/modele/visiteur.php";
7  include_once "$racine/modele/visiteurDAO.php";
8  include_once "$racine/modele/connexionDAO.php";
9
10 visiteurDAO::logout();
11
12 include "$racine/vue/connexion.html.php";
13 include "$racine/vue/footer.html.php";
14
```

La fonction logout() va donc vérifier si une session est en cours et si et seulement si elle est en cours alors grâce au unset il permettre au visiteur de se déconnecter.

```
69  .... public static function logout(){
70  ....     if( !isset($_SESSION) ){
71  ....         .... session_start();
72  ....     }
73  ....     .... unset($_SESSION["login"]);
74  ....     .... unset($_SESSION["mdp"]);
75  .... }
76
77 }
```

## 4- Vérification de connexion

Tout au long de la présence du visiteur sur le site, nous allons vérifier pour des fins de sécurité si le visiteur est bien connecté afin de ne pas divulguer des données personnelles par exemple pour ce faire nous allons utiliser la fonction isLoggedIn() ;

```
public static function isLoggedIn() {
    if (!isset($_SESSION)) {
        session_start();
    }
    $ret = false;

    if (isset($_SESSION["login"])) {
        $visiteur = visiteurDAO::getVisiteurByLoginDAO($_SESSION["login"]);
        if ($visiteur->getLoginVisiteur() == $_SESSION["login"] && $visiteur->getmdpVisiteur() == $_SESSION["mdp"])
        {
            $ret = true;
        }
    }
    return $ret;
}
```

Pour vérifier la connexion nous allons utiliser les variables de sessions et les comparer à leur valeur dans le tableau de visiteur. Cette fonction va pouvoir s'utiliser dans les contrôleurs afin de voir si on peut afficher la page ou non.

Par exemple, ici on vérifie si le visiteur est bien connecté s'il l'est alors nous allons afficher le détail des médecins, sinon on affiche la vue de connexion.

```
17     if (visiteurDAO::isLoggedIn()){
18         include "$racine/vue/entete.html.php";
19         include "$racine/vue/detailmedecin.html.php";
20         include "$racine/vue/footer.html.php";
21     }
22     else {
23
24         include "$racine/vue/vueconnexion.html.php";
25
26     }
27 }
```

## 5- Création de rapport

Le site va nous permettre la rédaction de rapport, pour ce faire nous allons utiliser le formulaire en dessous, le visiteur peut rentrer une date, un motif, un bilan, choisir un médicament et sa quantité et un médecin.

The diagram illustrates the data flow from a report creation form to a dropdown menu for selecting a doctor. The form fields include:

- Date: dd/mm/yyyy
- Motif du rapport: demande de visite
- Bilan du rapport: médecin très intéressé
- Liste des médicaments: AMOPIL
- Quantité: 2
- Liste des médecins: AGOSTINI Amélie
- Validation button: Valider

Arrows indicate the flow of data from the form fields to the dropdown menu. A large arrow points from the "Liste des médecins" field to a dropdown menu containing a list of names. Another arrow points from the "Valider" button in the main form to the same dropdown menu.

**Dropdown Menu (Liste des médecins):**

- ✓ ABEILLE Marie
- ACHARD Dominique
- ADAM Anselme
- AGNEL Béatrice
- AGOSTINI Amélie
- AGRESTI Irénée
- AGUENI Roby
- AILLAUD André
- ALARCON Andrew
- ALBANO Rosalie
- ALBERT Armelle
- ALBERTINI Gilles

**Dropdown Menu (Liste des médicaments):**

- ✓ TRIMYCINE
- ADIMOL
- AMOPIL
- AMOXAR
- AMOXI Gé
- APATOUX Vitamine C
- BACTIGEL
- BACTIVIL
- BIVALIC
- CARTION
- CLAZER
- DEPRAMIL
- DIMIRTAM
- DOLORIL
- NORMADOR
- EQUILAR
- EVEILLOR
- INSECTII

Les deux listes déroulantes sont possibles grâce à :

```
<select name="medicament" required>
    <option value="" disabled selected hidden> Liste des médicaments </option>
    <?php
    for ($i = 0; $i < count($collectionmedicaments); $i++) { ?>
        <option value="<?php echo $collectionmedicaments[$i]->getidmedicament(); ?>><?php echo $collectionmedicaments[$i]->getnomcommercial(); ?>"><?php echo $collectionmedicaments[$i]->getnomcommercial(); ?></option>
    <?php } ?>
</select>
```

```
<select name="medecin" class="ui fluid search dropdown" required>
    <option value="" disabled selected hidden> Liste des médecins </option>
    <?php
    for ($i = 0; $i < count($collectionmedecins); $i++) { ?>
        <option value="<?php echo $collectionmedecins[$i]->getidmedecin(); ?>><?php echo $collectionmedecins[$i]->getnommedecin() . " " . $collectionmedecins[$i]->getprenommedecin(); ?>"><?php echo $collectionmedecins[$i]->getnommedecin() . " " . $collectionmedecins[$i]->getprenommedecin(); ?></option>
    <?php } ?>
</select>
```

Les collections de médicament et de médecins sont créées dans le contrôleur :

```
$collectionmedecins = medecinDAO::getMedecinAscDAO();
$collectionmedicaments = medicamentDAO::getMedicamentDAO();
```

Elles font appels à deux fonctions, getMedecinAscDAO() de la classe medecinDAO qui va nous permettre de récupérer la liste des médecins par ordre alphabétique et getMedicamentDAO de la classe medicamentDAO qui va nous permettre de récupérer une liste de médicaments.

```
public static function getMedicamentDAO()
{
    $res = array();
    $connect = connexionDAO::connexionPDO();
    $request = $connect->prepare("select * from medicament");
    $request->execute();
    $ligne = $request->fetchall();
    foreach ($ligne as $key => $value) {
        $collectionmedicament = new medicament($value['id'], $value['nomCommercial'], $value['idFamille'], $value['composition'], $value['effets'], $value['contreIndications']);
        $res[] = $collectionmedicament;
    }
    return $res;
}
public static function getMedecinAscDAO(){
    $connect = connexionDAO::connexionPDO();
    $request = $connect->prepare("select * from medecin order by nom ASC");
    $request->execute();
    $ligne=$request->fetchall();
    foreach($ligne as $key => $value){
        $collectionmedecin = new medecin($value['id'],$value['nom'],$value['prenom'],$value['adresse'],$value['tel'],$value['specialitecomplementaire'],$value['departement']);
        $res[]=$collectionmedecin;
    }
    return $res;
}
```

Ensuite dans la vue nous avons deux for() qui vont nous permettre de tous les afficher en tant qu'option dans le select. Nous allons maintenant voir le formulaire dans son ensemble et comment il fonctionne.

```
<div class="rapp1">
-- <a class="cool"> <i class="fas fa-edit"></i>Créer un rapport</a>
-- <form action="./?action=CreationRapport" method="post" id="formrapport">

-- <input type="date" name="date" value=<?php date('Y-m-d'); ?>" required />
-- <textarea name="motif" placeholder="Motif du rapport" required></textarea>
-- <textarea name="bilan" placeholder="Bilan du rapport" required></textarea>
-- <select name="medicament" required>
--     <option value="" disabled selected hidden> Liste des médicaments </option>
--     <?php
--         for ($i = 0; $i < count($collectionmedicaments); $i++) { ?>
--             <option value=<?php echo $collectionmedicaments[$i]->getIdmedicament(); ?>><?php echo $collectionmedicaments[$i]->getnomcommercial(); ?></option>
--         <?php } ?>
--     </select>

-- <input type="number" name="quantite" value="quantite" placeholder="Quantité" required />

-- <select name="medecin" class="ui fluid search dropdown" required>
--     <option value="" disabled selected hidden> Liste des médecins</option>
--     <?php
--         for ($i = 0; $i < count($collectionmedecins); $i++) { ?>
--             <option value=<?php echo $collectionmedecins[$i]->getIdmedecin(); ?>><?php echo $collectionmedecins[$i]->getnommedecin() . " "; $collectionmedecins[$i]->getprenommemedecin(); ?></option>
--         <?php } ?>
--     </select>
--     <input type="submit" value="Valider" required />
-- </form>
</div>
```

Le formulaire porte sur l'action CreationRapport du contrôleur. Il récupère les données à travers les name que nous utiliserons dans le contrôleur avec \$\_POST.

```

if (isset($_POST["date"])){$date = date('Y-m-d', strtotime($_POST['date']));}
if (isset($_POST["motif"])){$motif = $_POST["motif"];}
if (isset($_POST["bilan"])){$bilan = $_POST["bilan"];}
if (isset($_POST["medicament"])){$medicament = $_POST["medicament"];}
if (isset($_POST["quantite"])){$quantite = $_POST["quantite"];}

$loginVisiteur = $_SESSION['login'];
$VisiteurRapport=visiteurDAO::getvisiteurbyloginDAO($loginVisiteur);
$id = $VisiteurRapport->getidvisiteur();

if (isset($_POST["medecin"])){$medecin = $_POST["medecin"];}

$req_createrapport = rapportDAO::createrapport($date, $motif, $bilan, $id, $medecin);

$lesRapports = rapportDAO::getRapportsDAO();
$dernierrapport = end($lesRapports);
$iddernierrapport = $dernierrapport->getidrapport();
$req_addoffer = OffrirDAO::addMedicamentRapport($iddernierrapport, $medicament, $quantite);

if ($req_createrapport == true && $req_addoffer == true){
    include "$racine/vue/entete.html.php";
    include "$racine/vue/confirmationrapport.html.php";
}

```

Dans le contrôleur ici nous pouvons voir que nous allons donner pour valeur à \$date, \$motif, \$bilan, \$medicament, \$quantite, \$medecin leur valeur récupérer dans les \$\_POST. Avec cela nous allons pouvoir faire appel à la fonction createRapport(\$date, \$motif, \$bilan, \$id, \$medecin) de la classe rapportDAO. Nous utilisons \$req\_createrapport pour vérifier si l'action a bien été effectué dans le modèle.

```

static public function createrapport($date, $motif, $bilan, $idvisiteur, $idmedecin)
{
    $check = false;

    try {
        $connect = connexionDAO::connexionPDO();
        $request = $connect->prepare('INSERT INTO rapport (date, motif, bilan, idVisiteur, idMedecin)
VALUES (:date, :motif, :bilan, :idVisiteur, :idMedecin)');
        $request->bindValue(':date', $date, PDO::PARAM_STR);
        $request->bindValue(':motif', $motif, PDO::PARAM_STR);
        $request->bindValue(':bilan', $bilan, PDO::PARAM_STR);
        $request->bindValue(':idVisiteur', $idvisiteur, PDO::PARAM_STR);
        $request->bindValue(':idMedecin', $idmedecin, PDO::PARAM_INT);
        $request->execute();
        $check = true;
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage();
        die();
    }
    return $check;
}

```

S'il passe dans la fonction et que le create a bien été fait alors le résultat de \$req\_createrapport sera égale à true sinon false. Cette variable nous servira plus tard pour la confirmation de création.

Après nous allons récupérer l'id du dernier rapport afin de pouvoir utiliser la prochaine fonction. De plus lors de la rédaction d'un rapport nous avons une \$req\_adoffer qui prend la fonction addMedicamentRapport(\$iddernierrapport, \$medicament, \$quantite) de la classe offrirDAO qui va nous permettre d'ajouter un médicament à un rapport.

```
public static function addMedicamentRapport($unRapport, $unMedicament, $uneQuantite)
{
    $check = false;
    try {
        $connect = connexionDAO::connexionPDO();
        $request = $connect->prepare("INSERT INTO offrir (idRapport, idMedicament, quantite)
        VALUES (:idRapport, :idMedicament, :quantite)");
        $request->bindValue(':idRapport', $unRapport, PDO::PARAM_INT);
        $request->bindValue(':idMedicament', $unMedicament, PDO::PARAM_STR);
        $request->bindValue(':quantite', $uneQuantite, PDO::PARAM_INT);
        $request->execute();
        $check = true;
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage();
        die();
    }
    return $check;
}
```

Si la création s'est bien faite nous allons récupérer comme valeur pour \$req\_adoffer true sinon false. \$req\_adoffer et \$req\_createrapport vont donc nous servir à vérifier si la création du rapport a été faite et si nous pouvons afficher la confirmation de création. Lors de la création nous pouvons retrouver le dernier rapport écrit en haut de la page d'accueil dans la partie dernier rapports.

## 6- Tableau des rapports et médecins

Les tableaux des médecins et des rapports utilisent le même fonctionnement, pour les afficher j'utilise l'outil DataTables (jquery) afin d'avoir une pagination et un champ de recherche. Nous allons prendre comme exemple le tableau des médecins mais le fonctionnement reste le même.

Identifier	Nom	Prénom	Adresse	Téléphone	Spécialité	Département	Modifier	Rapports
1	MARTINI	ALAIN	9 Rue Huntziger	<a href="#">695048525</a>	Bon contact	6	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	BLANC	Anne-Lucie	39 rue des gatinnes BILLIAT 01200	<a href="#">356895400</a>	kiné	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	GARCIA	Camille	58 rue du stade MESSINCOURT 08110	<a href="#">0365489929</a>		8	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	MARTINEZ	Alice	12 rue des Pigeons JOIGNY-SUR-MEUSE 08700	<a href="#">0319692016</a>		8	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Duchemann	Alexis	9 Rue Huntziger	<a href="#">695048525</a>	médecin chirurgien	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	ROUX	Anne-Lucie	49 rue des Ormes ATTILLY 02490	<a href="#">313817061</a>	DENTISTE	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	FABRE	Marie	78 rue de Poligny YONCQ 08210	<a href="#">0388716930</a>	HOMEOPATHIE	8	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	ARNAUD	Andrew	29 rue des Pigeons SAVINES-LE-LAC 05160	<a href="#">0477740994</a>		5	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	FERNANDEZ	Julien	45 rue du général Scott THIN-LE-MOUTIER 08460	<a href="#">0321760709</a>	MEDECINE APPLIQUEE AUX SPORTS	8	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	LOPEZ	Aurèle	16 rue Alphonse Daudet ORCIERES 05170	<a href="#">0485568083</a>		5	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Le tableau est écrit comme un tableau normal, j'utilise juste un for afin de récupérer tout la liste des médecins et d'ensuite récupérer les éléments un par un (nom, prénom etc..)

La collection \$listemedecins est récupérer dans le contrôleur.

```
<div>
    <table id="example" class="ui_celled_table">
        <thead>
            <tr>
                <th>Identifier</th>
                <th>Nom</th>
                <th>Prénom</th>
                <th>Adresse</th>
                <th>Téléphone</th>
                <th>Spécialité</th>
                <th>Département</th>
                <th>Modifier</th>
                <th>Rapports</th>
            </tr>
        </thead>
        <php
            for ($i = 0; $i < count($listemedecin); $i++) { ?>
            <tr>
                <td><?php echo "<p>" . $listemedecin[$i] ->getidmedecin() . "</p>"; ?></td>
                <td><?php echo "<p>" . $listemedecin[$i] ->getnommedecin() . "</p>"; ?></td>
                <td><?php echo "<p>" . $listemedecin[$i] ->getprenommemedecin() . "</p>"; ?></td>
                <td><?php echo "<p>" . $listemedecin[$i] ->getadressememedecin() . "</p>"; ?></td>
                <td><?php echo "<a href='tel:'>" . $listemedecin[$i] ->gettelmedecin() . "</a>"; ?>
                <td><?php echo "<p>" . $listemedecin[$i] ->getspecialitecomplementaire() . "</p>"; ?></td>
                <td><?php echo "<p>" . $listemedecin[$i] ->getdepartement() . "</p>"; ?></td>
                <td><?php echo "<a href='/?action=detailmedecin&idmedecin=" . $listemedecin[$i] ->getidmedecin() . "'><i class='fas fa-edit'></i></a></td>
                <td><?php echo "<a href='/?action=modifmedecin&idmedecin" . $listemedecin[$i] ->getidmedecin() . "'><i class='fas fa-list'></i></a></td>
            </tr>
        <?php } ?>
    </table>
</div>
```

Pour mettre en place le tableau avec la pagination et la recherche j'utilise le CDN de jquery et dataTables.

```
<script type="text/javascript" src="https://code.jquery.com/jquery-3.5.1.js"></script>
<script type="text/javascript" src="https://cdn.datatables.net/1.11.3/js/jquery.dataTables.min.js"></script>
```

```
<script>
    $(document).ready(function() {
        $('#example').DataTable({
            "lengthChange": false,
            "language": {
                "search": "_INPUT_",
                "searchPlaceholder": "Recherche...",
                "processing": true,
                "serverSide": true,
                "deferLoading": 57
            });
    });
</script>
```

C'est ce script qui va permettre de formater le tableau et de le rendre plus accessible.

Nous pouvons aussi remarquer deux boutons, l'un va pouvoir permettre au visiteur de modifier les informations d'un médecin, l'autre d'afficher les précédents rapports du médecins concernés.

## 7- Modification rapports et médecins

Les visiteurs ont la possibilité de modifier les informations des médecins et des rapports à travers les formulaires suivants :

**Identifiant :**

1

**Nom :**

MARTINI

**Prénom :**

ALAIN

**Adresse :**

9 Rue Huntziger

**Téléphone :**

695048525

**Spécialité Complémentaire :**

Bon contact

**Département :**

6

[Modifier](#)

**Id rapport :**

1634

**Motif :**

chirurgie

**Bilan :**

nouveau médicaments envoyés

[Modifier](#)

Les informations sont récupérées à travers le contrôleur , nous vérifions si les champs ont été remplies et sinon on reprends la valeur de base.

Ensute nous utiliserons la requête update\_MedecinByID().

```
$idmedecin = $_GET["idmedecin"];
$medecin = medecinDAO::get_medecinById($idmedecin);

if(empty($_POST['id'])) {
    $id = $medecin->getidmedecin();
}
else{
    $id = $_POST["id"];
}

if(empty($_POST['nom'])) {
    $nom = $medecin->getnommedecin();
}
else{
    $nom = $_POST["nom"];
}

if(empty($_POST['prenom'])) {
    $prenom = $medecin->getprenommemedecin();
}
else{
    $prenom = $_POST["prenom"];
}

if(empty($_POST['adresse'])) {
    $adresse = $medecin->getadressememedecin();
}
else{
    $adresse = $_POST["adresse"];
}

if(empty($_POST['tel'])) {
    $tel = $medecin->gettelmetmedecin();
}
else{
    $tel = $_POST["tel"];
}

if(empty($_POST['specialitecomplementaire'])) {
    $specialite = $medecin->getspecialitecomplementaire();
}
else{
    $specialite = $_POST["specialitecomplementaire"];
}

if(empty($_POST['departement'])) {
    $departement = $medecin->getdepartement();
}
else{
    $departement = $_POST["departement"];
}

$req = medecinDAO::update_MedecinById($id, $nom, $prenom, $adresse, $tel, $specialite, $departement);

if ($req == true) {
    include "$racine/vue/entete.html.php";
    include "$racine/vue/confirmationmedecin.html.php";
}
```