

# Documentation technique : GSB Visites

Projet d'AP 2ème année - 1er semestre  
Lim Pascal  
SIO2B



**GSB Visites** est un site web médical permettant à un visiteur médical de gérer les rapports qu'il a réalisés par rapport à ses visites, et il peut aussi consulter et/ou modifier les informations des médecins avec qui il collabore.

Dans l'application, le visiteur peut créer des rapports, les consulter ou bien les modifier. Il peut aussi modifier certaines informations de médecins si elles sont erronées ou si elles ont changées.

# Contexte

## L'activité à gérer

L'activité commerciale d'un laboratoire pharmaceutique est principalement réalisée par les visiteurs médicaux. En effet, un médicament remboursé par la sécurité sociale n'est jamais vendu directement 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 donc important, pour l'industrie pharmaceutique, de promouvoir ses produits directement auprès des praticiens.

## Les Visiteurs Médicaux

L'activité des visiteurs médicaux consiste à visiter régulièrement les médecins généralistes, spécialistes, les services hospitaliers ainsi que les infirmiers et pharmaciens pour les tenir au courant de l'intérêt de leurs produits et des 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 sera nécessaire d'informatiser les comptes rendus de visite.

## L'activité des visiteurs

L'activité est composée principalement de visites : réalisées auprès d'un praticien (médecin dans son cabinet, à l'hôpital, pharmacien, chef de clinique...), on souhaite en connaître la date, le motif (6 motifs sont fixés au préalable), et savoir, pour chaque visite, les médicaments présentés et le nombre d'échantillons offerts. Le bilan fourni par le visiteur (le médecin a paru convaincu ou pas, une autre visite a été planifiée...) devra aussi être enregistré.

## Les médecins

Les médecins sont le cœur de cible des laboratoires. Aussi font-ils 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.

## Les produits

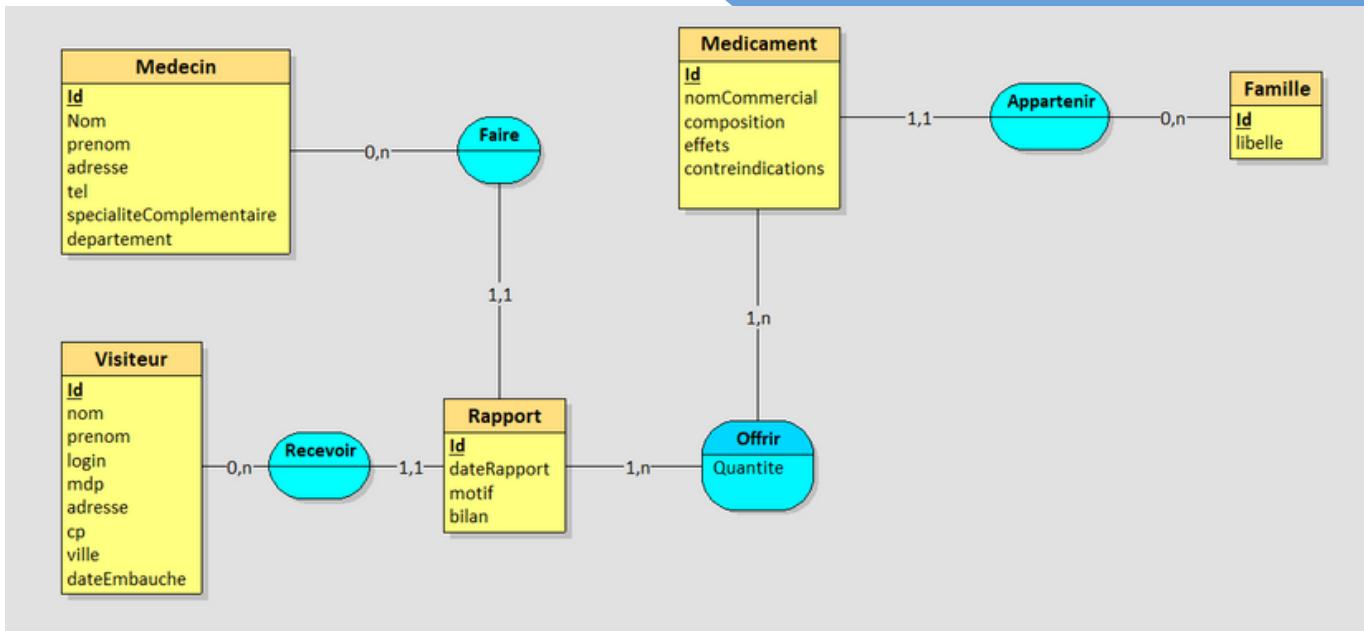
Les produits distribués par les laboratoires sont des médicaments : ils sont identifiés par un numéro de produit qui correspond à un nom commercial (utilisé par les visiteurs et les médecins). Comme tout médicaments, ils possèdent les attributs suivant : une composition, des effets secondaires et des contreindications.

Un produit relève d'une famille (antihistaminique, antidépresseur, antibiotique, ...).

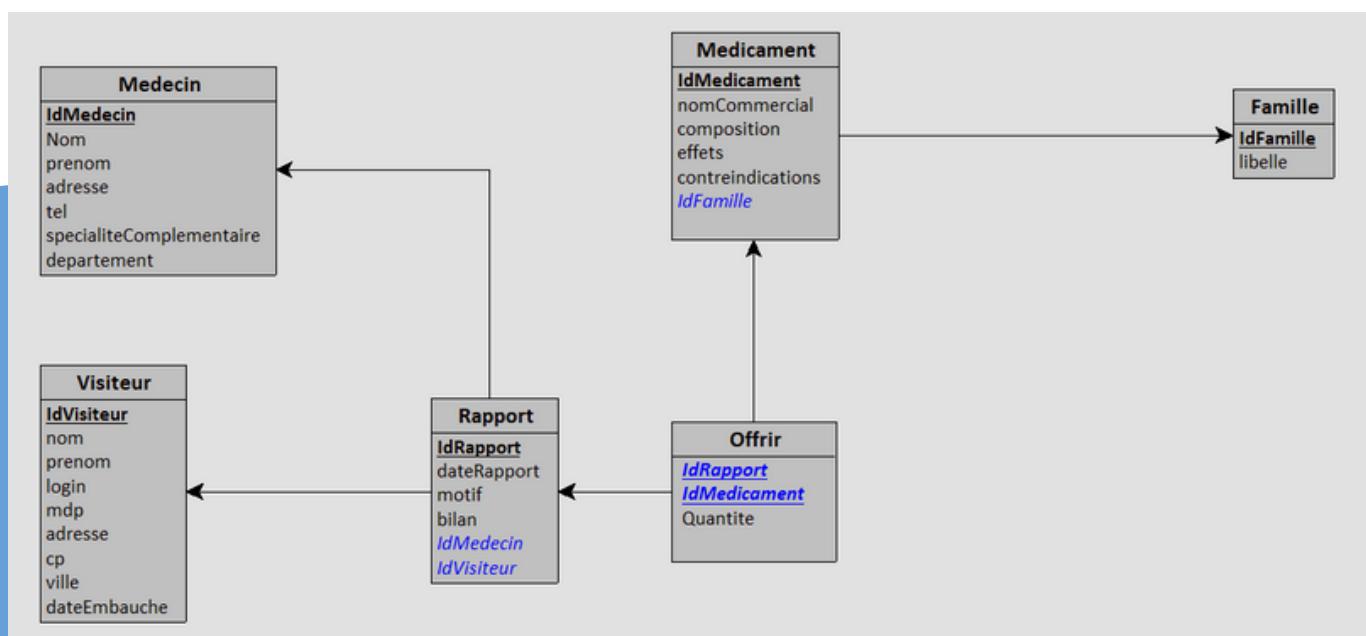
Lors d'une visite auprès d'un médecin, un visiteur présente un ou plusieurs produits pour lesquels il pourra laisser des échantillons.

# Modélisation

## Modèle conceptuel de données :



## Modèle logique des données :



# Présentation des technologies utilisées

Le site web a été réalisé sous le modèle **MVC** (Modèle Vue Contrôleur).

Les langages web **HTML et CSS** ont été utilisées pour tout ce qui est interface, affichage et mise en forme.

Pour interagir avec la base de données et dynamiser le contenu du site, c'est le langage **PHP (orienté objet)** qui est utilisé.

Pour la gestion de la base de données, c'est **MySQL** qui est utilisé.



Le **HTML** est un langage web de type descriptif. C'est lui qui permet de composer des pages web. Son but principal est d'encadrer les différents éléments d'une page afin d'être mis en forme par la suite pour que ce soit agréable visuellement.



Le **CSS** est un langage web de mise en forme et de stylisation, il permet grâce à des feuilles de styles de contrôler l'affichage de chaque éléments d'une page HTML ce qui permet de créer des pages web à l'apparence soignée.



Le **PHP** est un langage de programmation principalement utilisé pour produire des pages web dynamiques via un serveur HTTP (ou local). Il est le plus souvent utilisé côté serveur pour traiter des données et/ou intéragir avec une base de données.



**SQL** est un langage informatique normalisé permettant d'exploiter des bases de données relationnelles. Globalement, il permet de créer ou de modifier la structure de la base de données, de l'optimiser et d'en contrôler l'accès.



L'IDE utilisé pour développer le site est Visual Studio Code.

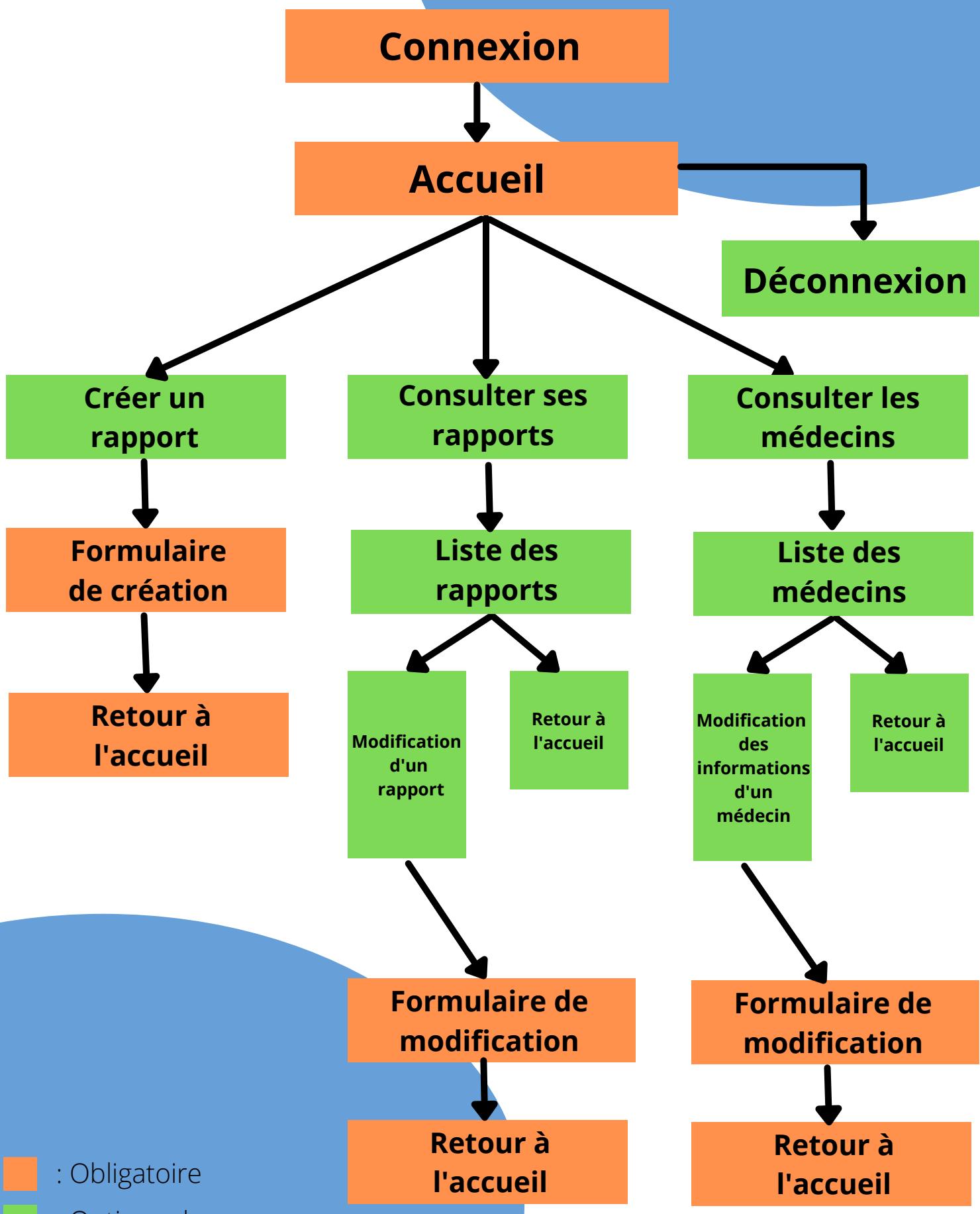


Pour pouvoir utiliser une base de données localement, c'est le serveur UwAmp qui a été utilisé.

## **Sommaire**

<b>1 - Arborescence du site</b>	6
<b>2 - Connexion</b>	7
• Vue	
• Stylisation	
• Traitement des données	
• Gestion des erreurs	
<b>3 - Accueil</b>	12
• Vue	
• Stylisation	
• Affichage des données	
<b>4 - Création/modification</b>	16
• Rapports	
• Médecins	
<b>5 - Consultation</b>	23
• Rapports	
• Médecins	
<b>6 - Recherche/tri</b>	29
• Rapports	
• Médecins	

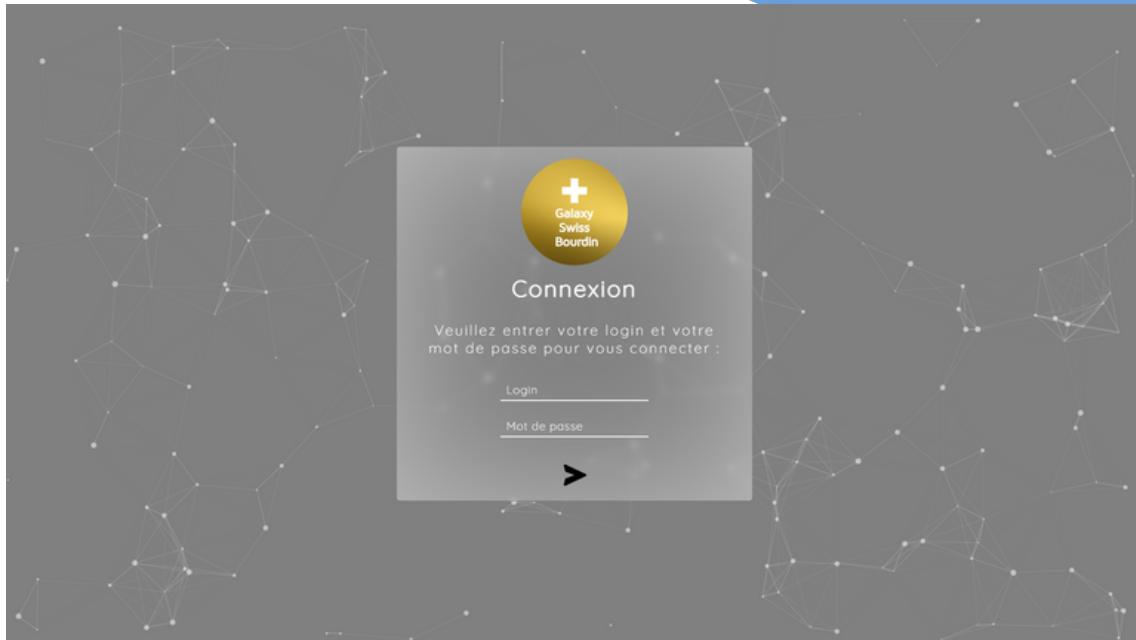
# Arborescence du site



# Connexion

## Vue :

Dès que l'on arrive sur le site, on arrive sur le formulaire de connexion :



ce qui correspond en code HTML à cela :

```
1
2
3 <div id="cnx"> <!-- C'est la card entière qui contient le logo, le formulaire de connexion, ... -->
4
5   <div class="placement-mid">
6     <div id="logo-entier">
7
8       <div class="placement-mid"> <!-- Cette div avec cette classe permet de centrer n'importe quel élément -->
9         <i class='bx bx-plus-medical'></i>
10       </div>
11
12       <div id="nom-logo">Galaxy Swiss Bourdin</div>
13     </div>
14   </div>
15
16   <!-- Cela permet d'afficher un message d'erreur -->
17   <?php if ($erreur!=null){ ?><p class="erreurCnx"><?php echo $erreur; ?> </p><?php }?>
18
19   <h2>Connexion </h2>
20
21   <div class="formulaireConnexion">
22     <form action="./action-connexion" method="post">
23
24       <p class="formTitre">Veuillez entrer votre login et votre mot de passe pour vous connecter : </p>
25
26         <input type="text" name="login" class="page-cnx" autocomplete="off" placeholder="Login" /><br />
27
28         <input type="password" name="mdp" class="page-cnx" placeholder="Mot de passe" /><br />
29
30         <input type="submit" class="boutonForm" value="" />
31     </form>
32   </div>
33
34 </div>
```

# Stylisation :

La stylisation et la mise en forme du formulaire de connexion est réalisé par ce code CSS :

```
/* Page de connexion */
.placement-mid
{
    display: flex;
    justify-content: center;
    align-items: center;
    text-align: center;
}

#cnx
{
    box-shadow: inset 0 0 2000px rgba(255, 255, 255, .5);
    backdrop-filter: blur(6px);
    text-align: center;
    border-radius: 5px;
    width: 40%;
    padding: 20px 50px 50px 50px;
    min-width: 552px;
    max-width: 600px;
    height: fit-content;
    max-height: 670px;
    padding-bottom: 15px;

    /* Centrer un élément HORIZONTALEMENT et VERTICALEMENT en position absolue */
    margin: auto;
    position: absolute;
    top: 0;
    left: 0;
    bottom: 0;
    right: 0;

    /* Centrer un élément en position absolue */
    /* position: absolute;
    margin-left: auto;
    margin-right: auto;
    top:10vh;
    left: 0;
    right: 0;
    text-align: center; */

}

.page-cnx /* Input login de la page de connexion */
{
    background: none;
    outline: none;
    padding: 5px 0 5px 10px;
    width: 50%;
    height: 10px;
    color: white;
    letter-spacing: 1px;
    font-family: 'Quicksand', sans-serif;
    font-size: 1.2em;
    margin-top: 15px;
    margin-bottom: 10px;
    transition-duration: 500ms;

    border-top: 0px;
    border-right: 0px;
    border-left: 0px;
    border-bottom: 2px solid white ;
}

.page-cnx:focus
{
    border-bottom: 2px solid #rgb(255,215,0);
    width: 70%;
}

.page-cnx::placeholder
{
    color: white;
}

input#zoneCnx:focus
{
    /* border: 2px solid rgb(255,215,0); */
    border: 3px solid #859398;
    width: 70%;
}

input#zoneCnx.page-med:focus
{
    background-color: black;
}

#zoneCnx::placeholder
{
    color: white;
}

.boutonForm
{
    color: white;
    letter-spacing: 2px;
    margin-top: 30px;
    padding: 10px;
    font-size: 1em;
    border-radius: 15px;
    background: url("images/send-solid-48.png");
    border: none;
    width: 48px;
    height: 48px;
    background: url("images/send-solid-48.png");
    transition-duration: 500ms;
}

.boutonForm:hover
{
    filter: invert(1);
}

.erreurCnx
{
    color: red;
    border: 2px solid red;
    border-radius: 15px;
    letter-spacing: 3px;
    font-size: 1.3em;
    font-weight: bold;
    padding: 10px 10px ;
    width: 85%;
    margin: 20px auto 0 auto ;
}
```

# Traitement des données :

Le traitement des données envoyées par le formulaire de connexion sont gérée par le contrôleur controleurConnexion.php ci-dessous :

```

<?php

if ( $_SERVER["SCRIPT_FILENAME"] == __FILE__ ){
    $racine="";
}

include_once "getRacine.php";
include_once "$racine/modèle/pdo.php";
include_once "$racine/modèle/visiteur.php";
include_once "$racine/modèle/visiteurDAO.php";
include_once "$racine/modèle/connexionDAO.php";

$erreur =null;

$logInVisiteur=null;
$mdpVisiteur=null;

if ($_POST["login"] != null && $_POST["mdp"] != null){ // on vérifie que les champs ne sont pas nuls
    if (isset($_POST["login"]) && isset($_POST["mdp"])){ // on vérifie que la variable $_POST est bien remplie une 2nde fois

        $logInVisiteur=$_POST["login"];
        $mdpVisiteur=$_POST["mdp"];

        $Visiteur=visiteurDAO::get_visiteurByLogin($logInVisiteur); // récupération des informations de l'utilisateur correspondant au mail entré

        if ($Visiteur != null){ // on vérifie si le login rentré est bien dans la base de données
            // si le mail et le mdp entre sont pareil que ceux de l'utilisateur concerné on crée la connexion
            if ($logInVisiteur == $Visiteur[0]->get_login() && $mdpVisiteur == $Visiteur[0]->get_mdp()) {
                session_start();
                $_SESSION["login"] = $Visiteur[0]->get_login();
                $_SESSION["mdp"] = $Visiteur[0]->get_mdp();
                include "$racine/controleur/controleurAccueil.php";
            }else {
                $title = "Connexion";
                $erreur = "Mauvais login ou mot de passe.";
                include "$racine/vue/vueEntete.php";
                include "$racine/vue/vueConnexion.php";
                require "$racine/vue/vueFooter.php";
            }
        }else{
            $erreur = "Mauvais login ou mot de passe.";
            include "$racine/vue/vueEntete.php";
            include "$racine/vue/vueConnexion.php";
            require "$racine/vue/vueFooter.php";
        }
    }else{
        $erreur = "Veuillez remplir les champs.";

        include "$racine/vue/vueEntete.php";
        include "$racine/vue/vueConnexion.php";
        require "$racine/vue/vueFooter.php";
    }
}
?>
```

### ControleurConnexion.php

Tout d'abord, il faut inclure les classes du modèle dont nous allons avoir besoin pour la connexion; dans ce cas là, les classes Visiteur, VisiteurDAO et connexionDAO.

```

class Visiteur{
    private $IdVisiteur;
    private $NomVisiteur;
    private $PrenomVisiteur;
    private $login;
    private $mdp;
    private $AdresseVisiteur;
    private $Cp;
    private $Ville;
    private $DateEmbauche;
    private $TimeSpan;
    private $Ticket;

    public function __construct ($unIdVisiteur,$unNom,$unPrenom,$unLogin,$unMdp,$uneAdresse,$unCp,$uneVille,$uneDate,$unTimespan,$unTicket){
        $this->IdVisiteur=$unIdVisiteur;
        $this->NomVisiteur=$unNom;
        $this->PrenomVisiteur=$unPrenom;
        $this->login=$unLogin;
        $this->mdp=$unMdp;
        $this->AdresseVisiteur=$uneAdresse;
        $this->CpVisiteur=$unCp;
        $this->VilleVisiteur=$uneVille;
        $this->DateEmbaucheVisiteur=$uneDate;
        $this->TimeSpan=$unTimespan;
        $this->Ticket=$unTicket;
    }

    public function get_IdVisiteur() {return $this->IdVisiteur; }
    public function get_NomVisiteur() {return $this->NomVisiteur; }
    public function get_PrenomVisiteur() {return $this->PrenomVisiteur; }
    public function get_login() {return $this->login; }
    public function get_mdp() {return $this->mdp; }
    public function get_AdresseVisiteur() {return $this->AdresseVisiteur; }
    public function get_CpVisiteur() {return $this->cpVisiteur; }
    public function get_VilleVisiteur() {return $this->VilleVisiteur; }
    public function get_DateEmbaucheVisiteur() {return $this->DateEmbaucheVisiteur; }
    public function get_Timespan() {return $this->TimeSpan; }
    public function get_Ticket() {return $this->Ticket; }
}
```

# Classe VisiteurDAO :

```
class VisiteurDAO{

    static public function get_visiteurByLogin($unLogin){
        $resultat = array();

        try {
            $cnx = pdo::connexionPDO();

            $req = $cnx->prepare('SELECT * FROM visiteur WHERE login=:lelogin;');
            $req->bindValue(':lelogin',$unLogin,PDO::PARAM_STR);
            $req->execute();

            $ligne = $req->fetchall();

            foreach($ligne as $key => $val){
                $visiteur = new Visiteur($val['id'],$val['nom'],$val['prenom'],$val['login'],$val['mdp'],
                $val['adresse'],$val['cp'],$val['ville'],$val['dateEmbauche'],$val['timespan'],$val['ticket']);
                $resultat[] = $visiteur;
            }
        }catch (PDOException $e) {
            print "Erreur !: " . $e->getMessage();
            die();
        }
        return $resultat;
    }
}
```

# Classe connexionDAO :

```
class connexionDAO{

    static public function logout() {
        if (!isset($_SESSION)) {
            session_start();
        }
        unset($_SESSION["login"]);
        unset($_SESSION["mdp"]);
    }

    static public function get_loginVisiteurLoggedOn(){
        if (connexionDAO::isLoggedOn() == true){
            $ret = $_SESSION["login"];
        }
        else {
            $ret = "";
        }
        return $ret;
    }

    static public function isLoggedOn() []
    {
        if (!isset($_SESSION)) {
            session_start();
        }
        $ret = false;

        if (isset($_SESSION["login"])){
            $visiteur = visiteurDAO::get_visiteurByLogin($_SESSION["login"]);
            if ($visiteur[0]->get_login() == $_SESSION["login"] && $visiteur[0]->get_mdp() == $_SESSION["mdp"]){
                $ret = true;
            }
        }
        return $ret;
    }
}
```

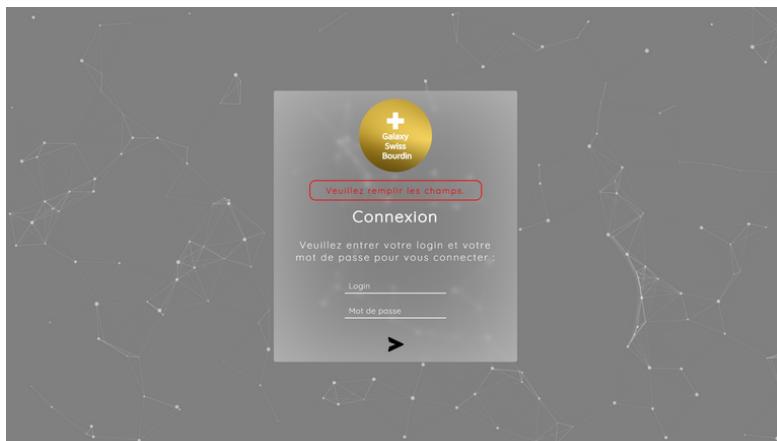
Pour que la connexion se fasse, il est d'abord vérifié si les variables `$_POST["login"]` et `$_POST["mdp"]` sont différentes de "null", c'est-à-dire d'un champ vide et si elles sont bien remplies. Si c'est le cas, le login et le mot de passe envoyés par le formulaire (les `$_POST`) sont chacun stockés dans une variable, `$loginVisiteur` pour le login et `$mdpVisiteur` pour le mot de passe.

Un objet de la classe Visiteur est ensuite créé à partir du login rentré dans le formulaire grâce à la fonction ***get\_VisiteurByLogin()*** de la classe VisiteurDAO qui récupère toute les informations à partir de la BDD du login correspondant (*\$loginVisiteur*). Si le login (*\$loginVisiteur*) et le mot de passe (*\$mdpVisiteur*) rentré sont les même que celui de l'objet Visiteur crée (les *get()* login et mdp), alors la connexion se fait (***session\_start()***) et l'utilisateur accède à la page d'accueil.

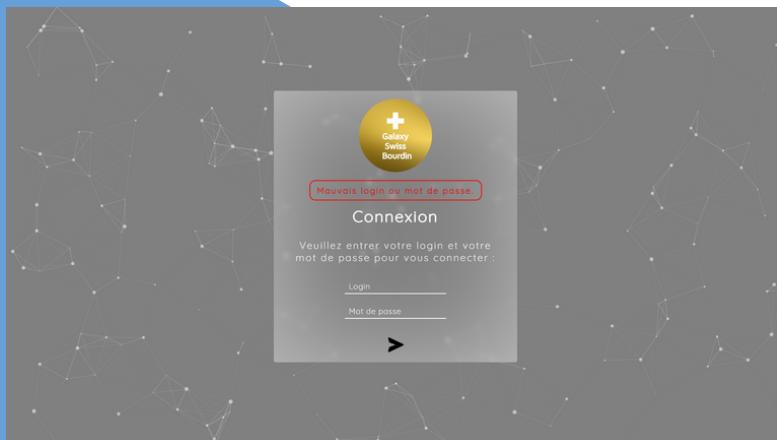
## Gestion des erreurs

Elle correspond à tout les *else* du contrôleur.

Si l'utilisateur a retourné des champs vides, un message d'erreur apparaît lui demandant de remplir les champs :



Si l'utilisateur a rempli les champs mais le login ou le mot de passe n'existe pas dans la base de données : un message d'erreur apparaît prévenant que le login ou le mdp n'est pas bon :



# Accueil

## Vue :

Une fois connecté, voici la vue de l'accueil :

The screenshot shows a dark-themed dashboard. On the left is a sidebar with a user profile picture of a doctor, the name LIM Pascal, address 5 allée du passeur, 93320 PAVILLONS-SOUS-BOIS, and the date of hire: 2021-10-11. Below this are sections for 'Rapports' (Creation d'un rapport, Consultation des rapports), 'Médecins' (Consultation des médecins), and a 'Se déconnecter' button. The main area is titled 'Tableau de bord' and contains a section 'Vos 5 derniers rapports' with a table:

ID	Date	Motif	Bilan	ID du médecin	ID du médicament	Quantité offerte
1691	2021-11-10	Visite du samedi	Très bien passé	981	Aucun	Aucun
1690	2021-11-10	Visite du samedi	Très bien passé	981	Aucun	Aucun
1689	2021-11-04	Visite hebdomadaire	Rien	981	AMOPIL7	2
1686	2021-11-07	UFC 268	gogog	716	AMOPIL7	2
1685	2021-11-04	a ingéré un insecte	Destruction complète de l'insecte	961	INSKTS	2

ce qui correspond en HTML à cela :

- pour la navbar à gauche présente sur TOUTES les pages :

```

1  <!-- Permet de mettre de l'espace (padding: 30px) tout autour du tableau de bord sans pour autant affecter le particles.js du body -->
2  <!-- Le display:flex ici est primordial car c'est lui qui met la navbar et la partie droite côté à côté -->
3  <div style="padding:30px; display:flex; width:100%;>
4
5    <nav id="navbar">
6
7      <div id="navbar-img"></div>
8
9      <div id="info-card-content">
10        <div id="para1">
11          <p><?php echo strtoupper($Visiteur[0]->get_NomVisiteur())." ".$Visiteur[0]->get_PrenomVisiteur()."<br/>">
12          .<?php echo strtoupper($Visiteur[0]->get_AdresseVisiteur())." ".<br/>">
13          .<?php echo strtoupper($Visiteur[0]->get_CpVisiteur())." ".<?php echo strtoupper($Visiteur[0]->get_VilleVisiteur())."<br/>"?>
14          <strong> Date de votre embauche :</strong> <?php echo $Visiteur[0]->get_DateEmbaucheVisiteur(); ?></p>
15        </div>
16
17      </div>
18
19      <a href=".?action=accueilNW" class="menu"> <i class="bx bxs-home'id="home-small-icons" ></i>Accueil</a>
20
21      <h4> Rapports </h4>
22
23        <a href=".?action=CreerRapport' class="menu"><i class='bx bxs-bookmark-alt-plus' id="home-small-icons"></i>Création d'un rapport </a>
24        <a href=".?action=profil' class="menu"> <i class='bx bx-book-open' id="home-small-icons"></i> Consultation des rapports </a>
25
26      <h4> Médecins </h4>
27
28        <a href=".?action=medecin" class="menu"><i class='bx bx-plus-medical' id="home-small-icons"></i> Consultation des médecins </a>
29
30      <a href=".?action=deconnexion" id="deconnexion" class="placement-mid"> Se déconnecter </a>
31
32    </nav>
33

```

- pour la partie à droite de l'accueil :

```
1 <div class="partieAccueil"> <!-- Tout le tableau de bord de l'accueil -->
2
3     <h1 id="accueil">Tableau de bord </h1>
4
5     <div class="placement-mid">
6
7         <div id="home-last-rapport"> <!-- Bloc du tableau concernant le dernier rapport -->
8
9             <div id="home-title-card" class="right-title"><h2>Vos 55 derniers rapports</h2></div>
10
11             <div id="info-card-content" class="home-tab"><?php
12
13                 if($lesRapportVisiteur != null){ ?>
14                     <table id="rapport-table" class="home">
15                         <thead>
16                             <tr>
17                                 <th colspan="1"><strong>Id </strong></th>
18                                 <th colspan="1"><strong>Date </strong></th>
19                                 <th colspan="1"><strong>Motif </strong></th>
20                                 <th colspan="1"><strong>Bilan </strong></th>
21                                 <th colspan="1"><strong>Id du médecin </strong></th>
22                                 <th colspan="1"><strong>Id du médicament </strong></th>
23                                 <th colspan="1"><strong>Quantité offerte </strong></th>
24                             </tr>
25                         </thead>
26
27                         <tbody> <?php
28
29                             $last = count($lesRapportVisiteur);
30                             for ($i = $last - 1 ; $i > $last - 6 ; $i-- ){
31
32                                 $IDMedicamentsRapport= OffrirDAO::get_IdMedicamentOfferRapport($lesRapportVisiteur[$i]->get_IdRapport()); // ID des médicaments du rapport
33                                 $QteMedicamentsRapport= OffrirDAO::get_QteMedicamentOfferRapport($lesRapportVisiteur[$i]->get_IdRapport()); // Quantité des médicaments du rapport
34
35                                 <tr>
36                                     <td><?php echo $lesRapportVisiteur[$i]->get_IdRapport(); ?></td>
37                                     <td><?php echo $lesRapportVisiteur[$i]->get_DateRapport() ?></td>
38                                     <td><?php echo $lesRapportVisiteur[$i]->get_Motif() ?></td>
39                                     <td><?php echo $lesRapportVisiteur[$i]->get_Bilan() ?></td>
40                                     <td><?php echo $lesRapportVisiteur[$i]->get_IdMedecin() ?></td>
41                                     <td><?php
42                                         if($IDMedicamentsRapport != null){
43                                             for ($nombreMedicament = 0 ; $nombreMedicament < count($IDMedicamentsRapport) ; $nombreMedicament++){
44                                                 $IDMedicament = $IDMedicamentsRapport[$nombreMedicament]['idMedicament']; // récupération dans une variable de l'ID du médicament ?>
45                                                 <p><?php echo $IDMedicament ?></p><?php
46                                         }
47                                         else{
48                                             echo ("Aucun");
49                                         }
50                                     }
51
52                                     </td>
53                                     <td><?php
54                                         if ($IDMedicamentsRapport != null){
55                                             for ($nombreMedicament = 0 ; $nombreMedicament < count($IDMedicamentsRapport) ; $nombreMedicament++){
56                                                 $quantiteOfferte = $QteMedicamentsRapport[$nombreMedicament]['quantite']; // récupération de la quantité offerte du médicament ?>
57                                                 <p><?php echo $quantiteOfferte;
58                                         }
59                                         else{
56                                             echo "Aucun";
57                                         }
58                                     }
59
60                                     </td>
61                                     </tr> <?php
62                             } ?>
63
64                         </tbody>
65                     </table> <?php
66                 } ?>
67             </div>
68         </div>
69     </div>
70
71     <div id="info-card-content" class="mid"><?php
72
73         if($message != null){ ?>
74             <h2><?php echo $message ?></h2> <?php
75         } ?>
76
77     </div>
78
79 
```

## Stylisation :

```
361     .partieAccueil
362     {
363         box-shadow: inset 0 0 2000px rgba(255, 255, 255, .5);
364         backdrop-filter: blur(6px);
365         border-radius: 5px;
366         border-top-left-radius: 0px;
367         border-bottom-left-radius: 0px;
368         display: block;
369         width: 80%;
370     }
```

Ce code CSS correspond au background de la partie à droite de toutes les pages du site.

La tableau de l'accueil utilise le même CSS que le tableau de la liste des rapports, sauf que pour le différencier de celui-ci, une `class="home"` a été ajouté sur celui de l'accueil.

Voici ce qui différencie le tableau de rapports de l'accueil de celui de la vue de liste des rapports :

```
430  /* Tableau de l'accueil */
431
432  #home-last-rapport
433  {
434      border-radius: 7px;
435      width: 90%;
436  }
437
438  #info-card-content.home-tab
439  {
440      padding: 0px;
441      border-collapse: none;
442      border-style: none;
443      border-spacing: 0;
444  }
445
446  table#rapport-table.home th
447  {
448      border-top:0px;
449  }
450
451  table#rapport-table.home th:first-child
452  {
453      border-top-left-radius: 0px;
454  }
455
456  table#rapport-table.home th:last-child
457  {
458      border-top-right-radius: 0px;
459  }
```

# Affichage des données :

```
1 <?php
2
3 if ( $_SERVER["SCRIPT_FILENAME"] == __FILE__ ){
4     $racine= "..";
5 }
6
7 $title = "GSB Visites : Accueil";
8
9 include_once "getRacine.php";
10 include_once "$racine/model/e/connexionDAO.php";
11 include_once "$racine/model/e/visiteurDAO.php";
12 include_once "$racine/model/e/rapportDAO.php";
13 include_once "$racine/model/e/offrirDAO.php";
14
15 if ( connexionDAO::isLoggedOn()){
16
17     $visiteur= VisiteurDAO::get_visiteurByLogin($_SESSION['login']); // Obligatoire pour que les infos du visiteur sur la navbar soient affichées
18     $idVisiteur = $visiteur[0]->get_IdVisiteur(); // récupération de l'ID du visiteur
19     $nomVisiteur = $visiteur[0]->get_NomVisiteur(); // récupération du nom du visiteur pour affichage message d'accueil
20     $prenomVisiteur = $visiteur[0]->get_PrenomVisiteur(); // récupération du prénom du visiteur pour affichage message d'accueil
21     @var array $lesRapportVisiteur
22     $lesRapportVisiteur = RapportDAO::get_rapportByIdVisiteur($idVisiteur); // récupération de tout les rapports du visiteur
23
24     // Vérification si le visiteur a des rapports
25     if ($lesRapportVisiteur == null){
26         $message = "Vous n'avez aucun rapport.";
27     }
28
29     require "$racine/vue/vueEntete.php";
30     require "$racine/vue/vueWelcomMessage.php";
31     include "$racine/vue/vueNavbar.php";
32     require "$racine/vue/vueAccueil.php";
33     require "$racine/vue/vueFooter.php";
34 }else{
35     $title = "GSB Visites : Connexion";
36     $erreur = null;
37     include "$racine/vue/vueEntete.php";
38     include "$racine/vue/vueConnexion.php";
39 }
40
41 ?>
```

ControleurAccueil.php

Tout les rapports du visiteur sont stockés dans la variable `$lesRapportVisiteur` lors de l'appel de se contrôleur.

Pour qu'ensuite, les 5 derniers rapports soient affichés, une boucle *for* spéciale a été utilisée : elle commence au *count()* du nombre de rapports de visiteur -1(c'est-à-dire le dernier élément du tableau) et sa condition d'arrêt est que l'indice *i* est supérieur au nombre de rapport - 6 (sachant que dans un tableau, le premier élément est à 0 ce qui implique qu'à l'indice équivalent au nombre total de rapport, il n'y a rien) ce qui va permettre de récupérer les 5 derniers éléments du tableau donc les 5 derniers rapports.

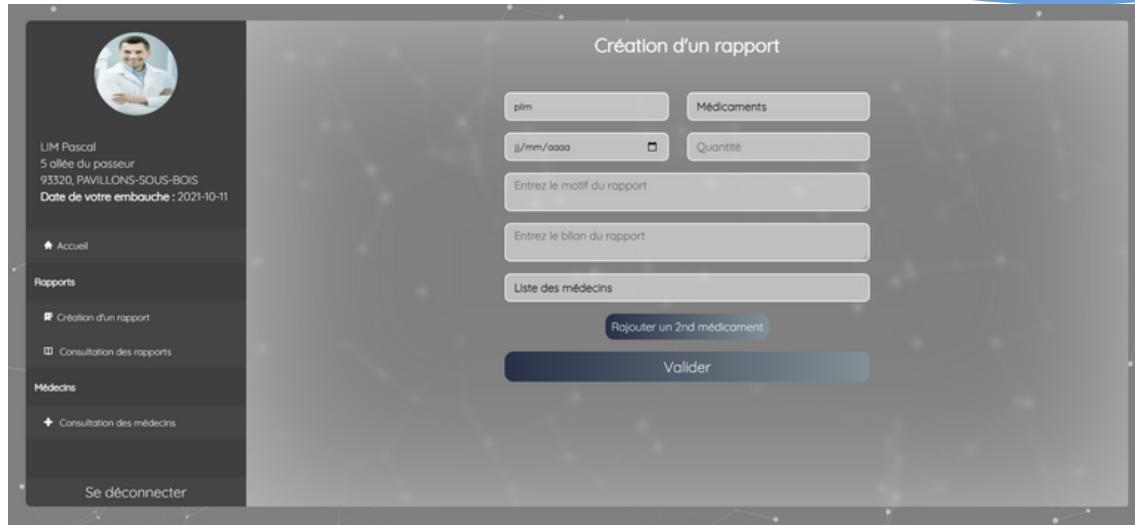
```
$last = count($lesRapportVisiteur);
for ($i = $last - 1 ; $i > $last - 6 ; $i-- ){
```

# Création /modification

## Création :

### Rapports :

Voici la vue qui est affichée lorsque l'on clique sur le bouton de création de rapports :



qui correspond au code HTML suivant :

```
<div class="partieAccueil"> <!-- Cette div avec cette classe correspond à toute la partie droite qui englobe tout son contenu -->
<h1 style="text-align:center;">Création d'un rapport </h1>

<div id="full-formulaire-rapport">

<form action=".?action=CreerRapport" method="post" id="FormRapport">

<div id="form-bloc">
    <div id="form-bloc-left">
        <input type="text" name="idVisiteur" id="FormDate" class="form-disabled" value=<?php echo $VisiteurRapport[0]->get_IdVisiteur(); ?> readonly="true" required/>
        <input type="date" name="dateRapport" value=<?php echo date('Y-m-d'); ?> id="FormDate" required/>
    </div>

    <div id="form-bloc-right">
        <select name="medicament" id="FormMedicament">
            <option value="" disabled selected hidden> Médicaments </option>
            <?php
                for ($i=0; $i<count($lesMedicaments) ; $i++) {
                    <option value=<?php echo $lesMedicaments[$i]->get_IdMedicament(); ?>><?php echo $lesMedicaments[$i]->get_NomCommercial(); ?></option>
                }
            </?php >
        </select>

        <input type="number" name="quantite" value="quantite" id="FormQuantite" placeholder="Quantité" min="0" max="99"/>
    </div>
</div>

<textarea name="motifRapport" placeholder="Entrez le motif du rapport" required></textarea>
<textarea name="bilanRapport" placeholder="Entrez le bilan du rapport" required></textarea>

<select name="medecin" id="FormMedecin" required>
    <option value="" disabled selected hidden> Liste des médecins</option>
    <?php
        for ($i=0; $i < count($lesMedecins) ; $i++) {
            <option value=<?php echo $lesMedecins[$i]->get_IdMedecin(); ?>><?php echo $lesMedecins[$i]->get_nom()." ". $lesMedecins[$i]->get_prenom(); ?></option>
        }
    </?php >
</select>

<!-- C'est une balise en HTML qui permet de cacher du contenu et pouvoir le faire apparaître/disparaître au clic -->
<summary>Rajouter un 2nd médicament</summary>

<select name="medicament2" id="FormMedicament">
    <option value="" disabled selected hidden> Médicaments </option>
    <?php
        for ($i=0; $i<count($lesMedicaments) ; $i++) {
            <option value=<?php echo $lesMedicaments[$i]->get_IdMedicament(); ?>><?php echo $lesMedicaments[$i]->get_NomCommercial(); ?></option>
        }
    </?php >
</select>
```

```

<input type="number" name="quantite2" value="quantite" id="FormQuantite" placeholder="Quantité min="0" max="99"/>
<input type="submit" value="Valider" id="FormSubmitRapport" required/>

</form>
</div>
</div>

```

# Traitement des données :

```

1  <?php
2
3  include "getRacine.php";
4  include "$racine/model/rapportDAO.php";
5  include "$racine/model/medecinDAO.php";
6  include "$racine/model/medicamentDAO.php";
7  include "$racine/model/offrirDAO.php";
8
9  $title = "GSB Visites : Cr ation rapport ";
10
11 if (connexionDAO::isLoggedOn()){
12     $Visiteur= VisiteurDAO::get_visiteurByLogin($_SESSION['login']); // Obligatoire pour que les infos du visiteur sur la navbar soient affich es
13 }
14
15 // Cette condition me permet d'utiliser 2x le m me controleur pour faire la modification d'un m decin
16 // Par dfaut les $_POST sont vides donc la condition est forc m ent vraie et va donc m'afficher la vue du formulaire de modification
17 // Une fois que le formulaire est rempli on fait appelle au m me controleur, sauf que la les $_POST ne seront pas vides donc c'est ce qui est dans le else qui va tre excut .
18 if (empty($_POST['dateRapport']) && empty($_POST['motifRapport']) && empty($_POST['bilanRapport']) && empty($_POST['medicament']) && empty($_POST['quantite']) && empty($_POST['idVisiteur']))
19
20     $lesMedecins = MedecinDAO::get_medecins(); // Agr emer les options du champ select de m decins
21     $lesMedicaments = MedicamentDAO::get_medicaments(); // Agr emer les options du champ select de M dicaments
22
23     $loginVisiteur = $_SESSION['login']; // r cup rer le login du visiteur connect 
24     $VisiteurRapport=VisiteurDAO::get_visiteurByLogin($loginVisiteur); // cr er l'objet Visiteur du visiteur connect 
25     $idVisiteur = $VisiteurRapport[0]->get_IdVisiteur(); // r cup rer l'id du visiteur pour s'en servir dans le formulaire
26
27     include "vue/vueEntete.php";
28     include "vue/vueNavbar.php";
29     include "vue/vueFormulaireRapport.php";
30     require "$racine/vue/vueFooter.php";
31 }else{
32
33     $dateRapport=null; // Date du rapport
34     if (isset($_POST["dateRapport"])){
35         $dateRapport = date('Y-m-d', strtotime($_POST['dateRapport']));
36     }
37
38     $motifRapport = null; // Motif du rapport
39     if (isset($_POST["motifRapport"])){
40         $motifRapport = $_POST["motifRapport"];
41     }
42
43     $bilanRapport = null; // Bilan du rapport
44     if (isset($_POST["bilanRapport"])){
45         $bilanRapport = $_POST["bilanRapport"];
46     }
47
48     $idMedicament=null; // Id du m dicament
49     if (isset($_POST["medicament"])){
50         $idMedicament = $_POST["medicament"];
51     }
52
53     $quantiteOfferte=null; // Quantit  offerte pour le m dicament
54     if (isset($_POST["quantite"])){
55         $quantiteOfferte = $_POST["quantite"];
56     }
57
58     $idMedicamentTwo=null; // Id du deuxi me m dicament
59     if (isset($_POST["medicament2"])){
60         $idMedicamentTwo = $_POST["medicament2"];
61     }
62
63     $quantiteOfferteTwo=null; // Quantit  offerte pour le deuxi me m dicament
64     if (isset($_POST["quantite2"])){
65         $quantiteOfferteTwo = $_POST["quantite2"];
66     }
67
68     $id = null; // Id du visiteur
69     if (isset($_POST["idVisiteur"])){
70         $id=$_POST["idVisiteur"];
71     }
72
73     $medecin=null; // Id du m decin
74     if (isset($_POST["medecin"])){
75         $medecin = $_POST["medecin"];
76     }
77
78     $req = RapportDAO::add_rapport($dateRapport, $motifRapport, $bilanRapport, $id, $medecin); // Ajout du rapport dans la BDD
79
80     $lesRapports = RapportDAO::get_rapports(); // R cup ration de tout les rapports
81     $dernierRapport = end($lesRapports); // R cup ration du dernier rapport ajout 
82     $idDernierRapport = $dernierRapport->get_IdRapport(); // R cup ration de l'id du dernier rapport pour s'en servir en tant que param tre
83
84     if ($idMedicament != null && $quantiteOfferte != null){
85         $reqOffrir = OffrirDAO::Add_MedicamentOffertRapport($idDernierRapport, $idMedicament, $quantiteOfferte); // requ te d'ajout des m dicaments offerts et de la qt  associ e dans
86     }
87
88     if ($idMedicamentTwo != null && $quantiteOfferteTwo != null){
89         $reqOffrirTwo = OffrirDAO::Add_MedicamentOffertRapport($idDernierRapport, $idMedicamentTwo, $quantiteOfferteTwo); // requ te d'ajout des m dicaments offerts et de la qt  assoc
90     }
91
92     if ($req == true){
93
94         $title = "GSB Visites : Confirmation rapport ";
95         include "vue/vueEntete.php";
96         include "vue/vueConfirmationRapport.php";
97         require "$racine/vue/vueFooter.php";
98     }else {
99         echo "Votre rapport n'a pas pu tre ajout ";
100    }
101 }

```

ControleurCreerRapport.php

Tout d'abord, lorsque le contrôleur est appellé, **il nous affiche la vue du formulaire** car la condition *if* avec des *empty(\$\_POST)* (elle vérifie que les *\$\_POST* sont vides) est forcément vraie vu qu'aucune données n'a été envoyée lors de son appel.

C'est lorsque le formulaire est rempli et envoyé que tout ce qui est dans le *else* est exécuté.

Toutes les données envoyées (les *\$\_POST*) sont assignées à une variable chacune qui sera ensuite utilisée en tant que paramètre pour la requête d'ajout du rapport dans la base de données.

Il y a une subtilité ici, c'est la requête d'ajout des médicaments offerts dans la table offrir qui nécessite en paramètre **l'ID du rapport qui vient d'être créé** en plus de l'ID du médicament et de la quantité offerte.

Étant donné que l'ID du rapport est une valeur auto-incrémenté dans la table rapport, pour avoir l'ID de ce rapport qui vient d'être créé (et pas une valeur que l'on a rentré dans le formulaire), juste après avoir fait la requête d'ajout, on récupère TOUT les rapports qui existent (*\$lesRapports*) , et ensuite on récupère le dernier rapport , grâce à la fonction *end()* qui nous permet de récupérer le dernier élément d'un tableau.

Dans ce cas là, c'est exactement ce qu'on a besoin : *\$dernierRapport = end(\$lesRapports)* pour avoir le dernier rapport qui vient d'être créé. Puis on utilise un *get* pour récupérer son ID et s'en servir dans la requête d'ajout dans la table offrir.

# Modification :

## Rapport :

Modification du rapport 1669

plim  
15/10/2021  
Motif  
Bilano  
37  
Valider

La modification d'un médecin se fait sur ce formulaire ci-dessus en cliquant sur le bouton de modification à droite qui apparaît lorsque l'on passe la souris sur la ligne du rapport que l'on souhaite modifier (voir image ci-dessous) :

Liste de vos rapports :

ID	Date du rapport	Motif du rapport	Bilan du rapport	Id du médecin	Id du médicament	Quantité offerte
1668	2021-10-01	Covid	Perdu covid	34	EVIR07	7
1669	2021-10-15	Motif	Bilano	37	PARMOL16	14
1670	2021-10-01	marina pu	toujours le K	739	Aucun médicament proposés	
1671	2021-10-01	Douleurs au genoux	5 séances de kiné prévues	739		2
1672	2021-11-06	Mal de dos	c'est réglé	85	PARMOL16	3
1673	2021-10-31	Réparation dentaire	2 séances à venir	324	Aucun médicament proposés	

En cliquant sur le bouton, on est redirigé vers la page de modification qui correspond au formulaire au dessus.

Le contrôleur qui gère cette modification est celui-ci :

```
1 <?php
2
3 include "getVisite.php";
4 include "visiteur/visiteurDAO.php";
5 include "securite/module/securiteDAO.php";
6 include "securite/module/medicamentDAO.php";
7 include "securite/module/medecinDAO.php";
8
9 if (connexionDAO::isloggedOn()){
10 |   $visiteur= VisiteurDAO::get_visiteurByLogin($_SESSION['login']); // Obligatoire pour que les infos du visiteur sur la navbar soient affichées
11 }
12
13 $idRapport = $_GET['id']; // Récupération de l'id du rapport choisi
14 $leRapport = RapportDAO::get_rapportByIdRapport($idRapport); // Créeation de l'objet rapport avec l'id du rapport en paramètre
15
16 // Cette condition me permet d'utiliser 2x le même contrôleur pour faire la modification d'un médecin
17 // Par défaut les $_POST sont vides donc la condition est forcément vraie et va donc m'afficher la vue du formulaire de modification
18 // Une fois que le formulaire est rempli on fait appelle au même contrôleur, sauf que là les $_POST ne seront pas vides donc c'est ce qui est dans le else qui va être exécuté.
19 if (empty($_POST['motifRapport'])) && empty($_POST['bilanRapport'])){
20
21   $title = "GSM Visites : Modification d'un rapport ";
22
23   include "vue/vueEntete.php";
24   include "vue/vueNavbar.php";
25   include "vue/vueModificationRapport.php";
26   include '$racine/vue/vueFooter.php';
27 }else{
28
29   $nouveauMotif =null;
30   if (isset($_POST['motifRapport'])){
31     $nouveauMotif = $_POST['motifRapport'];
32   }
33
34   $nouveauBilan =null;
35   if (isset($_POST['bilanRapport'])){
36     $nouveauBilan = $_POST['bilanRapport'];
37   }
38
39   $req = RapportDAO::update_rapportById($leRapport->get_IdRapport(), $nouveauMotif, $nouveauBilan);
40
41 if($req == true){
42
43   $title = "GSM Visites : confirmation de modification";
44
45   include "$racine/vue/vueEntete.php";
46   include "$racine/vue/vueConfirmationModificationRapport.php";
47   require "$racine/vue/vueFooter.php";
48 }else{
```

### ControleurModificationRapport.php

Les `$_POST` (les données envoyées du formulaire) sont chacun assignés à une variable pour s'en servir en tant que paramètre dans la requête de modification. Faire cela est utile pour rendre le code plus propre, utiliser les `$_POST` directement va rendre le code moins lisible, c'est pourquoi assigner chaque `$_POST` à une variable est utile pour rendre le code plus compréhensible.

Ici, la fonction utilisée pour modifier un rapport est `update_rapportById()` de la classe RapportDAO.

```
154 static public function update_rapportById($unId,$unMotif,$unBilan){
155
156     $check = false;
157     try{
158         $cnx = pdo::connexionPDO();
159
160         $req = $cnx->prepare("UPDATE rapport SET motif = :motif,bilan = :bilan WHERE id = :id");
161         $req->bindValue(':motif' , $unMotif , PDO::PARAM_STR);
162         $req->bindValue(':bilan' , $unBilan , PDO::PARAM_STR);
163         $req->bindValue(':id' , $unId , PDO::PARAM_INT);
164
165         $req->execute();
166         $check = true;
167     } catch (PDOException $e) {
168         print "Erreur !: " . $e->getMessage();
169         die();
170     }
171     return $check;
172 }
```

### Fonction `update_rapportById()` de la classe RapportDAO

Pour vérifier que la requête s'est bien exécutée, un booléen `$check` est initialisée comme `false` au départ, et si la requête s'est exécutée, `$check` devient `true`.

La fonction doit donc renvoyer un booléen `true` si la requête s'est bien exécutée (d'où la condition `if ($req == true)`).

# Médecin :

The screenshot shows a web-based medical application. On the left, there's a sidebar with a doctor's profile picture, the name LIM Pascal, address (5 allée du posseur, 93320 PAVILLONS-SOUS-BOIS), and a date of hire (2021-10-11). Below this are sections for Accueil, Rapports, and Médecins, with options like 'Création d'un rapport' and 'Consultation des rapports'. A 'Se déconnecter' button is at the bottom. The main area displays a list of doctors in cards:

- Dr. ABEILLE Marie (ID: 218, Adresse: 17 rue Victor Hugo NEUVILLE-LES-DAMES 01400, Département: 18, Téléphone: 456894892, Spécialité complémentaire: ACUPUNCTURE)
- Dr. ABEILLE Marie (ID: 731, Adresse: 28 avenue vercingétorix, Département: 68552, Téléphone: 1235845454, Spécialité complémentaire: Pédiatrie)
- Dr. ADAM Anselme (ID: 301, Adresse: 78 rue de la poste OYONNAX 01100, Département: 1, Téléphone: 0465555519, Spécialité complémentaire: Aucune)

Each card has edit and delete icons at the top right.

Pour modifier un médecin, un bouton de modification apparaît au survol de la souris sur la "card" d'informations du médecin, en appuyant dessus, on accède à la page de modification du médecin ci-dessous.

This screenshot shows the 'Modification des informations d'un médecin' (Edit doctor information) form. It features a preview image of Dr. ABEILLE Marie and fields for her details:

Votre ID :	731
Nom	Prénom
ABEILLE	Marie
Adresse	Département
28 avenue vercingétorix	68552
Téléphone	
1235845454	
Spécialité complémentaire	
Pédiatrie	

A 'Valider' (Validate) button is at the bottom right of the form.

Pour pouvoir récupérer les informations du médecin précisément et les afficher dans le formulaire, on a récupérer l'ID du médecin qui est contenu dans le lien avec un `$_GET['id']` dans le contrôleur.

```
a href=".?action=modificationMedecin&id=<?php echo $lesMedecins[$i]->get_IdMedecin(); ?>" id="medecin-modify-btn" <i class='bx bxs-edit' ></i></a>
```

Cela permet de voir les informations actuelles du médecin dans le formulaire.

Le contrôleur qui gère cette modification est similaire au contrôleur de modification de rapport, seules les variables sont différentes mais son fonctionnement reste le même.

Dans le contrôleur de modification des médecins, c'est cette fonction qui est utilisée :

```
124 |     static public function update_MedecinById($unId,$unNom,$unPrenom,$uneAdresse,$unTel,$uneSpecialite,$unDepartement){  
125 |         $check = false;  
126 |         try{  
127 |             $cnx = pdo::connexionPDO();  
128 |  
129 |             $req = $cnx->prepare("UPDATE medecin SET nom=:nom, prenom=:prenom, adresse=:adresse, tel=:tel, specialitecomplementaire=:specialitecomplementaire, departement=:departement WHERE id=:id");  
130 |  
131 |             $req->bindValue(":id", $unId, PDO::PARAM_INT);  
132 |             $req->bindValue(":nom", $unNom, PDO::PARAM_STR);  
133 |             $req->bindValue(":prenom", $unPrenom, PDO::PARAM_STR);  
134 |             $req->bindValue(":adresse", $uneAdresse, PDO::PARAM_STR);  
135 |             $req->bindValue(":tel", $unTel, PDO::PARAM_INT);  
136 |             $req->bindValue(":specialitecomplementaire", $uneSpecialite, PDO::PARAM_STR);  
137 |             $req->bindValue(":departement", $unDepartement, PDO::PARAM_INT);  
138 |  
139 |             $req->execute();  
140 |  
141 |             $check = true;  
142 |         }catch (PDOException $e) {  
143 |             print "Erreur !: " . $e->getMessage();  
144 |             die();  
145 |         }  
146 |         return $check;  
147 |     }  
148 }
```

Fonction *update\_medecinById()* de la classe **MedecinDAO**

Les *bindvalue()* dans les fonction permettent d'assigner les variables en paramètres à une variable de liaison pour s'en servir dans le *where* de la requête (*:nom* = *\$unNom* par exemple ).

## Stylisation :

Ces formulaires partagent le même CSS, d'où leur ressemblance. Seule leurs mise en forme est différente.

# Consultation

## Rapports :

### Vue :

Voici la vue lorsque l'on clique sur le bouton de consultation des rapports :

ID	Date du rapport	Motif du rapport	Bilan du rapport	Id du medecin	Id du medicament	Quantite offerte
1668	2021-10-01	Covid	Perdu covid	14	EVILR7	7
1669	2021-10-15	Motif	Bilano	37	PARMOL16	14
1670	2021-10-01	marina pu	toujours le K	739	Aucun medicament proposés	
1671	2021-10-01	Douleurs au genoux	5 séances de kiné prévues	739	PARMOL16	2
1672	2021-11-06	Mal de dos	c'est réglé	85	INSXT5	3
1673	2021-10-31	Réparation dentaire	2 séances à venir	324	Aucun medicament proposés	
1674	2021-10-31	Réparation dentaire	2 séances à venir	324	AMOX45	25
1675	2021-10-31	Réparation dentaire	2 séances à venir	324	AMOX45	25

Ce qui correspond en code HTML à cela :

Pour le titre en haut et la barre de recherche par date :

```

1 <div id="profil-content">
2
3   <h1 id="research-title"><?php if ($critere != null){echo ($critere) ;
4     }else{ echo ":";}></h1>
5
6   <?php
7     if($critereRecherche == null){ ?>
8
9     <div id="search-bar-medecin">
10
11       <form action=".?action=rechercheRapport" method="post">
12         <select name="dateRapport" id="zoneEcnx">
13           <option value="" disabled selected hidden>Date du rapport </option>
14           <?php for ( $o=0; $o<count($lesRapportVisiteur) ; $o++) {
15
16             if(in_array($lesRapportVisiteur[$o]->get_DateRapport(),$tableauDate)){ ?>
17               <option value=<?php echo $lesRapportVisiteur[$o]->get_DateRapport(); ?>><strong>Rapports du <?php echo $lesRapportVisiteur[$o]->get_DateRapport(); ?>
18               <?php }
19
20             $tableauDate[] = $lesRapportVisiteur[$o]->get_DateRapport(); } ?>
21           </select>
22
23           <input type="image" id="search-bar-submit-rapport" src="images/search-alt-2-regular-36.png">
24
25       </form>
26     </div> <?php
27   }

```

## Pour le tableau de rapprots :

```
28 if($lesRapportVisiteur != null){ ?>
29
30     <div id="full-rapport-card">
31
32         <table id="rapport-table">
33             <thead>
34                 <th> ID</th>
35                 <th> Date du rapport</th>
36                 <th> Motif du rapport</th>
37                 <th> Bilan du rapport </th>
38                 <th> Id du médecin </th>
39                 <th> Id du médicament </th>
40                 <th> Quantité offerte </th>
41             </thead>
42
43             <tbody>
44
45                 <?php
46                     /* Boucle pour créer 1 ligne du tableau qui correspond à UN rapport entier */
47                     for ( $i=0 ; $i<count($lesRapportVisiteur) ; $i++){ ?>
48
49                         // récupération dans la table offrir de l'ID du médicament et de sa quantité offerte
50                         $IdMedicamentsRapport= OffrirDAO::get_IdMedicamentOffertRapport($lesRapportVisiteur[$i]->get_IdRapport()); // ID DES médicaments du rapport
51                         $QtteMedicamentsRapport= OffrirDAO::get_QteMedicamentOfferRapport($lesRapportVisiteur[$i]->get_IdRapport()); // Quantités DES médicaments du rapport
52
53                         $lesMedicaments=[]; // Initialisation du tableau des médicaments
54
55                         /* Boucle pour récupérer tout les médicaments avec sa quantité associée du rapport d'indice i */
56                         for ( $nombreMedicament = 0 ; $nombreMedicament < count($IdMedicamentsRapport) ; $nombreMedicament++){
57
58                             $idMedicament = $IdMedicamentsRapport[$nombreMedicament]['idMedicament']; // récupération dans une variable de l'ID du médicament
59                             $leMedicament = MedicamentDAO::get_MedicamentById($idMedicament); // création de l'objet médicament avec l'ID en paramètre pour afficher ces données
60
61                             $lesMedicaments[] = $leMedicament; // Stockage des objets médicaments dans un tableau de médicaments
62
63                         }
64
65                         $leMedecin = MedecinDAO::get_medecinById($lesRapportVisiteur[$i]->get_IdMedecin()); // Récupération du médecin concerné pour le rapport d'indice i ?>
66
67                         <tr>
68                             <td><p><?php echo $lesRapportVisiteur[$i]->get_IdRapport(); ?></p></td>
69                             <td><?php echo $lesRapportVisiteur[$i]->get_DateRapport() ?></td>
70                             <td><?php echo $lesRapportVisiteur[$i]->get_Motif() ?></td>
71                             <td><?php echo $lesRapportVisiteur[$i]->get_Bilan() ?></td>
72
73                             <td> <!-- Case du tableau du médecin concerné -->
74                                 <div id="details-hover">
75                                     <p id="details" style="text-align:center;"> <?php echo $lesRapportVisiteur[$i]->get_IdMedecin() ?> </p>
76                                     <span>
77                                         <strong>Nom : </strong><?php echo $leMedecin->get_Nom(); ?> <br/>
78                                         <strong>Prénom : </strong><?php echo $leMedecin->get_Prenom(); ?> <br/>
79                                         <strong>Adresse : </strong><?php echo $leMedecin->get_Adresse(); ?> <br/>
80                                         <strong>Téléphone : </strong><?php echo $leMedecin->get_Tel(); ?> <br/>
81                                         <strong>Spécialité complémentaire : </strong><?php if ($leMedecin->get_Specialite() == null){ echo ("Aucune"); }else{echo $leMedecin->get_Specialite(); ?>
82                                         <strong>Département : </strong><?php echo $leMedecin->get_Departement(); ?> <br/>
83                                     </span>
84                                 </div>
85                             </td>
86
87                             <td> <!-- Case du tableau des médicaments -->
88                                 <?php
89                                     if ($IdMedicamentsRapport == null ){ ?>
90                                         <h3> Aucun médicament proposés </h3> <?php
91                                     }else{
92                                         // Boucle pour parcourir les médicaments (utile s'il y en a plusieurs)
93                                         for ( $j=0 ; $j < count($lesMedicaments) ; $j++){ ?>
94
95                                             <div id="details-hover">
96                                                 <p id="details" style="text-align:center;"><?php echo $lesMedicaments[$j]->get_IdMedicament(); ?> </p>
97                                                 <span>
98                                                     <strong>Nom : </strong><?php echo $lesMedicaments[$j]->get_NomCommercial(); ?> <br/>
99                                                     <strong>Composition : </strong> <?php echo $lesMedicaments[$j]->get_Composition(); ?> <br/>
100                                                     <strong>Effets : </strong> <?php echo $lesMedicaments[$j]->get_Effets(); ?> <br/>
101                                                     <strong>Contreindications : </strong> <?php echo $lesMedicaments[$j]->get_ContreIndication(); ?> <br/>
102                                                 </span>
103                                             </div> <?php
104                                         }
105                                         } ?>
106                                     </td>
107
108                                     <td> <!-- Case du tableau de la quantité de médicaments offerte -->
109                                         <?php
110
111                                         // Boucle pour afficher la quantité offerte du médicament associé (utile s'il y'en a plusieurs)
112                                         for ( $j=0 ; $j<count($lesMedicaments) ; $j++){ ?>
113                                             $quantiteOfferte = $QtteMedicamentsRapport[$j]['quantite'];
114
115                                             <p style="font-size:1.2em;"><?php echo $quantiteOfferte ; ?> </p> <?php
116                                         }
117
118                                         <a href=".?action=modificationRapport&id=<?php echo $lesRapportVisiteur[$i]->get_IdRapport(); ?>" id="rapport-modify-btn" > <i class="bx bxs-edit" >
119                                     </td>
120
121                                     </tr><?php
122                               }
123                           </tbody>
124                       </table>
125                   </div> <?php
126               }else{ ?>
127                   <h1> Vous n'avez aucun rapports. </h1> <?php
128               } ?>
```

Chaque ligne du tableau (*tr*) est créée par une boucle qui itère jusqu'à avoir atteint la fin du tableau de rapports (*\$lesRapportVisiteur*) et correspond à un seul rapport.

Les cases de chaque ligne (*td*) contiennent elles chacun des informations du rapport d'indice i (*[\$i]*) récupérées à l'aide de *getters*.

## Stylisation :

```
758 #search-bar-medecin
759 {
760     position: sticky;
761     top:75px;
762     z-index: 100;
763 }
764
765 select#zoneEcnx
766 {
767     padding: 10px 15px;
768     width: 18%;
769     height: 10%;
770     color: ■whitesmoke;
771     letter-spacing: 1px;
772     border-radius: 15px;
773     font-size: 1.2em;
774     background-color: □black;
775     outline: none;
776     border: 2px solid ■white ;
777     margin-top: 15px;
778     margin-bottom: 11px;
779     margin-right: 15px;
780     transition-duration: 500ms;
781 }
782
```

Ce code correspond au CSS de la barre de recherche par date dans la vue de la liste des rapports.

Le **position:sticky** avec le *top:75px* permet à la barre de recherche de rester collé dans l'écran à 75px du haut de l'écran si l'on scroll alors que normalement nous ne sommes pas censé la voir.

Voici un extrait du code CSS du tableau de rapports :

```
918 /* Tableau liste des rapport */
919
920 table#rapport-table
921 {
922     border-collapse: separate;
923     width: 85w;
924     border-spacing: 0;
925 }
926
927 table#rapport-table th{
928     text-align: center;
929     min-width: 100px;
930     padding: 1em;
931     border-top: 2px solid ■whitesmoke;
932     border-right: 2px solid ■whitesmoke;
933     border-bottom: 2px solid ■whitesmoke;
934     font-size: 1.2em;
935     background-color: □rgb(59, 59, 59);
936     position:sticky;
937     top: 0px;
938 }
939
940 table#rapport-table th:first-child
941 {
942     border-left: 2px solid ■whitesmoke;
943     border-top-left-radius: 15px;
944     border-bottom-left-radius: 15px;
945     z-index: 2;
946 }
947
948 table#rapport-table th:last-child
949 {
950     z-index: 2;
951     border-top-right-radius: 15px;
952     border-bottom-right-radius: 15px;
953 }
954
955 table#rapport-table tr td
956 {
957     transition-duration: 350ms;
958     height: 80px;
959 }
```

# Affichage des données :

```
1 <?php
2
3 if ( $_SERVER["SCRIPT_FILENAME"] == __FILE__ ){
4     $racine="";
5 }
6
7 include_once "getRacine.php";
8 include_once "$racine/modele/connexionDAO.php";
9 include_once "$racine/modele/rapportDAO.php";
10 include_once "$racine/modele/visiteur.php";
11 include_once "$racine/modele/visiteurDAO.php";
12 include_once "$racine/modele/offrirDAO.php";
13 include_once "$racine/modele/medicamentDAO.php";
14 include_once "$racine/modele/medecinDAO.php";
15
16 if (connexionDAO::isLoggedIn()){
17
18     $Visiteur= VisiteurDAO::get_visiteurByLogin($_SESSION['login']); // Obligatoire pour que les infos du visiteur sur la navbar soient affichées
19
20     $loginVisiteur=$_SESSION['login']; // récupération du login du visiteur connecté
21     $leVisiteur = VisiteurDAO::get_visiteurByLogin($loginVisiteur); // création de l'objet visiteur à partir du login en paramètre pour utiliser ces données
22     $idVisiteur = $leVisiteur[0]->get_IdVisiteur(); // récupération de l'ID du visiteur
23
24     $lesRapportVisiteur = RapportDAO::get_rapportByIdVisiteur($idVisiteur); // récupération de tout les rapports du visiteur
25
26     $title = "GSB Visites : Mes rapports ";
27
28     $critereRecherche = null; // Initialisation à null car par défaut, la barre de sélection doit être visible tant qu'aucune recherche n'a été faite
29     $critere = "Liste de vos rapports : "; // Le titre change lorsque une recherche est faite
30
31     include "$racine/vue/vueEntete.php";
32     include "$racine/vue/vueNavbar.php";
33     include "$racine/vue/vueProfil.php";
34     require "$racine/vue/vueFooter.php";
35 }
```

ControleurProfil.php

La variable la plus importante ici est la variable `$lesRapportVisiteurs`, sans elle aucun rapport n'est affiché.

Cette variable contient tout les rapports **du visiteur** grâce à la fonction `get_rapportByIdVisiteur()` avec l'ID du visiteur en paramètre de la classe RapportDAO.

Dans la vue, la boucle itère par rapport à cette variable, si il y a 10 rapports dans cette variable, 10 lignes de rapports seront affichées à l'écran; s'il y'en a zéro, il sera écrit que le visiteur n'a pas de rapports.

```
30     static public function get_rapportByIdVisiteur($unIdVisiteur){
31         $resultat = array();
32
33         try{
34             $cnx = pdo::connexionPDO();
35
36             $req = $cnx->prepare('SELECT R.id,date,motif,bilan,idVisiteur,idMedecin FROM rapport R, visiteur V
37             WHERE R.idVisiteur=:idVisiteur AND R.idVisiteur=V.id');
38             $req->bindValue(':idVisiteur',$unIdVisiteur, PDO::PARAM_STR); // assignation de la valeur en paramètre
39             $req->execute(); // exécution de la requête
40
41             $ligne = $req->fetchall(PDO::FETCH_ASSOC); // récupération des données dans un tableau associatif
42
43             foreach($ligne as $key => $val){
44                 $rapport = new Rapport($val['id'],$val['date'],$val['motif'],$val['bilan'],$val['idVisiteur'],$val['idMedecin']);
45                 $resultat[]=$rapport;
46             }
47         }catch (PDOException $e) {
48             print "Erreur !: " . $e->getMessage();
49             die();
50         }
51     }
52 }
```

Fonction `get_rapportByIdVisiteur()` de la classe RapportDAO

# Médecins :

## Vue :

Voici la vue qui est affichée lorsque l'on clique sur le bouton de consultation des médecins :

The screenshot shows a user interface for managing medical staff. On the left, a sidebar provides access to the dashboard, report creation, report consultation, and doctor management. The main content area is titled "Liste des médecins" and features a search function. Three doctor profiles are listed, each with a picture, name, ID, address, department, phone number, and a link to call the doctor.

Ce qui correspond en HTML à ça :

Pour la barre de recherche et la barre de tri :

```
1 <div id="profil-content">
2   <h1 id="research-title"> Liste des médecins : </h1>
3
4   <div id="search-bar-medecin">
5     <form action="./?action=rechercheMedecin" method="POST">
6       <input type="text" name="nom" id="zoneEcnx" class="page-med" autocomplete="off" required>
7       <input type="image" id="search-bar-submit" src="images/search-alt-2-regular-36.png">
8     </form>
9   </div>
10
11 <?php
12
13 if ($critere != null ){ ?>
14   <h3 id="medecin-search-criteria"><?php echo $critere ?> <a href="./?action=medecin"> X </a></h3> <?php
15 } ?>
16
17 <nav id="medecin-filter-bar">
18
19   <ul id="medecin-filter-bar">
20
21     <li style="margin-right:10px;"><p><strong>Trier par : </strong></p>
22
23       <a href="./?action=medecin&order=id"><li> ID </li></a>
24       <a href="./?action=medecin&order=Specialite"><li> Spécialité </li></a>
25       <a href="./?action=medecin&order=Departement"><li> Département </li></a>
26
27     <li id="medecin-list-nom"><a href="./?action=medecin" id="medecin-list-nom"> Nom </a>
28       <ul id="medecin-list-nom">
29         <li><a href="./?action=medecin&order=AZ" id="medecin-list-nom-under"> A - Z</a></li>
30         <li><a href="./?action=medecin&order=ZA" id="medecin-list-nom-under"> Z - A</a></li>
31       </ul>
32     </li>
33
34   </ul>
35
36 </nav>
```

Pour chaque "card" regroupant les informations d'un médecin :

```
61 if ($lesMedecins != null){ ?>
62 <div id="full-report-card"> @php
63
64     for($i=0 ; $i<count($lesMedecins) ; $i++){ ?>
65         <div id="medecin-card">
66
67             <div id="medecin-img"></div>
68
69             <h2 id="medecin-card-title"> Dr. <php echo $lesMedecins[$i]->get_Nom(); ?> <php echo $lesMedecins[$i]->get_Prenom(); ?></h2>
70             <div id="medecin-infos">
71                 <p><strong>ID : </strong><php echo $lesMedecins[$i]->get_IdMedecin(); ?></p>
72                 <p><strong>Adresse : </strong><php echo $lesMedecins[$i]->get_Adresse(); ?></p>
73                 <p><strong>Département : </strong><php echo $lesMedecins[$i]->get_Departement(); ?></p>
74                 <p><strong>Téléphone : </strong><php echo $lesMedecins[$i]->get_Tel(); ?></p>
75
76                 <div id="medecin-infos-under">
77                     <p class="left-infos-under"><strong>Spécialité complémentaire : </strong><strong><php if ($lesMedecins[$i]->get_Specialite() == null)( echo ("Aucune")); else(echo ($lesMedecins[$i]->get_Specialite())); ?></strong></p>
78                     <p class="right-infos-under"><a href="#tel"><php echo $lesMedecins[$i]->get_Tel(); ?></a> class="tel-link"><strong>Appeler le médecin</strong></a>
79                 </div>
80
81             </div>
82         </div> <?php
83     } ?>
84 } ?>
85 <div> Il n'y a pas de médecins <?php echo $critere ?> </div> <?php
86
87 </div>
88 <a href="#profile-content" id="back-top-button"><?a>
```

## Affichage des données :

```
1  C:\P&P
2  include "Medecin.inc.php";
3  include "MedecinDAO.inc.php";
4  include "MedecinDAO.class.php";
5
6  if (connection::getInstance() != null){
7      $visitor = visitor::getInstance();
8      $visitor->visitMedecin($lesMedecins);
9
10     $title = "TSS Visites : liste des médecins";
11
12     $critere = null; // Initialisation de la variable critère à null car il n'y en a pas par défaut
13
14     // Pour défaut, la variable $critere est nulle, donc elle ne peut pas être assignée à la variable $tri si celle-ci n'existe pas, il faut donc initialiser $tri dès le départ
15     $tri = null;
16
17     if ($critere != null){ // Si critere n'est pas null
18         $tri = $critere;
19     }
20
21     // Si tri n'est pas null, dépendant de la valeur de $critere, trier va faire appelle à une fonction spécifique pour $lesMedecins ce qui affectera la liste des médecins affichés sur la vue
22     // Critere n'a aussi son valeur différente au fonction de tri
23     if ($tri == "ID"){
24
25         $lesMedecins = MedecinDAO::get_medecinsOrderByID();
26
27         $critere = "trié par ID";
28
29     } elseif($tri == "Specialite"){
30
31         $lesMedecins = MedecinDAO::get_medecinsOrderBySpecialite();
32
33         $critere = "trié par spécialité";
34
35     } elseif($tri == "Departement"){
36
37         $lesMedecins = MedecinDAO::get_medecinsOrderByDepartement();
38
39         $critere = "trié par département";
40
41     } elseif($tri == "Nom"){
42
43         $lesMedecins = MedecinDAO::get_medecinsOrderByNom();
44
45         $critere = "trié par nom";
46
47     } elseif($tri == "Prenom"){
48
49         $lesMedecins = MedecinDAO::get_medecinsOrderByPrenom();
50
51         $critere = "trié par prénom";
52
53     } elseif($tri == "Adresse"){
54
55         $lesMedecins = MedecinDAO::get_medecinsOrderByAdresse();
56
57         $critere = "trié par adresse";
58
59     } elseif($tri == "Tel"){
60
61         $lesMedecins = MedecinDAO::get_medecinsOrderByTel();
62
63         $critere = "trié par numéro de téléphone";
64
65     }
66
67     include "visitesMedecin.php";
68     include "MedecinDAO.class.php";
69     include "MedecinDAO.inc.php";
70     require "MedecinDAO.class.php";
71 }
```

### ContrôleurMedecin.php

La variable primordiale dans ce contrôleur est la variable `$lesMedecins`, c'est de elle que dépend l'affichage dans la vue des médecins. La boucle itère par rapport à cette variable.

Par défaut, elle contient **tout** les médecins. Comme nous pouvons le voir, étant donné qu'aucun critère de tri n'est choisi, `$lesMedecins` correspond au résultat de cette fonction `get_medecins()`, c'est-à-dire tout les médecins.

```
8     static public function get_medecins(){
9
10         $resultat=array();
11         try {
12             $cnx = pdo::connexionPDO(); #Connexion à la base de données
13
14             $req = $cnx->prepare('SELECT * FROM medecin ORDER BY nom ASC'); #Préparation de la requête
15
16             $req->execute(); #Exécution de la requête
17
18             $ligne = $req->fetchall();
19
20             foreach ($ligne as $key => $val) {
21                 $collectionMedecins= new Medecin ($val['id'],$val['nom'],$val['prenom'],$val['adresse'],$val['tel'],
22                 $val['specialitecomplementaire'],$val['departement']);
23                 $resultat[]=$collectionMedecins;
24             }
25
26         } catch (PDOException $e) {
27             print "Erreur !: " . $e->getMessage();
28             die();
29         }
30
31         return $resultat;
32     }
```

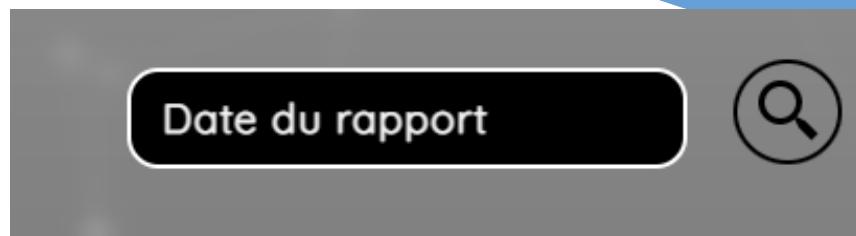
### Fonction `get_medecins()` de la classe `MedecinDAO`

# Recherche/tri

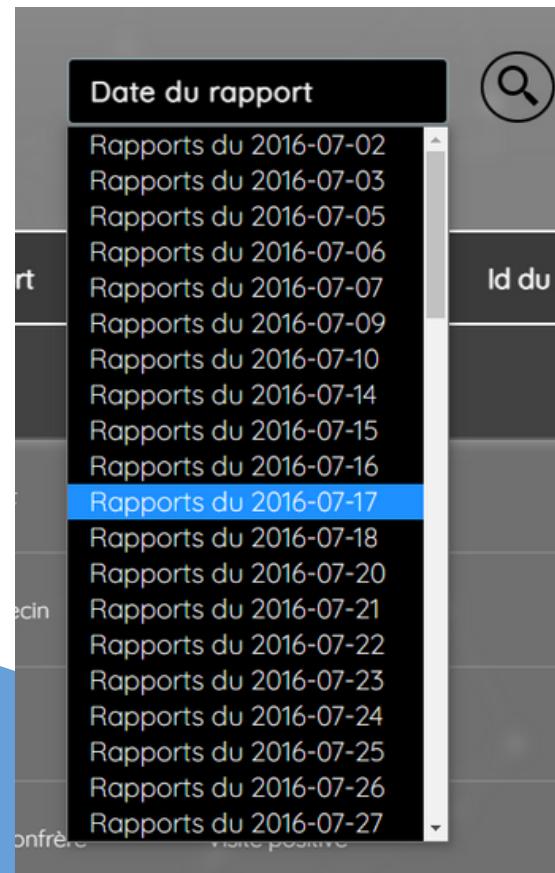
## Rapports :

### Tri des rapports par date :

Dans la vue de la liste des rapports, le tri des rapports par date se fait via cette barre :



Lorsque l'on clique sur cette barre de recherche, nous pouvons voir toutes les dates où il existe au moins un rapport concernant le visiteur :



A partir d'ici, nous pouvons choisir de voir tous les rapports réalisés à une date précise.

Le contrôleur qui gère le tri est celui-ci :

```
1 <?php
2
3 if ( $_SERVER["SCRIPT_FILENAME"] == __FILE__ ){
4     $racine=..;
5 }
6
7 include_once "getRacine.php";
8 include_once "$racine/modele/connexionDAO.php";
9 include_once "$racine/modele/rapportDAO.php";
10 include_once "$racine/modele/visiteur.php";
11 include_once "$racine/modele/visiteurDAO.php";
12 include_once "$racine/modele/offerDAO.php";
13 include_once "$racine/modele/medicamentDAO.php";
14 include_once "$racine/modele/medecinDAO.php";
15
16 $dateRapport=null; // Date du rapport
17 if (isSet($_POST['dateRapport'])){
18     $dateRapport = date('Y-m-d', strtotime($_POST['dateRapport']));
19 }
20
21 if ($dateRapport != null){
22
23     if (connexionDAO::isLoggedOn()){
24         $visiteur= VisiteurDAO::get_visiteurByLogin($_SESSION['login']);
25     }
26
27     $req = RapportDAO::get_rapportByDate($dateRapport);
28
29     $lesRapportVisiteur = $req; // Assignation du résultat de la requête de recherche à $lesRapportVisiteur car on utilise toujours la même vue pour l'affichage des rapports
30
31     $titre="GSB Visites : résultat de votre recherche";
32
33     $critereRecherche = $dateRapport;
34     $critere = " Liste de vos rapport datant du ".$critereRecherche."";
35
36     include "vue/vueEntete.php";
37     include "vue/vueNavbar.php";
38     include "vue/vueProfil.php";
39     require "$racine/vue/vueFooter.php";
40 }
```

### ControleurRechercheRapport.php

On récupère tout les rapport de la date choisie grâce à la fonction *get\_rapportByDate()* de la classe RapportDAO qui utilise en paramètre la date choisie.

Étant donné qu'on utilise la même vue pour l'affichage de la liste des rapports, on utilise la même variable *\$lesRapportVisiteur*, tout ce qui change est son contenu. La variable va contenir tout les rapports du visiteur dont la date est la date choisie.

Voici le résultat après avoir choisi la date des rapports du visiteur datant du 2021-10-31 :

The screenshot shows a web application interface. On the left, there is a sidebar with a user profile picture of a man, the name 'LIM Pascal', and the address '5 allée du passeur, 93320, PAVILLONS-SOUS-BOIS'. Below this, it says 'Date de votre embauche : 2021-10-11'. The sidebar also has navigation links: 'Accueil', 'Rapports' (selected), 'Création d'un rapport', 'Consultation des rapports', 'Médecins' (with a plus sign), and 'Se déconnecter'. The main content area has a title 'Liste de vos rapport datant du 2021-10-31.' Below this, there is a table with the following data:

ID	Date du rapport	Motif du rapport	Bilan du rapport	Id du médecin	Id du médicament	Quantité offerte
1673	2021-10-31	Réparation dentaire	2 séances à venir	324	Aucun médicament proposés	
1674	2021-10-31	Réparation dentaire	2 séances à venir	324	AMOX45	25
1675	2021-10-31	Réparation dentaire	2 séances à venir	324	AMOX45	25

# Médecins :

## Recherche par début du nom :

La recherche de médecin se fait par le début du nom du médecin.

Voici le résultat d'une recherche pour les médecins dont le nom commence par 'ar' :

The screenshot shows a user interface for managing doctors. On the left, there's a sidebar with a profile picture of a doctor, the name 'ARIBI Alain', address '8 rue des Charmes 46000, CAHORS', and hire date 'Date de votre embauche : 2005-12-21'. Below this are links for 'Accueil', 'Rapports' (with 'Création d'un rapport' and 'Consultation des rapports'), 'Médecins' (with 'Consultation des médecins'), and 'Se déconnecter'. The main area is titled 'Liste des médecins :'. It features a search bar with placeholder text 'dont le nom commence par 'ar''. Below it, three doctor profiles are displayed in cards:

- Dr. ARNAUD Andrew**  
ID : 8  
Adresse : 29 rue des Pigeons SAVINES-LE-LAC 05160  
Département : 5  
Téléphone : 0477740994  
Spécialité complémentaire : Aucune | Appeler le médecin
- Dr. ARNOUX Mohammed**  
ID : 125  
Adresse : 33 rue Bonaparte VAUX-CHAMPAGNE 08130  
Département : 8  
Téléphone : 0388412366  
Spécialité complémentaire : Aucune | Appeler le médecin
- Dr. ARMAND Habib**  
ID : 134  
Adresse : 96 rue de Marigny SAINTE-MARIE 05150  
Département : 5  
Téléphone : 0472203933  
Spécialité complémentaire : Aucune | Appeler le médecin

Le contrôleur qui gère la recherche de médecin est celui-ci :

```
1 <?php
2
3 if ( $_SERVER["SCRIPT_FILENAME"] == __FILE__ ){
4     $racine= "../";
5 }
6
7 include "getRacine.php";
8 include_once "$racine/modele/medecin.php";
9 include_once "$racine/modele/medecinDAO.php";
10
11 if (connexionDAO::isLoggedOn()){
12     $Visiteur= VisiteurDAO::get_visiteurByLogin($_SESSION['login']); // Obligatoire pour que les infos du visiteur sur la navbar soient affichées
13 }
14
15 // Étant donné que les nom des médecins sont tous en majuscules, et que lorsque l'on fait une recherche, on écrit pas forcément tout en majuscule, je transforme le texte du
16 // $_POST en majuscule étant donné que c'est une requête en LIKE
17 $idMedecin= null;
18 if (isset($_POST['nom'])){
19     $idMedecin = strtoupper($_POST['nom']);
20 }
21
22 if ($idMedecin != null){
23
24     $lesMedecins = MedecinDAO::get_medecinByNom($idMedecin);
25
26     $title = "GSB Visites : résultat de votre recherche ";
27
28     $critere = "dont le nom commence par '".$POST['nom']."'";
```

ControleurRechercheMedecin.php

Comme toujours, c'est simplement la variable `$lesMedecins` qui va changer étant donnée que l'on fait appel à la même vue.

Dans ce cas là, la variable `$lesMedecins` va contenir tout les médecins ont le nom commence par ce qui a été écrit.

La fonction qui est appellée pour cette recherche par nom est celle-ci de la classe MédecinDAO :

```
53 static public function get_medecinByNom($unNom){
54     $resultat = array();
55
56     try {
57         $cnx = pdo::connexionPDO();
58         $req = $cnx->prepare("SELECT * FROM medecin WHERE nom LIKE :nom ");
59         $req->bindValue(':nom', $unNom."%", PDO::PARAM_STR);
60
61         $req->execute();
62
63         $ligne = $req->fetchall(PDO::FETCH_ASSOC);
64
65         foreach($ligne as $key => $value){
66             $Medecin = new Medecin($value['id'], $value['nom'], $value['prenom'], $value['adresse'], $value['tel'], $value['specialitecomplementaire'], $value['departement']);
67             $resultat[] = $Medecin;
68         }
69
70     } catch (PDOException $e) {
71         print "Erreur !: " . $e->getMessage();
72         die();
73     }
74
75     return $resultat;
76 }
```

Elle récupère d'abord toutes les informations des médecins dont le nom commence par ce qui a été recherché (c'est ce qui est en paramètre ici) puis toutes les données de médecins récupérées sont stockées dans la variable `$ligne` sous forme d'un tableau associatif.

La fonction crée ensuite un objet de la classe Medecin pour chaque ligne du tableau associatif avec une boucle `foreach()` qui parcourt `$ligne`.

Les objets médecins crée sont ensuite directement insérées après leur création dans une variable `$resultat` qui les contiendra tous.

La variable `$resultat` est ensuite retournée à la fin ce qui permet à la fonction de renvoyer un tableau d'objet de la classe Medecin.

# Tri par critère :

```
1 <?php
2
3 include "getRacine.php";
4 include "modele/medecin.php";
5 include "modele/medecinDAO.php";
6
7 if (connexionDAO::isloggedOn()){
8     $visiteur= VisiteurDAO::get_visiteurByLogin($_SESSION['login']); // Obligatoire pour que les infos du visiteur sur la navbar soient affichées
9 }
10
11 $title = "GSB Visites : liste des médecins";
12
13 $critere = null; // Initialisation de la variable Critere à null car il n'y en a pas par défaut
14
15 //Par défaut, la variable $_GET est nulle, donc elle ne peut pas être assignée à la variable $tri si celle-ci n'existe pas, il faut donc initialiser $tri dès le départ
16 $tri=null;
17 if (!empty($_GET['order'])) {
18     $tri = $_GET['order'];
19 }
20
21 // Le tri se fait ici, dépendant de la valeur du $_GET, $tri va faire appelle à une fonction spécifique pour $lesMedecins ce qui affectera la liste des médecins affichés sur la vue
22 // $critere aura aussi une valeur différente en fonction de $tri
23 if ($tri == "ID"){
24
25     $lesMedecins = MedecinDAO::get_medecinsOrderByID();
26     $critere = "trié par ID.";
27 }
28 elseif($tri == "Specialite"){
29
30     $lesMedecins = MedecinDAO::get_medecinsOrderBySpecialite();
31     $critere = "trié par spécialité.";
32 }
33 elseif($tri == "Departement"){
34
35     $lesMedecins = MedecinDAO::get_medecinsOrderByDepartement();
36     $critere = "trié par département.";
37 }
38 elseif($tri == "AZ"){
39
40     $lesMedecins = MedecinDAO::get_medecinsOrderByNomAZ();
41     $critere = "trié par ordre alphabétique (A - Z).";
42 }
43 elseif($tri == "ZA"){
44
45     $lesMedecins = MedecinDAO::get_medecinsOrderByNomZA();
46     $critere = "trié par ordre alphabétique inversé (Z - A).";
47 }
48
49 include "vue/vueEntete.php";
50 include "$racine/vue/vueNavbar.php";
51 include "$racine/vue/vueMedecin.php";
52 require "$racine/vue/vueFooter.php";
```

## ControleurMedecin.php

Pour faire le tri des médecins par critère, un `$_GET` est utilisé pour récupérer le critère contenu dans le lien lorsque l'on en choisit un. La variable `$tri` est initialisée à `null` par défaut pour que la vue affiche tout les médecins en arrivant sur la vue des médecins, cependant c'est lorsque un critère est choisi que cette variable est assignée au `$_GET['order']`, ce qui va faire changer la variable primordiale de cette vue, `$lesMedecins`.

Dépendant du critère choisi, une fonction différente sera appellée sur `$lesMedecins` ce qui va donc changer son contenu et donc l'affichage sur la vue.

Ce qui va différencier les fonctions est le `order by` dans les requêtes `get_medecins()` de la classe `MedecinDAO` :

```
152     static public function get_medecinsOrderBySpecialite(){
153
154         $resultat=array();
155         try {
156             $cnx = pdo::connexionPDO(); #Connexion à la base de données
157
158             $req = $cnx->prepare('SELECT * FROM medecin ORDER BY specialitecomplementaire ASC'); #Préparation de la requête
159
160             $req->execute(); #Exécution de la requête
161
162             $ligne = $req->fetchall();
163
164             foreach ($ligne as $key => $val) {
165                 $collectionMedecins= new Medecin ($val['id'],$val['nom'],$val['prenom'],$val['adresse'],$val['tel'],
166                 $val['specialitecomplementaire'],$val['departement']);
167                 $resultat[]=$collectionMedecins;
168             }
169
170         } catch (PDOException $e) {
171             print "Erreur !: ". $e->getMessage();
172             die();
173         }
174         return $resultat;
175     }
```

Fonction `get_medecinsOrderBySpecialite()` de la classe `MedecinDAO`

Voici le résultat affichée lorsque le critère de tri choisi est par spécialité :

The screenshot shows a mobile application interface for managing medical staff. On the left, a sidebar displays a doctor's profile (LIM Pascal) and navigation links for Accueil, Rapports, and Médecins. The main content area is titled "Liste des médecins:" and includes a search bar and a sorting option "trié par spécialité." The list displays three doctor entries:

Docteur	ID	Spécialité	Adresse	Département
Dr. MARTIN Prosper	1		France BRIANCON 05100	5
Dr. GARCIA Camille	3		MESSINCOURT 08110	8
Dr. MARTINEZ Alice	4		JOIGNY-SUR-MEUSE 08700	8

Tout ce qui a changé est le contenu de la variable `$lesMedecins`.