

Documentation technique

I. Médicament

1. Connexion à la base de données

Le fichier connexion.inc sert à se connecter au serveur de base de données. Étant donné que nous allons avoir besoin de nous connecter très souvent, ce fichier est donc très utile.

```
<?php
```

```
//connexion au serveur de bases de données en PHP5 avec PDO
// Connect to an ODBC database using driver invocation
$serveur = 'mysql:host=localhost:3308;dbname=gsb_context;charset=utf8';
$login = 'UtilisateurPHP';
$mdp = 'ProjetPHP1';

$conn = new PDO($serveur, $login, $mdp);
// pour déclencher la gestion des erreurs en relation avec la base de données
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
?>
```

Il est accompagné d'un fichier, classConnexion.php, qui contient des méthodes qui permettent de simplifier l'utilisation de la base de données.

```
<?php
/*----- Déclaration de la classe -----*/
class cIstBDD {
/*----- Propriétés de la classe -----*/
var $connexion;
var $dan = "";
/*----- Accès aux propriétés -----*/
function getConnexion() {
return $this->connexion;
}
/*----- Connexion à une base via PDO -----*/
function connecte($pNomDSN, $pUtil, $pPasse) {
//tente d'établir une connexion à une base de données
//connexion à la base de données version PHP5
$this->connexion = new PDO($pNomDSN, $pUtil, $pPasse);
// version ODBC php4
// $this->connexion = odbc_connect( $pNomDSN, $pUtil, $pPasse );
return $this->connexion;
}
/*----- Requetes sur la base -----*/
function requeteAction($req) {
//exécute une requête action (insert, update, delete), ne retourne pas de résultat
// version PDO php5
$nombre_element = $this->connexion->exec($req);
return $nombre_element;
// version PHP4
//odbc_do($this->connexion, $req);
}

function requeteSelect($req) {
//interroge la base (select) et retourne le curseur correspondant
// version PDO php5
$lesenregistrements = $this->connexion->query($req);
// version PHP4
// $lesenregistrements = odbc_do($this->connexion, $req);
return $lesenregistrements;
}

function close() {
// version PDO php5
$this->connexion = null;
// version PHP4
//odbc_close($this->connexion);
}

function requeteP($req) {
$lesenregistrements = $this->connexion->prepare($req);
return $lesenregistrements;
}
?>
```

2 Liste déroulante

De la ligne 159 à la ligne 189 du fichier formMedicament.php il y a la création d'une liste déroulante avec les médicaments qui est remplie grâce à des requêtes sql sur la base de données. Sur cette liste déroulante on a un évènement onchange(ligne 165) qui est un évènement qui prend place quand on sélectionne un autre élément de la liste déroulante .

```
157
158 //appel du fichier de connexion à la base de données
159 include("../classConnexion/connexion.inc");
160
161 // remplir la liste déroulante des médicaments
162 echo '<span>Liste des médicaments : </span><br />';
163
164 // gestion de l'évènement sur la liste deroulante Médicaments
165 echo '<select id="lstMed" onchange="javascript:envoyerRequeteMed(this.value)">';
166
167 //Recupère tout les champs de la table médicament (et tri les médicament par ordre alphabétique)
168 $sql = 'select distinct * from MEDICAMENT order by MED_NOMCOMMERCIAL';
169 $leslignes = $conn->query($sql);
170 $ligne = $leslignes->fetch();
171
172 // parcourt la table des médicaments et remplit la liste déroulante
173 if ($ligne != null) {
174     $nomMED = $ligne['MED_NOMCOMMERCIAL'];
175
176     echo '<option selected value=' . $ligne['MED_DEPOTLEGAL'] . '>' . $ligne['MED_NOMCOMMERCIAL'];
177     $ligne = $leslignes->fetch();
178
179     while ($ligne != null) {
180         echo '<option value=' . $ligne['MED_DEPOTLEGAL'] . '>' . $ligne['MED_NOMCOMMERCIAL'];
181         $ligne = $leslignes->fetch();
182     }
183 } else {
184     $nomMED = null;
185 }
186 echo '</select>';
187
188 echo '<br />';
189 echo '<br />';
190
```

Cet évènement nous renvoie à une méthode qui utilise l'AJAX (Asynchronous JavaScript + XML), qui est une technique permettant de rafraîchir certains champs de la page sans avoir à tout rafraîchir.

```
function envoyerRequeteMed(id)
{
    var requeteHttp = getRequeteHttp();
    if (requeteHttp === null)
    {
        alert("Impossible d'utiliser Ajax sur ce navigateur");
    } else
    {
        // declenche un post sur la page getinfoclasse.php puis declenchera recevoirInfoMed
        requeteHttp.open('POST', '../getInfo/getInfoMed.php', true);
        requeteHttp.onreadystatechange = function () {
            recevoirInfoMed(requeteHttp);
        };
        requeteHttp.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
        requeteHttp.send('MED_DEPOTLEGAL=' + escape(id));
    }
}
```

La méthode envoyerRequeteMed(id) est une méthode qui permet d'envoyer à la page « getInfoMed.php » l'id du médicament qui est sélectionné dans la liste déroulante en \$POST , et dans la page « getInfoMed.php »

3. Le fichier getInfoMed.php

Dans ce fichier on utilise l'id du médicament sélectionné dans la liste déroulante récupérée grâce à la méthode `envoyerRequeteMed` de la page `formPraticien.php` en `$POST` pour qu'à chaque événement on change sur la liste déroulante on génère dynamiquement des champs HTML contenant les informations du médicament sélectionné dans la liste déroulante (via l'id) dans un tableau associatif et grâce à ce procédé on évite de devoir rafraîchir toute la page à chaque changement dans la liste déroulante.

```
try {
    //appel du fichier de connexion à la base de données
    include("../classConnexion/connexion.inc");
    $medDEPOT = filter_input(INPUT_POST, 'MED_DEPOTLEGAL');
    $sql = 'select * from MEDICAMENT where MED_DEPOTLEGAL = ' . "'" . $medDEPOT . "'";
    $lesMED = $conn->query($sql);
    $unMED = $lesMED->fetch();

    //Affichage des informations du médicament sélectionné
    echo '
    <br />
    <label class="titre">DEPOT LEGAL :</label><input type="text" value="' . $unMED["MED_DEPOTLEGAL"] . '" size="10" name="MED_DEPOTLEGAL" class="zone" readonly />
    <br />
    <label class="titre">FAMILLE :</label><input type="text" value="' . $unMED["FAM_CODE"] . '" size="10" name="FAM_CODE" class="zone" readonly />
    <br />
    <label class="titre">COMPOSITION :</label><input type="text" value="' . $unMED["MED_COMPOSITION"] . '" size="66" name="MED_COMPOSITION" class="zone" readonly />
    <br />
    <label class="titre">EFFETS :</label><textarea rows="5" cols="50" size="10" name="MED_EFFETS" class="zone" readonly >' . $unMED["MED_EFFETS"] . '</textarea>
    <br />
    <label class="titre">CONTRE INDICATIONS:</label><textarea rows="5" cols="50" size="10" name="MED_CONTREINDIC" class="zone" readonly >' . $unMED["MED_CONTREINDIC"] . '</i>
    <br />
    <label class="titre">PRIX ECHANTILLON :</label><input type="text" value="' . $unMED["MED_PRIXECHANTILLON"] . '" size="10" name="MED_PRIXECHANTILLON" class="zone" read:
    echo ' <label class="titre">&nbsp;</label><div class="zone"><input type="button" value="<" onClick="precedent();"></input><input type="button" value=">" onClick="suivant();"></i>
    ' ;

    // fermeture de la connexion à Mysql
    $conn = null;
} // fin Try
catch (Exception $e) {
    //echo "erreur";
    //throw new Exception('Erreur Exception declenchee');
    die();
} // fin catch
```

Pour finir, dans la page `formMédicament` ces informations sont appelées grâce à un formulaire qui inclus le fichier `getinfomed.php`.

4. Fonction precedent et suivant

Les fonction JavaScript `precedent()` et `suivant()` permettent de défiler dans la liste déroulante

```
85  function precedent() {
86      //Récupère le numéro de l'index selectionné
87      var numMed = document.getElementById("lstMed").selectedIndex;
88      //Retire -1 pour avoir l'indice du precedent
89      numMed = numMed - 1;
90
91      //Recupère la valeur de id du precedent
92      var numMedprecedent = document.getElementById("lstMed").options[numMed].value;
93
94      //precedent que si c'est possible
95      if (numMed >= 0) {
96          //Met a jour les informations dans la page
97          envoyerRequeteMed(numMedprecedent);
98          //Met a jour le nom dans la liste déroulante
99          document.getElementById("lstMed").value = numMedprecedent;
100      }
101
102  }
```

```

105 |
106 | //Récupère la taille de la liste
107 | var taille = document.getElementById("lstMed").length;
108 |
109 | //Récupère le numéro de l'index sélectionné
110 | var numMed = document.getElementById("lstMed").selectedIndex;
111 | //ajoute +1 pour avoir l'indice du suivant
112 | numMed = numMed + 1;
113 |
114 | //Recupère la valeur de id du precedent
115 | var numMedsuivant = document.getElementById("lstMed").options[numMed].value;
116 |
117 | //suivant que si c'est possible
118 | if (numMed < taille) {
119 |     //Met a jour les informations dans la page
120 |     envoyerRequeteMed(numMedsuivant);
121 |     //Met a jour le nom dans la liste déroulante
122 |     document.getElementById("lstMed").value = numMedsuivant;
123 | }
124 |
125 |

```

ces deux méthodes procèdent de la même manière : elles vont chercher la valeur de l'index actuel de la liste déroulante stocké dans la variable numMed, elles ajoutent ou enlèvent la valeur 1 a cet index et la stocke , puis grâce à cet index on va pouvoir retrouver la valeur de celle ci. Pour finir les deux fonctions utilisent la fonction envoyerRequeteMed() avec en paramètre la valeur trouvée La seule différence est la gestion des erreurs, dans la fonction précédent() on va juste vérifier si numMed est bien supérieur a 0, et dans la fonction suivant() on va chercher à ne pas dépasser la longueur de la liste déroulante.

5. Test boites noires

N°	Chaine/programme	description	attendu	constate	resultat
1	Liste déroulante	Liste déroulante où tous les médicaments disponible devront apparaître	On s'attend a avoir bien tous les médicaments inscrit dans la liste déroulante	Aucune erreur, on constate que tous les médicaments apparaissent	OK
2	Afficher information	Après sélection d'un médicament ses informations devront apparaître	On s'attend a avoir les informations du médicament sélectionné	Aucune erreur, on constate que les informations de chaque médicament apparaissent	OK
3	Bouton suivant	Bouton permettant de voir les informations du médicament suivant	On s'attend à avoir les informations du médicament suivant en cliquant sur le bouton, et a ce que cela s'arrête lorsqu'on arrive au dernier médicament	Aucune erreur, on constate que le bouton fonctionne	OK
4	Bouton précédent	Bouton permettant de voir les informations du médicament précédent	On s'attend à avoir les informations du médicament précédent en cliquant sur le bouton, et a ce que cela s'arrête lorsqu'on arrive au premier médicament	Aucune erreur, on constate que le bouton fonctionne	OK

6. Tests boites blanches

N°	Chaîne/programme	description	attendu	constate	résultat
1	Accès aux serveur de base de donnée	Cet accès se fait grâce au fichier connexion.inc.	On s'attend a pouvoir accéder a la base de données sur phpmyadmin	Aucune erreur, on constate que l'accès a la base de donnée se fait bien	OK
2	Récupération des champs de la table MEDICAMENT	L'accès a ces champs se fait par une requête SQL préparés.On utilise la methode requetePreprare() et execute() et fetch() ou fetchAll() .On compare les résultats avec PhpMyAdmin.	On s'attend a recevoir les champs valide	On constate que les valeurs sont reçus	OK
3	Fonction utilisant AJAX (envoyerRequeteMed)	Fonction utilisant l'architecture AJAX	On s'attend à ce que les valeurs modifiées grâce a cette méthode nous mène pas à un rafraîchissement complet de la page	On constate que les valeurs ont changé sans avoir rechargé la page	OK
4	Chargement de la liste déroulante	La liste déroulante est remplie dès le chargement de la page par le nom commercial du médicament de la table MEDICAMENT	On s'attend à ce que la liste déroulante soit remplie pour chaque option par le nom commercial du médicament MEDICAMENT	On constate que la liste est correctement remplie	OK
5	Affichage informations médicaments	Afficher les informations de la table MEDICAMENT	On s'attend a un affichage de toutes les informations du médicaments	On constate que informations attendues sont affichées	OK

II. PRATICIENS

Fonctionnalité des Praticiens permettant d'afficher les informations enregistrées de chaque Praticien.

1. Liste déroulante Praticiens

On va créer une liste déroulante permettant de lister les différents praticiens de la base GSB.

1.1. Connexion a la base

On se connecte a la base a l'aide d'un fichier externe qui permet de centraliser les connexions, représentées par la variable \$conn à la ligne 107 du fichier « formPraticien.php ». C'est une variable du fichier externe connexion.inc, on peut l'utiliser grâce à l'include fait à la ligne 98. On ajoutera tout le code de la liste dans un bloc try pour pouvoir gérer les exceptions éventuelles.

La ligne 98 est la ligne où se situe l'include faisant le lien avec le fichier connexion.inc. Permettant alors l'utilisation de la variable \$conn dans formPraticien.php :

```
98 | | | include("../classConnexion/connexion.inc");
```

la ligne 107 où est utilisée la variable \$conn dans formPraticien.php. C'est ici que où est faite une connexion à la base de données :

```
107 | | | $leslignes = $conn->query($sql);
```

1.2 Appel des informations depuis la requête SQL

La récupération des champs se traduit par l'utilisation d'une requête SQL dans le fichier formPraticien.php récupérant :

-le numéro

-le nom

-et le prénom

des praticiens. Ces informations seront dans l'ordre alphabétique .

```
106 | | | $sql = 'select distinct* from PRATICIEN order by PRA_NOM,PRA_PRENOM';
107 | | | $leslignes = $conn->query($sql);
108 | | | $ligne = $leslignes->fetch();
```

1.3 Construction de la liste déroulante

Pour construire la liste, on remplit les options pour chaque ligne de la requête.

```
106 $sql = 'select distinct* from PRATICIEN order by PRA_NUM,PRA_PRENOM';
107 $leslignes = $conn->query($sql);
108 $ligne = $leslignes->fetch();
109
110 // parcours la table des praticiens et remplit la liste déroulante
111 if ($ligne != null) {
112
113     /*echo '<option selected value=' . $ligne['PRA_NUM'] . '>' . $ligne['PRA_NUM'] . " " . $ligne['PRA_PRENOM'];
114     $ligne = $leslignes->fetch();*/
115
116     while ($ligne != null) {
117         echo '<option value=' . $ligne['PRA_NUM'] . '>' . $ligne['PRA_NUM'] . " " . $ligne['PRA_PRENOM'];
118         $ligne = $leslignes->fetch();
119     }
120 }
121 echo '</select>';
122 echo '<br />';
123
124 echo '<br />';
```

1.4 Gestion de l'exception du chargement

Si il y a une erreur dans le chargement de la liste on déclenchera une exception permettant de ne pas exécuter le code suivant et de gérer l'erreur.

```
129 catch (Exception $e) {
130     echo "erreur";
131     //throw new Exception('Erreur Exception declenchee');;
132     die();
133 }// fin catch
```

1.5 Évènement OnClick

Une fois la liste sélectionnée on déclenchera l'évènement Onclick qui appelle la fonction chercher avec la valeur (le numéro du praticien sélectionné).

```
104 echo '<select name="lstPrat" class="titre" onClick="chercher(this.value);">';
```

2 Fonction chercher

La méthode chercher utilise l'architecture Ajax(Asynchronous javascript+XML)qui permet de modifier l'information affichée dans le navigateur de manière plus fluide et dynamique à l'aide de Javascript et DOM(Document Object Model)tout en respectant la structure.

```
14 <script language = "javascript">
15     function chercher($pNumero) {
16         var xhr object = null;
17         if (window.XMLHttpRequest) // Firefox
18             xhr_object = new XMLHttpRequest();
19         else if (window.ActiveXObject) // Internet Explorer
20             xhr_object = new ActiveXObject("Microsoft.XMLHTTP");
21         else { // XMLHttpRequest non supporté par le navigateur
22             alert("Votre navigateur ne supporte pas les objets XMLHttpRequest...");
23             return;
24         }
25         //traitement de la réception des données
26         xhr_object.onreadystatechange = function () {
27             if (xhr_object.readyState === 4 && xhr_object.status === 200) {
28                 var formulaire = document.getElementById("formPraticien");
29                 formulaire.innerHTML = xhr_object.responseText;
30             }
31         };
32         //communication vers le serveur
33         xhr_object.open("POST", "../getInfo/getInfoPrat.php", true);
34         xhr_object.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
35         var data = "pratNum=" + $pNumero;
36         xhr_object.send(data);
37     }
38 }
39
```

Le but étant de mettre le paramètre choisi \$pNumero, dans le tableau \$POST pour pouvoir traiter l'information plus tard en l'envoyant en \$POST à la page getInfoPrat.php

3 getInfoPrat.php

La variable \$POST['pratnum'] envoyée depuis la page formPraticien. Sert ici à pouvoir réaliser des requêtes sql en fonction du praticien sélectionné dans la liste déroulante.

```
14 try {
15     //appel du fichier de connexion à la base de données
16     include("../classConnexion/connexion.inc");
17
18     $sql = 'select * from PRATICIEN where PRA_NUM=' . "'" . $_POST['pratNum'] . "'";
19     $lesPrat = $conn->query($sql);
20     $unPrat = $lesPrat->fetch();
21     $sql_TYP_CODE = 'select * from TYPE_PRATICIEN where TYP_CODE=' . "'" . $unPrat["TYP_CODE"] . "'";
22     $code_prat = $conn->query($sql_TYP_CODE);
23     $un_typ_prat = $code_prat->fetch();

```

On va exécuter plusieurs requêtes pour pouvoir récupérer tout les champs associés à la table Praticien.

Après la récupération des données on va utiliser les informations du praticien qui à pour numéro dans la base de données la variable \$POST['pratnum'] correspondant au numéro du praticien sélectionné dans la liste déroulante pour remplir les champs HTML.

3.1 Affichage des champs

```
25 echo '  
26 <label class="titre">NUMERO :</label><label name="PRA_NUM"><mark>'.SunPrat["PRA_NUM"].'</mark></label>  
27 <br />  
28 <br />  
29 <label class="titre">NOM :</label><label name="PRA_NOM"><mark>'.SunPrat["PRA_NOM"].'</mark></label>  
30 <br />  
31 <br />  
32 <label class="titre">PRENOM :</label><label name="PRA_PRENOM"><mark>'.SunPrat["PRA_PRENOM"].'</mark></label>  
33 <br />  
34 <br />  
35 <label class="titre">ADRESSE :</label><label name="PRA_ADRESSE"><mark>'.SunPrat["PRA_ADRESSE"].'</mark></label>  
36 <br />  
37 <br />  
38 <label class="titre">CP :</label><label name="PRA_CP"><mark>'.SunPrat["PRA_CP"].' ' .SunPrat["PRA_VILLE"].'</mark></label>  
39 <br />  
40 <br />  
41 <label class="titre">COEFF. NOTORIETE :</label><label name="PRA_COEFNOTORIETE"><mark>'.SunPrat["PRA_COEFNOTORIETE"].'</mark></label>  
42 <br />  
43 <br />  
44 <label class="titre">TYPE :</label><label name="TYP_CODE"><mark>'.Sun_typ_prat["TYP_LIBELLE"].'</mark></label>  
45 <br />  
46 <br />  
47 <label class="titre">&nbsp;</label><div class="zone"><input type="button" value="<" onClick="precedent();"></input><input type="button" value=">" onClick='
```

Avec les informations du praticiens sélectionné nous allons générer des champs HTML pour chaque changements qui est effectué dans la liste déroulante, ainsi, pour chaque élément de la liste déroulante, les champs HTML correspondants aux informations du praticien seront changés d'une manière dynamique d'où l'avantage d'utiliser l'AJAX.

3.4 Suivant et Précédent

Deux boutons seront présent pour passer du praticien Suivant et Précédent. Un événement sur chaque bouton déclenche, respectivement ,la fonction suivant() et precedent().

Les deux fonctions récupère l'index de la liste déroulante sélectionné et l'incrémante pour le bouton Suivant ou le décrémante pour le Précédent. Ensuite, on récupère la valeur du nouvel index pour ensuite le traiter avec la fonction chercher.

```
41 function suivant() {  
42     var numeroPraticien;  
43     var listeLength;  
44     var numPratSuivant;  
45     listeLength = document.getElementsByName("lstPrat")[0].length;  
46     numeroPraticien = parseInt(document.getElementsByName("lstPrat")[0].selectedIndex);  
47     if (numeroPraticien < listeLength) {  
48         numeroPraticien = numeroPraticien + 1;  
49         numPratSuivant = document.getElementsByName("lstPrat")[0].options[numeroPraticien].value;  
50         chercher(numPratSuivant);  
51         document.getElementsByName("lstPrat")[0].value = numPratSuivant;  
52     }  
53 }  
54  
55 // recule d'une valeur de la liste déroulante  
56 function precedent() {  
57     var numeroPraticien;  
58     var numPratPrecedent;  
59     console.log(document.getElementsByName("lstPrat")[0].selectedIndex);  
60     numeroPraticien = parseInt(document.getElementsByName("lstPrat")[0].selectedIndex);  
61     if (0 < numeroPraticien) {  
62         numeroPraticien = numeroPraticien - 1;  
63         numPratPrecedent = document.getElementsByName("lstPrat")[0].options[numeroPraticien].value;  
64         chercher(numPratPrecedent);  
65         document.getElementsByName("lstPrat")[0].value = numPratPrecedent;  
66     }  
67 }  
68
```

4. Tests

4.1 Tests boîte noires

N°	CHAINE/ PROGRAMME	DESCRIPTION	ATTENDU	CONSTATE	RÉSULTAT
1	Liste déroulante	On clique sur une valeur dans la liste déroulante	On s'attend a ce que la valeur soit sélectionné .L'affichage correspond a celle-ci	Aucune erreur, on constate que les changements de la valeur et de l'affichage a changé	OK
2	Bouton Suivant	On clique sur le bouton suivant	La valeur de la liste déroulante passe a la valeur suivante. L'affichage correspond a celle-ci	Aucune erreur, on constate que les changements de la valeur et de l'affichage a changé	OK
3	Bouton Précédent	On clique sur le bouton précédent	La valeur de la liste déroulante passe a la valeur suivante. L'affichage correspond a celle-ci	Aucune erreur, on constate que les changements de la valeur et de l'affichage a changé	OK
4	Bouton Suivant fin de la liste	On clique sur le bouton suivant alors que la valeur est la dernière de la liste	La valeur de la liste déroulante ne change pas.	Aucun changement	OK
5	Bouton Précédent fin de la liste	On clique sur le bouton suivant alors que la valeur est la première de la liste	La valeur de la liste déroulante ne change pas.	Aucun changement	OK

4.2 Tests boîte blanches

N°	CHAINE/ PROGRAMME	DESCRIPTION	ATTENDU	CONSTATE	RÉSULTAT
1	Accès a la base de données centralisé	L'accès a la base de données se fait toujours sous PDO mais cette fois-ci sur une page sur une page externe qu'on va appeler avec la méthode include().	On s'attend pouvoir se connecter a la base depuis n'importe quelle page	On constate que la base est connecté	OK
2	Récupération des champs de la table PRATICIEN, POSSEDER, SPECIALITE	L'accès a ces champs se fait par différentes requêtes SQL préparés. On utilise la methode requetePreprare() et execute() et fetch() ou fetchAll() . On compare les résultats avec PhpMyAdmin.	On s'attend a recevoir les champs valide	On constate que les valeurs sont reçus	OK
3	Fonction utilisant AJAX (chercher)	Fonction utilisant l'architecture AJAX en envoyant une .	On s'attend à ce que les valeurs modifiés grâce a cette méthode nous mène pas a un rafraîchissement complet de la page	On constate que les valeurs ont changé sans avoir rechargé la page	OK
4	Chargement de la liste déroulante	La liste déroulante est remplie dès le chargement de la page par les noms et prénoms de la table PRATICIEN	La liste déroulante sera remplie pour chaque option par un nom et prénom de la table Praticien par ordre alphabétique	On constate que la liste est correctement remplie	OK
5	Affichage informations praticiens	Les informations de la table PRATICIEN sont affiché par lignes	On s'attend a un affichage du «NUMERO», « NOM», « PRENO M», « ADRESSE», « CP », »CO EFF », « TYPE » ainsi que les champs correspondant en face	On constate que informations attendues sont affichées	OK
6	Tableau d'affichage spécialités	Si le praticien a des informations dans la table POSSEDER alors on voudra les afficher sous forme de tableau	On s'attend a un tableau sous les informations de base. Le tableau à 3 colonnes « DIPLOME», « COEFF PRESCRIPTION(S) »,	On constate que le tableau remplie s'affiche	OK

N°	CHAINE/ PROGRAMME	DESCRIPTION	ATTENDU	CONSTATE	RÉSULTAT
			« SPECIALITE(S) ». Chaque case de chaque aura une valeur attribué		