

DOCUMENTATION TECHNIQUE

GSB : GESTION DES VISITES



Ateliers de professionnalisation 3

ALBARICO Marjorie
BTS SIO SLAM (2ème année)

SOMMAIRE

I.	Le contexte	3
II.	Les fonctionnalités attendues	4
III.	La modélisation des données	5
IV.	Le modèle conceptuel de données	6
V.	L'architecture Modèle-Vue-Contrôleur	7
VI.	L'arborescence de site	8
VII.	Test de site web	9
VII.	Quelques explications	17

I. LE CONTEXTE

Dans le cadre des ateliers de professionnalisation de 2ème année de BTS Services Informatiques aux Organisations option Solutions Logicielles et Applications Métiers, on nous demande de développer un site web pour le laboratoire pharmaceutique Galaxy Swiss Bourdin (GSB).

L'activité commerciale d'un laboratoire pharmaceutique est principalement réalisée par 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.

Le laboratoire GSB désire mettre à disposition des visiteurs médicaux un site web permettant de centraliser les rapports de visite. Ces rapports permettront d'améliorer la démarche de communication des praticiens auprès des médecins pour vendre les produits de l'entreprise.

Pour réaliser ce projet, j'ai utilisé différents langages de programmation tels que :

- **HTML (Hypertext Markup Language)** : utilisé pour créer des pages affichées sur le World Wide Web ou www.
- **CSS (Feuilles de style en cascade)** : décrit la présentation des documents HTML
- **PHP** : est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.
- **JavaScript** : un langage de script orienté objet. On le retrouve principalement dans les pages Internet. Il permet, entre autres, d'introduire sur une page web ou HTML des petites animations ou des effets.
- **jQuery** : une bibliothèque JavaScript gratuite, libre et multiplateforme. La bibliothèque jQuery permet, entre autres, de gagner en rapidité dans l'interaction avec le code HTML d'une page Web. Elle propose comme principales fonctionnalités :
 - la gestion des événements (mouvements de souris, clics, etc.),
 - la création d'effets d'animation,

- la manipulation des feuilles de style en cascade
- et bien d'autres encore.
- **SQL** : un langage de manipulation de données, cela signifie qu'il permet de sélectionner, insérer, modifier ou supprimer des données dans une table d'une base de données relationnelle.

Pour pouvoir écrire et exécuter les codes, j'ai utilisé des outils tels que :

- **XAMPP** : un ensemble de logiciels permettant de mettre en place un serveur Web local
- **Visual Studio Code** : un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions.
- **Navigateur (Safari/Google/etc)**

II. LES FONCTIONNALITÉS ATTENDUES

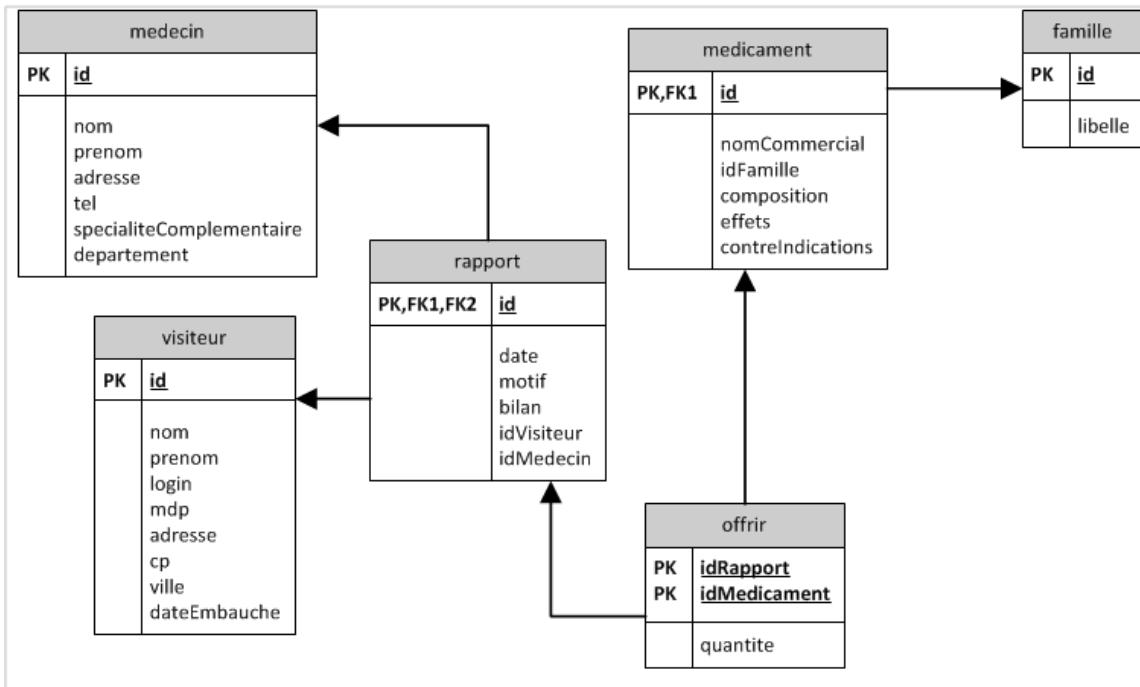
Les fonctionnalités attendues sont :

- a. Identification de l'utilisateur et accès au site web
 - a.1. Connexion
 - a.2. Déconnexion
- b. Gestion des rapports de visite
 - b.1 Création d'un nouveau rapport
 - b.2 Consultation les rapports effectués
 - b.3 Modification d'un rapport
 - b.4 Suppression d'un rapport
- c. Gestion des médecins
 - c.1 Consultation des informations d'un médecin
 - c.2 Consultation des rapports d'un médecin
 - c.3 Modification des informations concernant d'un médecin

III. La modélisation des données

La modélisation des données (*data modeling*) est un processus de description de la structure, des associations, des relations et des contraintes relatives aux données disponibles. Elle sert à établir des normes et à coder des règles de gestion (modèles) des data dans l'organisation.

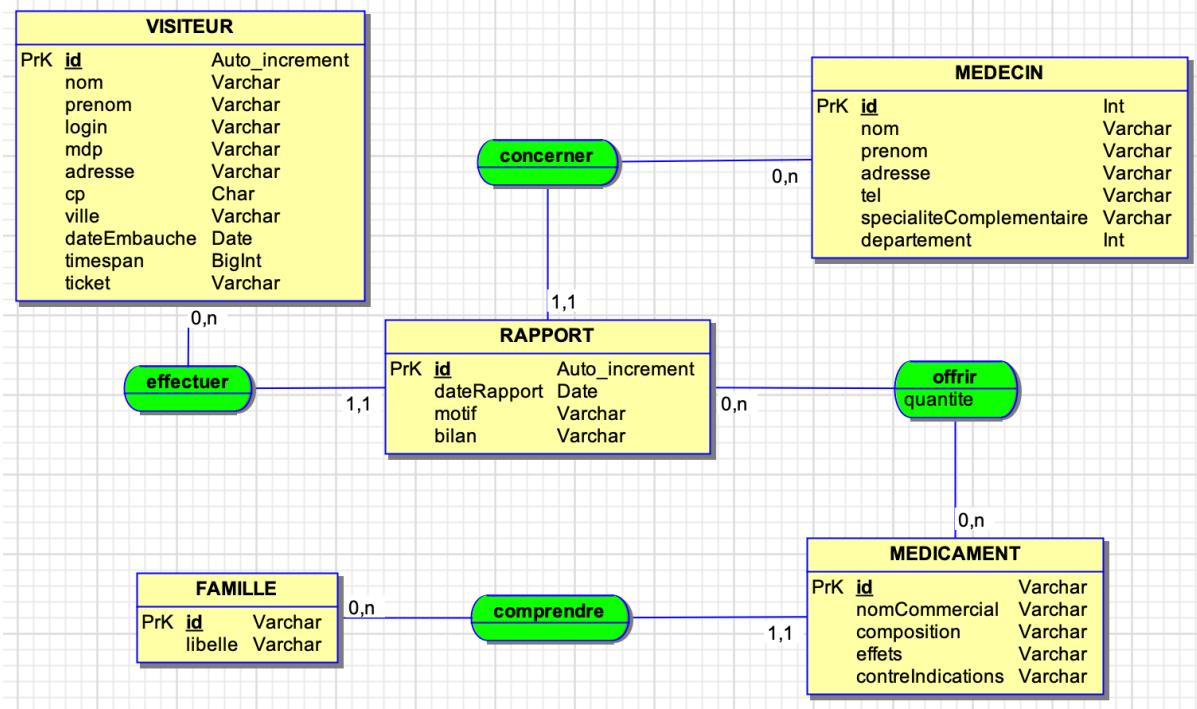
Voici la modélisation complète des données de mon projet :



IV. Le modèle conceptuel de données (MCD)

Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

Voici le MCD de mon projet :



VISITEUR (id, nom, prenom, login, mdp, adresse, cp, ville, dateEmbauche, timespan, ticket)

MEDECIN (id, nom, prenom, adresse, tel, specialiteComplementaire, departement)

RAPPORT (id, dateRapport, motif, bilan, #idVisiteur, #idMedecin)

FAMILLE (id, libelle)

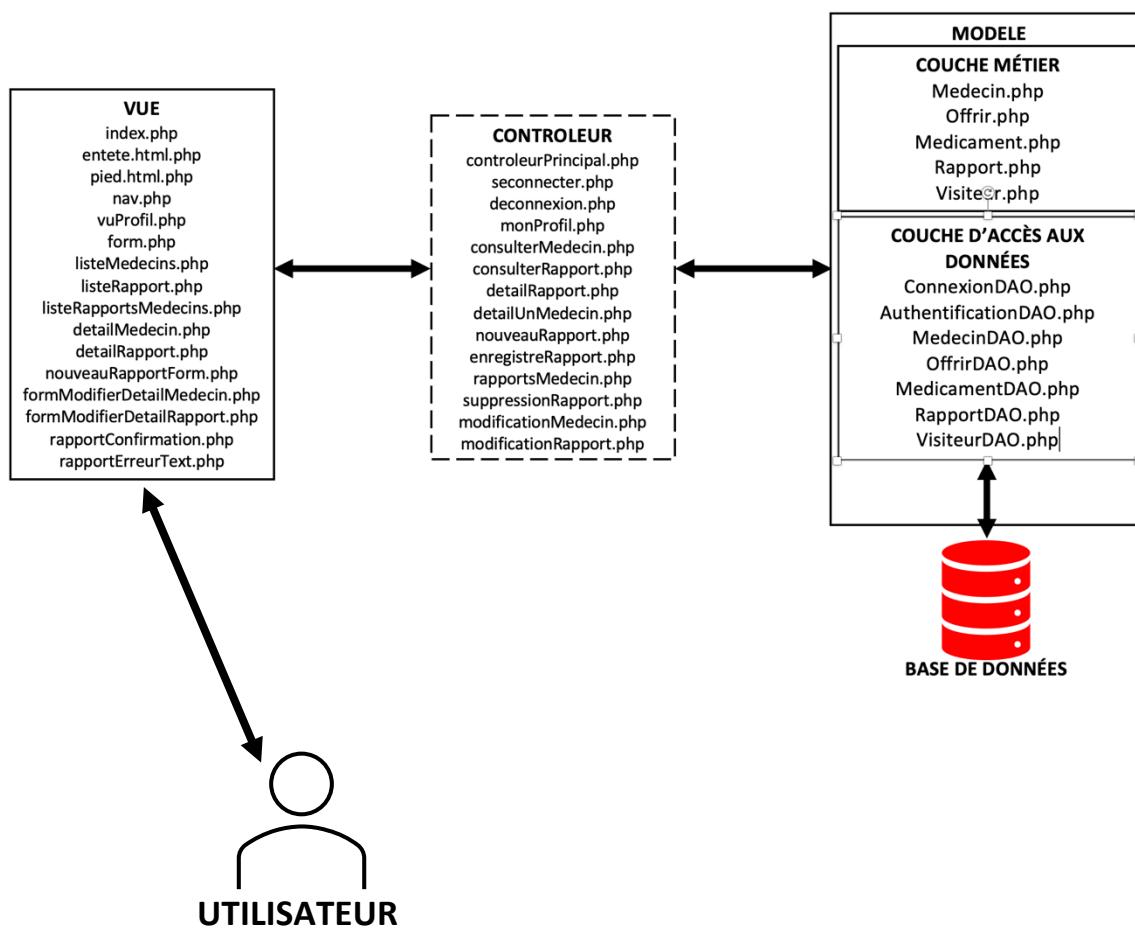
MEDICAMENT (id, nomCommercial, composition, effets, contreindications, #idFamille)

OFFRIR (#idRapport, #idMedicament, quantite)

V. L'architecture Modèle-Vue-Contrôleur (MVC)

Le MVC est un motif de conception (design pattern) qui propose une solution générale au problème de la structuration d'une application. Le MVC définit des règles qui déterminent dans quelle couche de l'architecture, et dans quelle classe (orientée-objet) de cette couche, doit être intégrée une fonctionnalité spécifique. Une application conforme à ces règles est plus facile à comprendre, à gérer et à modifier. Ces règles sont issues d'un processus d'expérimentation et de mise au point de bonnes pratiques qui aboutit à une architecture standard.

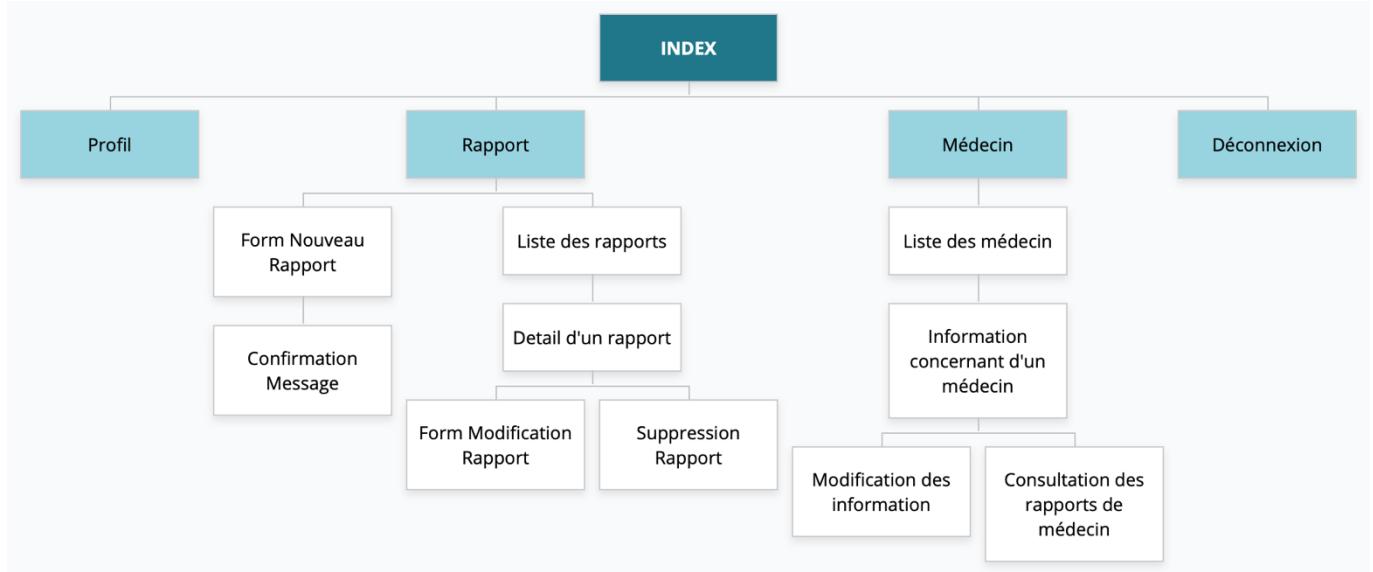
Voici l'architecture MVC de mon projet :



VI. L'arborescence du site

L'arborescence d'un site désigne donc la structure et l'organisation des pages (représentées par leur URL) contenues sur un site Web. Elle offre une vision globale du site Internet en hiérarchisant en mode graphe les différents contenus proposés par le site.

Voici l'arborescence de mon projet site web :



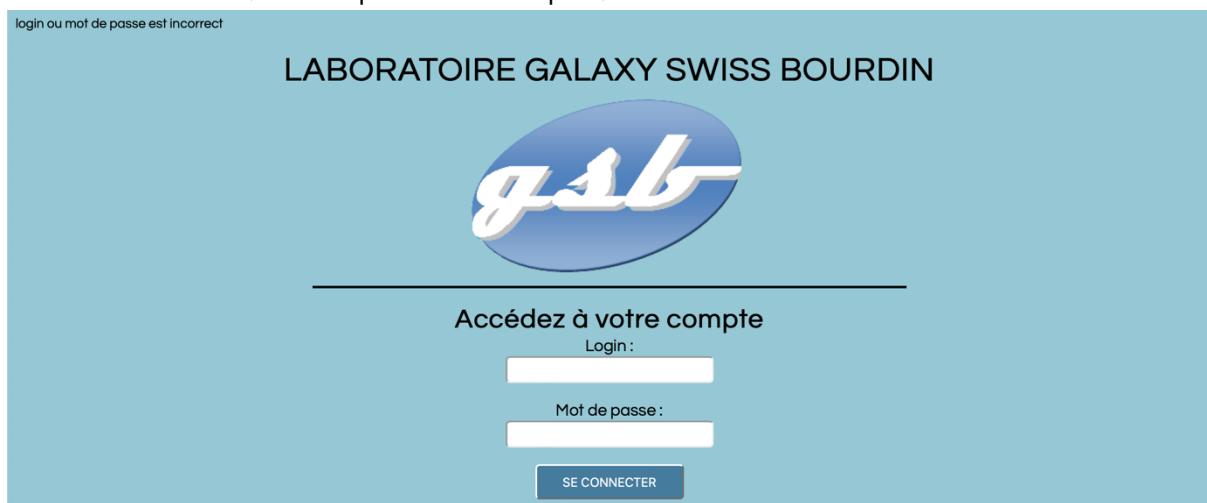
VII. Test de site web

Une fois le site est terminé j'ai effectué des tests afin de vérifier le bon fonctionnement du site et corriger les erreurs éventuelles. J'ai également demandé à quelques camarades de classes de tester le bon fonctionnement du site et recevoir des commentaires sur d'éventuelle amélioration.

INDEX :



en cas d'erreur (le compte n'existe pas) :



si le compte existe, page PROFIL va s'afficher :
(ici, par exemple, c'est le compte d'Alain ARIBI)

The screenshot shows the gsb PROFIL page. At the top left is the gsb logo. To the right are three buttons: "Rapport", "Médecin", and "Déconnexion". Below the header, a message reads "Bienvenue Alain Aribi!". Underneath, a list of actions is provided:

Voici ce que vous pouvez faire :

- Crée un nouveau rapport
- Consulter, modifier et/ou supprimer un rapport que vous avez effectué
 - Consulter un/plusieurs médecin/s
 - Consulter tous les anciens rapports d'un médecin
 - Consulter et/ou modifier les informations concernant un médecin

LA BARRE DE MENU EST COMPOSÉE DE :

The screenshot shows the gsb menu bar. It has three items: "Rapport" (which is highlighted in blue), "Médecin", and "Déconnexion". Below the menu bar, two options are listed:

- Créer un nouveau rapport
- Consulter les rapports

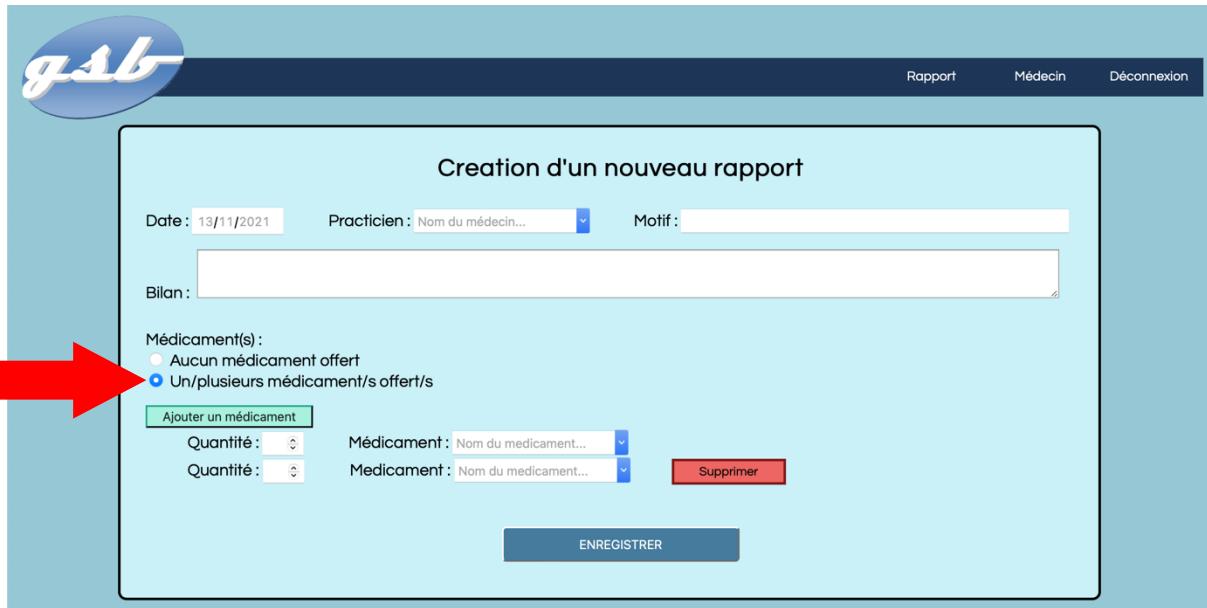
page CRÉATION D'UN NOUVEAU RAPPORT :

The screenshot shows the "Creation d'un nouveau rapport" form. At the top center is the title "Creation d'un nouveau rapport". Below it are three input fields: "Date" (13/11/2021), "Praticien" (Nom du médecin...), and "Motif". There is also a large text area labeled "Bilan". Below these fields is a section for "Médicament(s)":

- Aucun médicament offert
- Un/plusieurs médicament/s offert/s

At the bottom right of the form is a blue button labeled "ENREGISTRER".

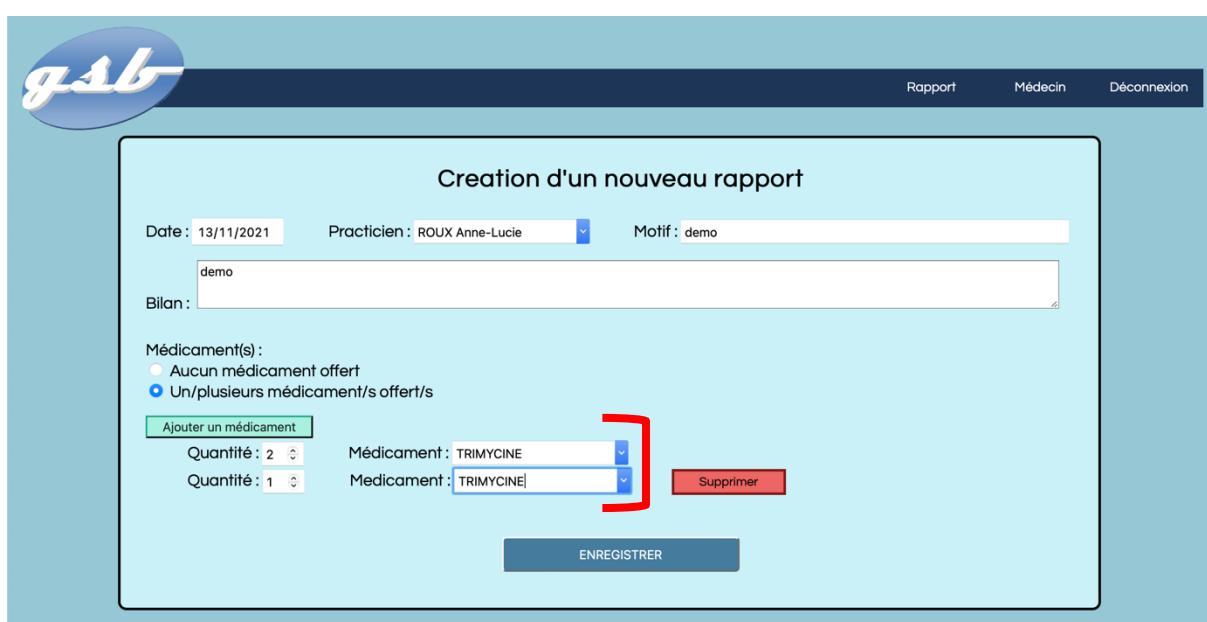
si le rapport contient un ou plusieurs médicament/s offert/s :



The screenshot shows the 'Creation d'un nouveau rapport' (Create a new report) screen. At the top, there are fields for 'Date' (13/11/2021), 'Praticien' (Nom du médecin...), and 'Motif' (Reason). Below these are fields for 'Bilan' (Diagnosis) and 'Médicament(s)' (Medicaments). Two radio buttons are available: 'Aucun médicament offert' (No medicine offered) and 'Un/plusieurs médicament/s offert/s' (One or more medicines offered). The second option is selected. Below this, there is a button 'Ajouter un médicament' (Add a medicine) and two dropdown menus for 'Quantité' (Quantity) and 'Médicament' (Medicine). The first dropdown shows '2' and the second shows 'TRIMYCINE'. A red arrow points to the 'Médicament(s)' section.

en cas d'erreur :

par exemple, ici, on a mal rempli la partie médicament (mêmes médicaments)



The screenshot shows the same 'Creation d'un nouveau rapport' screen. The 'Praticien' field is set to 'ROUX Anne-Lucie' and the 'Motif' field to 'demo'. The 'Bilan' field contains 'demo'. In the 'Médicament(s)' section, the 'Un/plusieurs médicament/s offert/s' radio button is selected. The 'Ajouter un médicament' button is visible. Below it, there are two rows of input fields. The first row has 'Quantité: 2' and 'Médicament: TRIMYCINE'. The second row has 'Quantité: 1' and 'Medicament: TRIMYCINE'. A red bracket groups the 'Médicament' dropdowns of both rows, which both contain 'TRIMYCINE'. A red arrow points to this bracket.

un message d'erreur va s'afficher :

The screenshot shows a web page with a blue header containing the logo 'gsb'. In the top right corner, there are links for 'Rapport', 'Médecin', and 'Déconnexion'. Below the header, a message in French reads 'Veuillez remplir ce formulaire correctement SVP'. A large form box titled 'Creation d'un nouveau rapport' contains fields for 'Date' (set to '19/11/2021'), 'Praticien' (a dropdown menu), 'Motif' (a text input field), and 'Bilan' (a text input field). Below these fields is a section for 'Médicament(s)' with two radio button options: 'Aucun médicament offert' (selected) and 'Un/plusieurs médicament/s offert/s'. At the bottom of the form is a dark blue 'ENREGISTRER' button.

en cas de succès de sauvegarde, un message de confirmation va s'afficher:

The screenshot shows a web page with a blue header containing the logo 'gsb'. In the top right corner, there are links for 'Rapport', 'Médecin', and 'Déconnexion'. Below the header, a message in French reads 'Vous venez d'effectuer un nouveau rapport !'. Underneath the message are two buttons: 'Voir ce rapport' and 'Créer un nouveau rapport'.

Vérification dans la base de données :

table RAPPORT :

	← T →	▼	id	▼	1	date	motif	bilan	idVisiteur	idMedecin
<input type="checkbox"/>				1661		2021-11-13	demo	demo	a131	2

table OFFRIR :

	← T →	▼	idReport	▼	1	idMedicament	quantite
<input type="checkbox"/>				1661		APATOUX22	1
<input type="checkbox"/>				1661		3MYC7	2

PAGE CONSULTATION DES RAPPORTS :

Rechercher un rapport (JJ-MM-AAAA): 13/11/2021

Tous vos rapports

Vous avez effectué 102 rapport(s)

Date (AAAA-MM-JJ): 2016-01-07 Médecin : GATTO, Sophie Voir plus →	Date (AAAA-MM-JJ): 2016-07-03 Médecin : RIBOULET, Bernard Voir plus →	Date (AAAA-MM-JJ): 2016-07-05 Médecin : LANTELME, Rosalie Voir plus →
Date (AAAA-MM-JJ): 2016-07-06 Médecin : HONORAT, Martin Voir plus →	Date (AAAA-MM-JJ): 2016-07-07 Médecin : ESPOSITO, Caline Voir plus →	Date (AAAA-MM-JJ): 2016-07-07 Médecin : MONNIER, Camille Voir plus →
Date (AAAA-MM-JJ): 2016-07-07 Médecin : GARGUILLO, Caline Voir plus →	Date (AAAA-MM-JJ): 2016-07-09 Médecin : TOMASINI, Austine Voir plus →	Date (AAAA-MM-JJ): 2016-07-10 Médecin : PELLET, Aristote Voir plus →

recherche par date :

Rechercher un rapport (JJ-MM-AAAA): 13/11/2021

Tous vos rapports

Vous avez effectué 1 rapport(s) le 2021-11-13

Date (AAAA-MM-JJ): 2021-11-13 Médecin : BLANC, Anne-Lucie Voir plus →

PAGE DETAILS D'UN RAPPORT :

Informations concernant le rapport :

Le 2021-11-13	
Médecin	BLANC, Anne-Lucie
Motif	demo
Bilan	demo
2 médicament/s offert/s:	2 TRIMYCINE 1 APATOUX Vitamine C

[Modifier les informations](#)
[Supprimer ce rapport](#)
[Retourner](#)

PAGE MODIFICATION DES DETAILS D'UN RAPPORT :

The screenshot shows a web application interface for modifying a medical report. At the top, there's a logo for 'gsl' and navigation links for 'Rapport', 'Médecin', and 'Déconnexion'. Below the header, a section titled 'Informations concernant le rapport:' displays the date 'Le 2021-11-13' and details about the report: Médecin (BLANC, Anne-Lucie), Motif (demo), Bilan (demo), and 2 médicament(s) offert(s): (2 TRIMYCINE, 1 APATOUX Vitamine C). Below this, there are three buttons: 'Modifier les informations', 'Supprimer ce rapport', and 'Retourner'. A second section titled 'Modification des informations' contains fields for Date (13/11/2021), Practicien (BLANC Anne-Lucie), Motif (demo), Bilan (demo), and Médicament(s) offert(s): (2 TRIMYCINE, 1 APATOUX Vitamine C). An 'ENREGISTRER' button is at the bottom of this section.

PAGE CONSULTATION DES MÉDECINS :

The screenshot shows a 'Liste des médecins' (List of Doctors) page. At the top, there's a search bar labeled 'Rechercher un médecin: Nom du médecin...' with a 'RECHERCHER' button. Below the search bar, a section titled 'Liste des médecins' displays a grid of twelve doctor profiles, each with a blue circular icon, name, and a 'Voir plus →' button. The profiles are arranged in two rows of six. The names and their corresponding 'Voir plus' links are:

- MARTIN Prosper (Voir plus →)
- BLANC Anne-Lucie (Voir plus →)
- GARCIA Camille (Voir plus →)
- MARTINEZ Alice (Voir plus →)
- MICHEL Vénus (Voir plus →)
- ROUX Anne-Lucie (Voir plus →)
- FABRE Marie (Voir plus →)
- ARNAUD Andrew (Voir plus →)
- FERNANDEZ Julien (Voir plus →)
- LOPEZ Aurèle (Voir plus →)
- BERNARD Johnny (Voir plus →)
- GIRAUD Andrée (Voir plus →)

PAGE CONSULTATION DES INFORMATIONS CONCERNANT UN MÉDECIN :

The screenshot shows a web page titled "Informations concernant le medecin :". At the top, there is a logo for "gsb" and navigation links for "Rapport", "Médecin", and "Déconnexion". Below the title is a table with the following data:

BLANC, Anne-Lucie	
Spécialité complémentaire	Pédiatre
Adresse	39 rue des gatinettes BILLIAT 01200
Téléphone	0356895400
Numéro département	1

Below the table are three buttons: "Modifier les informations", "Voir tous les anciens rapports de visite concernant ce médecin", and "Retourner".

PAGE MODIFICATION DES INFORMATIONS CONCERNANT UN MÉDECIN :

The screenshot shows a web page titled "Informations concernant le medecin :". At the top, there is a logo for "gsb" and navigation links for "Rapport", "Médecin", and "Déconnexion". Below the title is a table with the following data:

BLANC, Anne-Lucie	
Spécialité complémentaire	Pédiatre
Adresse	39 rue des gatinettes BILLIAT 01200
Téléphone	0356895400
Numéro département	1

Below the table are three buttons: "Modifier les informations", "Voir tous les anciens rapports de visite concernant ce médecin", and "Retourner".

Below this section is a larger form titled "Modification des informations". It contains fields for "Nom": "BLANC", "Prenom": "Anne-Lucie", "Téléphone": "0356895400", "Adresse": "39 rue des gatinettes BILLIAT 01200", "Spécialité complémentaire": (empty), "Département": "1", and a "ENREGISTRER" button.

PAGE CONSULTATION DES RAPPORTS D'UN MÉDECIN :

Tous les anciens rapports de visite de BLANC, Anne-Lucie

Il existe 2 rapport(s)

Date : 2017-02-07
Visiteur : Bentot, Pascal
Motif : recommandation de confrère
Bilan : Sur sa réserve
O médicament offert

Date : 2021-11-13
Visiteur : Aribi, Alain
Motif : demo
Bilan : demo
2 médicament/s offert/s:
2 TRIMYCINE
1 APATOUX Vitamine C

Modifier ce rapport

si le visiteur d'un rapport et l'utilisateur est la même personne, un bouton « Modifier ce rapport » va s'afficher et qui va envoyer l'utilisateur à la page *modification des détails d'un rapport.*

VIII. Quelques explications

a. Login

index.php (cf p.9) affiche le formulaire Log in, une fois que l'utilisateur a fini de le remplir, le contrôleur : seconnecter.php va récupérer les valeurs saisies et ces valeurs saisies seront envoyées au modèle : AuthenticationDAO.php qui va vérifier si le compte existe bien dans la base de données. Si le compte existe alors il affiche vuProfil.php sinon, reaffiche le formulaire d'index.php.

extrait du code seconnecter.php :

```
<?php

include_once "./modele/AuthenticationDAO.php";

// recuperation des donnees GET, POST, et SESSION
if (isset($_POST["login"]) && isset($_POST["mdp"])){
    $login=$_POST["login"];
    $mdp=$_POST["mdp"];
}
else
{
    $login="";
    $mdp="";
}

// traitement si necessaire des donnees recuperees
if($login != ""){
    AuthenticationDAO::login($login,$mdp);
}
if (AuthenticationDAO::isLoggedOn()){ // si l'utilisateur est connecté on redirige vers le contrôleur monProfil
    include "./vue/entete.html.php";
    include "./vue/nav.php";
    include "./vue/vuProfil.php";
    include "./vue/pied.html.php";
}
else{ // l'utilisateur n'est pas connecté,
    // appel du script de vue
    $titre = "authentification";
    include "./vue/entete.html.php";
    include "./vue/form.php";
    include "./vue/pied.html.php";
}

?>
```

extrait du code AuthenticationDAO.php :

```
public static function login($login, $mdp) {
    if (!isset($_SESSION)) {
        session_start();
    }
    $util= VisiteurDAO::getVisiteurBylogin($login);
    $mdpBD = $util["mdp"];
    if ($mdpBD == $mdp) {
        // le mot de passe est celui de l'Visiteur dans la base de donnees
        $_SESSION["login"] = $login;
        $_SESSION["mdp"] = $mdpBD;
        $_SESSION["nom"] = $util["nom"];
        $_SESSION["prenom"]=$util["prenom"];
        $_SESSION["id"]=$util["id"];
    }else{
        echo "login ou mot de passe est incorrect";
    }
}

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

    if (isset($_SESSION["login"])){
        $util = VisiteurDAO::getVisiteurBylogin($_SESSION["login"]);
        if ($util["login"] == $_SESSION["login"] && $util["mdp"] == $_SESSION["mdp"]){
            $ret = true;
        }
    }
    return $ret;
}
```

b. Les listes

Pour afficher la liste des médecins, des médicaments, etc, on prendra par exemple la liste des médecins dans la page création d'un nouveau rapport (cf. p10), il faut récupérer la liste des médecins grâce au modèle MédecinDAO.php et stocker le résultat au contrôleur nouveauRapport.php qui va être utilisé pour l'affichage.

extrait du code nouveauRapport.php :

```
// appel des fonctions permettant de recuperer les donnees  
utiles a l'affichage  
$listeMedecins = MedecinDAO::chargeMedecin();  
$listeMedicaments = MedicamentDAO::getMedicaments();  
  
if (AuthentificationDAO::isLoggedOn()){ // si l'utilisateur  
est connecté on redirige vers le contrôleur monProfil  
    include "./vue/entete.html.php";  
    include "./vue/nav.php";  
    include "./vue/nouveauRapportForm.php";  
    include "./vue/pied.html.php";  
}
```

extrait du code html (vue : nouveauRapportForm.php) :

```
<!--MEDECIN-->  
<label for="practicien" class="nouvRapLabel">Practicien :</label>  
<input list="practiciens" name="practicien" id="practicien" class="inputPrac" placeholder="Nom  
du médecin..." required oninvalid="this.setCustomValidity('Veuillez remplir ce champ')"  
oninput="this.setCustomValidity('')"/>  
    <datalist id="practiciens">  
        <?php  
            for ($indexM=0; $indexM<count($listeMedecins); $indexM++){  
                ?>  
                <option value="<?php echo $listeMedecins[$indexM] ->getNom()." ".$listeMedecins  
[$indexM] ->getPrenom()?">">  
            <?php  
            }  
        ?>  
    </datalist>
```

extrait du code

Médecin.DAO.php :

```
public static function chargeMedecin(){  
    try {  
        $cnx = ConnexionDAO::connexionPDO();  
        $req = $cnx->prepare("select * from medecin");  
        $req->execute();  
  
        $ligne = $req->fetch(PDO::FETCH_ASSOC);  
        while ($ligne) {  
            $resultat[] = new Medecin($ligne["id"], $ligne["nom"], $ligne["prenom"], $ligne  
["adresse"], $ligne["tel"], $ligne["specialitecomplementaire"], $ligne["departement"]  
);  
            $ligne = $req->fetch(PDO::FETCH_ASSOC);  
        }  
    } catch (PDOException $e) {  
        print "Erreur !: " . $e->getMessage();  
        die();  
    }  
    return $resultat;  
}
```

L'affichage :



c. Les informations d'un rapport ou médecin

À la page précédente j'ai expliqué comment j'ai affiché la liste des médecins mais on peut voir dans la page consultation des médecins (cf. p15) que dans la liste des médecins, il n'y a pas tous les détails d'un médecin.
extrait du code vue – listeMedecins.php :

```
<div class="nouveauMedForm">
    <form action=". /?action=chercherMedecin" method="POST" id="centerDivMedecin">
        <label for="practicien" class="textFormMed">Rechercher un médecin :</label>
        <input type="text" name="nom" class="labelRMed" placeholder="Nom du médecin..." value="<?php
$nom ?>" />
        <input type="submit" value="RECHERCHER" class="submitRap2">
    </form>
</div>

<h1 class="headlisteMed">Liste des médecins</h1>
<div class="listeMedecinsDiv">

<?php
for ($i = 0; $i < count($listeMedecins); $i++) {
    ?>

        <div class="cardMed">
            
            <div class="nomMedCard"><?php echo $listeMedecins[$i]['nom'] . " <br>" . $listeMedecins[$i]
['prenom'] . "<br>" ; ?>
            </div>
            <?php echo "<a class='buttonVoirPlus' href='/?action=detailUnMedecin&id=" . $listeMedecins
[$i]['id'] . "'>Voir plus </a>"; ?>
        </div>
    <?php
}
?>
</div>
```



L'affichage :



Ici, on peut voir que j'ai rajouté un bouton « Voir plus » à chaque carte de médecin qui permet d'envoyer l'id de médecin pour accéder à la page consultation des informations d'un médecin(cf. p15).

Par exemple, l'utilisateur souhaite regarder les informations de médecin ROUX Anne-Lucie alors il va cliquer sur le bouton « Voir plus », le contrôleur va récupérer l'id de médecin et va appeler une fonction du modèle MedecinDAO.php pour récupérer les informations dans la base de données.

L'affichage :



extrait du code vue listeMedecins.php :

```
<?php echo "<a class='buttonVoirPlus' href='./?action=detailUnMedecin&id=" . $listeMedecins[$i]['id'] . "'>Voir plus </a>"; ?>
```

extrait du code contrôleur detailUnMedecin.php :

```
recuperation des donnees GET, POST, et SESSION
$id = $_GET["id"];

// appel des fonctions permettant de recuperer les donnees utiles a l'affichage
$unMedecin = MedecinDAO::getMedecinById($id);

// appel du script de vue qui permet de generer l'affichage des donnees
$titre = "detail d'un medecin";
include "./vue/entete.html.php";
include "./vue/nav.php";
include "./vue/detailMedecin.php";
include "./vue/pied.html.php";

<h1 class="hOneDetMed">Informations concernant le medecin : </h1>
<table id="tableMedecin">
    <tr>
        <td id="nomTableMed" colspan="2"><?php echo $unMedecin['nom']. ", ". $unMedecin['prenom']?></td>
    </tr>
    <?php if ($unMedecin['specialitecomplementaire']!='') {
        ?>
        <tr>
            <td>Spécialité complémentaire</td>
            <td><?php echo $unMedecin['specialitecomplementaire']?></td>
        </tr>
    <?php
    }
    ?>
    <tr>
        <td>Adresse</td>
        <td><?php echo $unMedecin['adresse']?></td>
    </tr>
    <tr>
        <td>Téléphone</td>
        <td><?php echo $unMedecin['tel']?></td>
    </tr>
    <tr>
        <td>Numéro département</td>
        <td><?php echo $unMedecin['departement']?></td>
    </tr>
</table>
<?php echo "<a class='buttonInfoMed' href='./?action=modifiedMedecin&id=" . $unMedecin['id'] . "'>Modif... </a>" ; ?>
```

extrait du code vue detailMedecin.php :

```
<?php
try {
    $cnx = ConnexionDAO::connexionPDO();
    $req = $cnx->prepare("select * from medecin where id=:id");
    $req->bindValue(':id', $id, PDO::PARAM_INT);

    $req->execute();

    $resultat = $req->fetch(PDO::FETCH_ASSOC);
} catch (PDOException $e) {
    print "Erreur !: " . $e->getMessage();
    die();
}
return $resultat;
}

public static function getMedecinById($id) {
    $resultat = self::getMedecin($id);
    return $resultat;
}
```

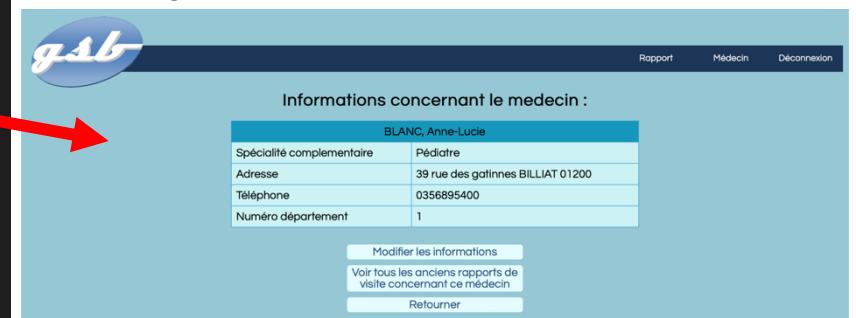
extrait du code modèle MedecinDAO.php :

```
try {
    $cnx = ConnexionDAO::connexionPDO();
    $req = $cnx->prepare("select * from medecin where id=:id");
    $req->bindValue(':id', $id, PDO::PARAM_INT);

    $req->execute();

    $resultat = $req->fetch(PDO::FETCH_ASSOC);
} catch (PDOException $e) {
    print "Erreur !: " . $e->getMessage();
    die();
}
return $resultat;
```

L'affichage :



d. La modification des données

À partir de la page consultation des informations d'un médecin (cf. p15), il y a un bouton « Modifier les informations » qui permet d'afficher un formulaire / la page modifications des informations concernant un médecin (cf. p15). L'utilisateur va remplir que les champs qu'il souhaite modifier. Le contrôleur modificationMedecin.php va récupérer toutes les valeurs saisies et va appeler des fonctions de modèle MedecinDAO.php pour changer les informations dans la base de données.

Par exemple, l'utilisateur veut modifier le nom de médecin alors il va remplir le champ nom, le contrôleur prendra la valeur et va appeler la fonction updateNomMedecinById de modèle MedecinDAO.php et en paramètres sont l'id de médecin et la valeur saisie.

extrait du code contrôleur – modificationMedecin.php :

```
// recuperation des donnees GET, POST, et SESSION
$id = $_GET["id"];

// recuperation des variables du formulaire
$nom=$_POST["nom"];
$prenom=$_POST["prenom"];
$tel=$_POST["tel"];
$specialiteComplementaire=$_POST["specialiteComplementaire"];
$departement=$_POST["departement"];
$adresse=$_POST["adresse"];

// appel des fonctions permettant de recuperer les donnees utiles a l'affichage
$unMedecin = MedecinDAO::getMedecinById($id);

//modifications dans la base de données
if ($nom!=""){
    MedecinDAO::updateNomMedecinById($id, $nom);
    header("Refresh:0");
}
```

extrait du code modèle - MedecinDAO.php :

```
public static function updateNomMedecinById($id, $nom) {
    $cnx = ConnexionDAO::connexionPDO();
    $req = $cnx->prepare("update medecin set nom=upper(:nom) where id=:id");
    $req->bindValue(':nom', $nom, PDO::PARAM_STR);
    $req->bindValue(':id', $id, PDO::PARAM_INT);

    $req->execute();
    return 1;
}
```

le « header("Refresh:0"); » permet de rafraîchir la page courante pour que l'utilisateur peut vérifier les mises à jour qu'il vient de faire.

e. L'enregistrement d'un rapport

On peut voir dans la page création d'un nouveau rapport (cf. p10) qu'il y a un formulaire à remplir (tous les champs) par l'utilisateur.

extrait du code vue – nouveauRapportForm.php :

```
<!--MEDICAMENTS-->
<br>
<p class="nouvRapLabelMedic">Médicament(s) : </p>

<div>
    <label class="nouvRapLabel"><input class="radioclass" type="radio" name="sansMedicCheckbox" value="true" required
        oninvalid="this.setCustomValidity('Veuillez cocher ce champ')" oninput="this.setCustomValidity('')"/>Aucun médicament
        offert</label><br>
    <label class="nouvRapLabel"><input class="radioclass" type="radio" name="sansMedicCheckbox" value="false"/>Un/
        plusieurs médicament/s offert/s</label>
</div>
<hr>
```

L'affichage :

The screenshot shows a web application interface for creating a new report. At the top, there is a logo for 'gsb' and a navigation bar with links for 'Rapport', 'Médecin', and 'Déconnexion'. Below the navigation bar is a title 'Creation d'un nouveau rapport'. The form contains several input fields: 'Date' (set to '17/11/2021'), 'Praticien' (set to 'MARTIN Prosper'), and 'Motif' (set to 'demo'). There is also a text area labeled 'Bilan:' containing the text 'demo'. At the bottom of the form, there is a question 'Veuillez cocher ce champ' followed by a note '(s) médicament(s):'. Two radio buttons are present: one for 'Aucun médicament offert' and another for 'Un/plusieurs médicament/s offert/s'. A red arrow points from the extracted code above to the 'Veuillez cocher ce champ' note in the screenshot.

Une fois que l'utilisateur a fini de remplir le formulaire. Le contrôleur enregistreRapport.php récupère les valeurs saisies et vérifie des conditions pour voir que le formulaire est rempli correctement, chaque condition consiste des différentes fonctions (cas rapport sans médicament et cas rapport avec médicament/s).

RECUPERATION DES VALEURS SAISIES :

extrait du code contrôleur – enregistreRapport.php :

```
// recuperation des variables du formulaire
$date = $_POST['date'];
$practicien=$_POST["practicien"];
$motif = $_POST['motif'];
$bilan = $_POST['bilan'];
$quantite=$_POST["mynumber"];
$medic=$_POST['mytext'];
$sansMedicCheckbox=$_POST['sansMedicCheckbox'];

//separation de $practicien
$nomMed = strtok($practicien, " ");
$prenomMed = substr($practicien, strpos($practicien, " ") + 1);
//recuperation d'id de medecin
$leMedecin = MedecinDAO::getMedecinByNomPrenom($nomMed, $prenomMed);
$idMedecin = $leMedecin['id'];
```

extrait du code modèle – MedecinDAO.php :

```
public static function getMedecinByNomPrenom($nom, $prenom) {
    try {
        $cnx = ConnexionDAO::connexionPDO();
        $req = $cnx->prepare("select * from medecin where nom=:nom and prenom=:prenom");
        $req->bindValue(':nom', $nom, PDO::PARAM_STR);
        $req->bindValue(':prenom', $prenom, PDO::PARAM_STR);

        $req->execute();

        $resultat = $req->fetch(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage();
        die();
    }
    return $resultat;
}
```

Comme j'ai expliqué que pour la liste des médecins, il y a que l'affichage de nom et prénom (cf. p18) donc pour récupérer l'id de médecin, j'ai choisi de séparer le \$practicien en utilisant des fonctions strtok, substr et strpos pour avoir le \$nomMed et \$prenomMed et appelle la fonction getMedecinByNomPrenom (avec \$nomMed et \$prenomMed dans le paramètre) de modèle MedecinDAO.php puis stocke que l'id de médecin.

EXEMPLE :

```
main.php x
1  <?php
2  $practicien = "ROUX Anne-Lucie";
3
4  $nomMed = strtok($practicien, " ");
5  $prenomMed = substr($practicien, strpos($practicien, " ")
+ 1);
6
7  echo "nom: ".$nomMed."\n";
8  echo "prenom: ".$prenomMed;
```

```
Console Shell
> run-project -l php
nom: ROUX
prenom: Anne-Lucie> []
```

Les variables \$quantite et \$medic sont des tableaux. Il faut maintenant regarder si ils sont remplis correctement.

Les cas d'erreurs sont :

- avoir un/des champ/s vide/s (quantité ou médicaments)
- avoir plusieurs mêmes médicaments

Médicament(s):
 Aucun médicament offert
 Un/plusieurs médicament/s offert

Ajouter un médicament

Quantité :	Médicament :
Quantité :	Medicament : Nom du medicament...

Supprimer

Médicament(s):
 Aucun médicament offert
 Un/plusieurs médicament/s offert

Ajouter un médicament

Quantité :	Médicament :
2	TRIMYCINE
1	TRIMYCINE

Supprimer

extrait du code contrôleur - enregisterRapport.php :

```
//verification si tt les qtt et medic sont corrects
$noEmptyQ=false;
$noEmptyM=false;
$medicDuplicates=false;

//verification que tout les quantités = rempli
foreach($quantite as $number){
    if ($number==""){
        $noEmptyQ=true;
    }
}

//verification que tout les medicaments = rempli
foreach($medic as $text){
    if ($text==""){
        $noEmptyM=true;
    }
}

//verification que $medic n'a que des valeurs uniques
if(count(array_unique($medic)) < count($medic)){
    $medicDuplicates=true;
}

//verification que les champs sont remplis correctement
$rapportOK=false;

if (!empty($date) && !empty($practicien) && !empty($motif) && !empty($bilan)) {
    //rapport sans medic
    if($sansMedicCheckbox=="true" && $noEmptyQ==true && $noEmptyM==true){
        // sauvegarde de ce rapport
        RapportDAO::insertNouveauRapport($date, $motif, $bilan, $idVisiteur, $idMedecin);
        $rapportOK=true;
    }elseif ($sansMedicCheckbox=="false" && $noEmptyQ==false && $noEmptyM==false && $medicDuplicates==false && count($medic)==count($quantite)){ //rapport avec medic
        // sauvegarde de ce rapport
        RapportDAO::insertNouveauRapport($date, $motif, $bilan, $idVisiteur, $idMedecin);
        //get id rapport
        $idRapport=RapportDAO::getIdLastRapport($idVisiteur);
        $idRapport=$idRapport['max(id)'];

        for ($k=0; $k<count($medic); $k++){
            $idMedicament=MedicamentDAO::getIdMedicByNomCommercial($medic[$k]); //
            recuperation d'id medicament
            $idMedicament=$idMedicament['id'];
            OffrirDAO::insertOffert($idRapport, $idMedicament, $quantite[$k]);
        }
        $rapportOK=true;
    }else{
        $rapportOK=false;
    }
}
```

Si le rapport est bien enregistré alors un message de confirmation va s'afficher sinon, le message d'erreur va s'afficher.
extrait du code contrôleur – `enregistreRapport.php` :

```
include "./vue/entete.html.php";
include "./vue/nav.php";
if($rapportOK==true){
    include "./vue/rapportConfirmation.php";
} else{
    // appel des fonctions permettant de recuperer les donnees utiles a l'affichage
    $listeMedecins = MedecinDAO::chargeMedecin();
    $listeMedicaments = MedicamentDAO::getMedicaments();
    include "./vue/rapportErreurText.php";
    include "./vue/nouveauRapportForm.php";
}
include "./vue/pied.html.php";
```

L'affichage (message de confirmation) :



L'affichage (message d'erreur) :

