GSB

Brice Matias

Pajot-Trohel Maxime

Fieux Diego

Bts SIO

Sommaire :

1° Présentation du projet

2° Organisation du projet

3° Architecture

4° Les différentes pages du projet

5° Base de donnée

6° Gestion du projet

7° Charte graphique

8° Cahier d’activité

9° Web service

Présentation du projet :

Ce projet est réalisé dans le cadre du BTS SIO et est une commande du laboratoire suisse Galaxy Swiss Bourdin ( GSB ). L’objectif était ici de créer un site web vitrine présentant les différents produits proposés par GSB ainsi que des activités auxquelles les utilisateurs puissent y participer et s’inscrire.

Pour mener à bien ce projet, nous avons reçu un cahier des charges contenant les différentes fonctionnalités nécessaires et les attentes du clients.

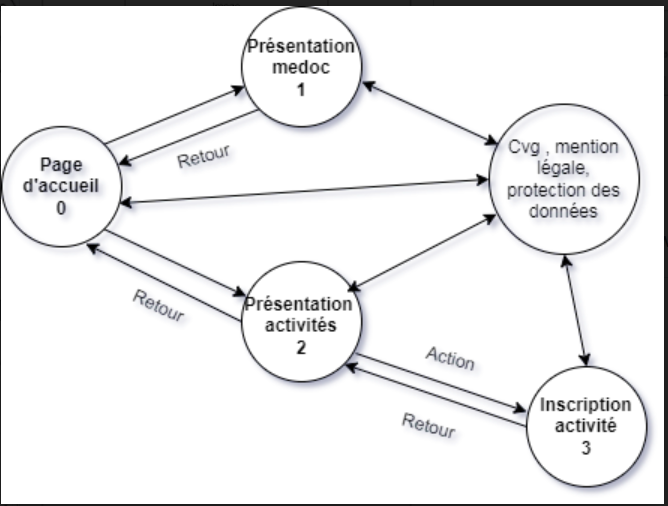
Pour répondre à ces besoins nous avons travaillé en équipe de 3 avec chacun des rôles bien définis.

Organisation du projet

Pour mener à bien ce projet, nous avons eu recours à la méthode agile comme méthode de développement. Ainsi nous avons d’abord défini un rôle pour chacun des membres de l’équipe. C’est alors que Brice s’est vu attribuer le rôle de product owner ( chef de projet ), Maxime Pajot-Trohel Le rôle de Scrum master ( chef de mêlée) et Diego Fieux comme Scrum Member ( membre du projet). Comme le préconise la méthode agile, la première chose que nous faisions les jours de projet était de faire une mêlée afin de mettre en commun les différentes avancées, retard, problèmes rencontrés et de poser des questions. Une fois cela fait, chacun reprend le travail sur le sprint qui lui est attribué. Les sprint sont des parties précise du projet qui sont réalisées et qui reposent sur les backlogs (liste évolutive de fonctionnalité) définie durant la mêlée.

En amont du projet nous avons mis en place l'enchaînement des écrans pour nous répartir les tâches et avoir un avant goût du rendu du site.

Nous avons commencé par la mise en place d’un enchaînement des écrans :

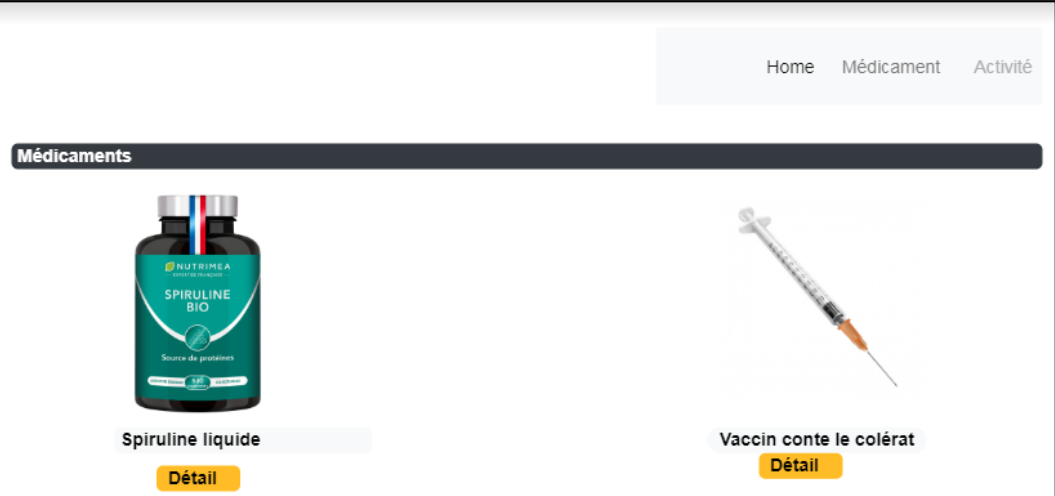


En suite nous nous somme mit un plan du site pour le design :   
Accueil :



Page activités :



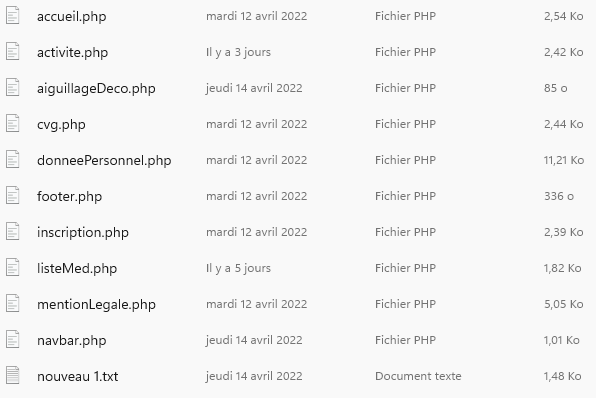
Page médicaments :   
  
  
Cvg :



Architecture

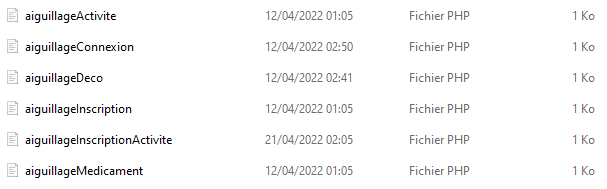
Le site se base sur une architecture mvc, modèle, vue et contrôleur. L’objectif de cette architecture est de séparer le code afin de séparer les différentes fonctionnalités.

En premier, nous avons les vues qui regroupent l’HTML et qui sont pour faire simple les pages qui seront affichées à l’écran.



Ces pages n'interagissent donc pas avec la base de données mais peuvent comporter du php afin d’utiliser les informations contenues dans la base de données récupérées dans les autres compartiments du code.

Ensuite nous avons les contrôleurs qui servent à relier le modèle qui lui interagit avec la base de données avec la vue.

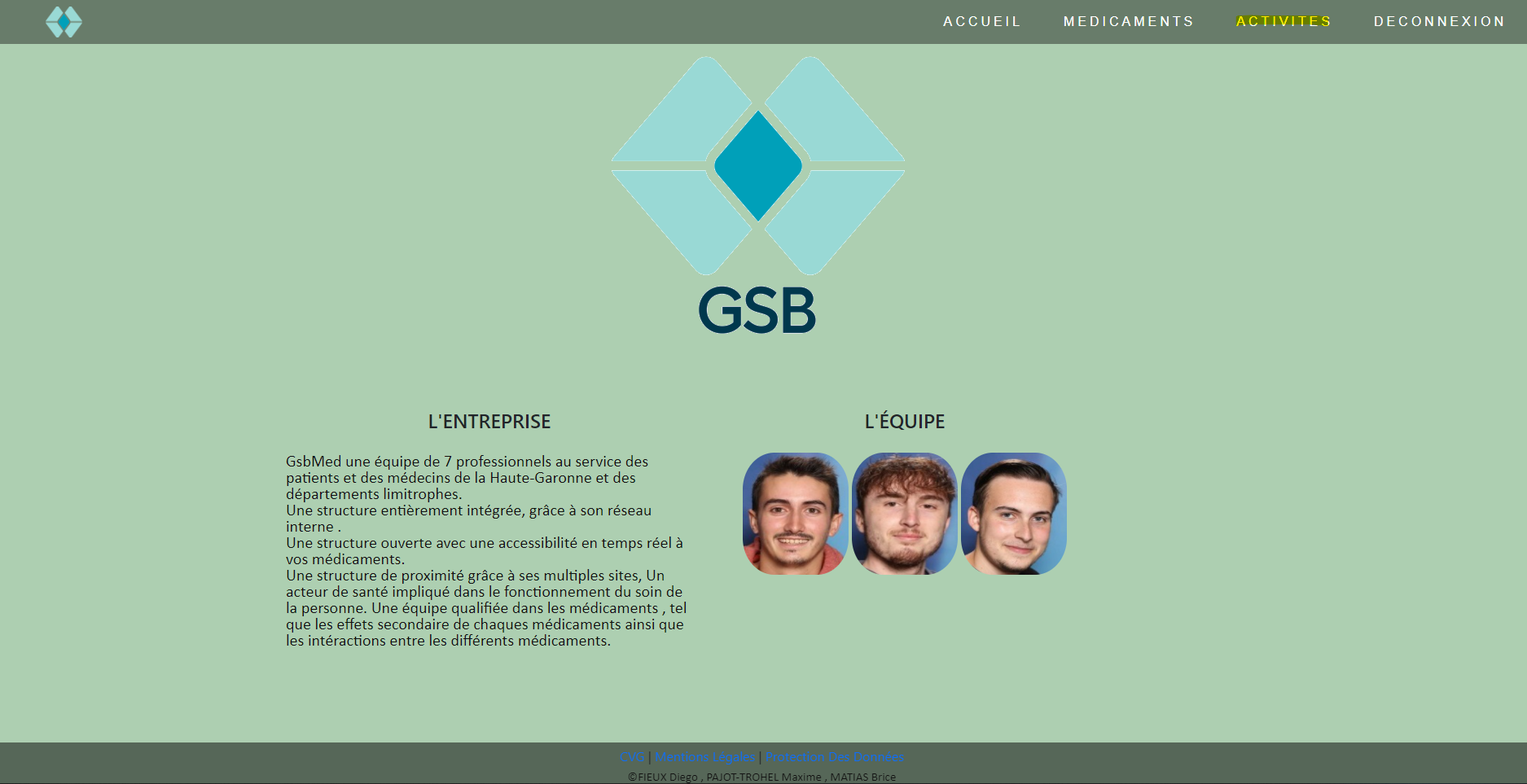


Et enfin nous avons le modèle qui contient toutes les fonctions qui interagissent avec la base de données.

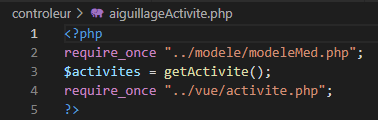


L'intérêt de cette séparation se trouve surtout au niveau de la cyber sécurité. Cela permet d’effectuer un meilleur contrôle sur les intéractions avec la base de données en la rendant presque inaccessible. L’architecture MVC s'arrête ici cependant, nous avions une contrainte supplémentaire imposé qui était de passer par des web services en C#.

Le mieux pour conclure cette architecture est de faire une démonstration et par exemple l’affichage des activités. Au début, nous sommes sur la page d’accueil et avons cliqué sur activité.

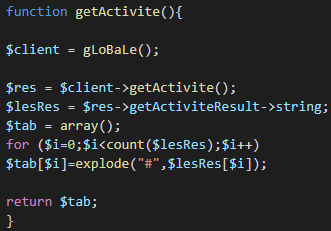
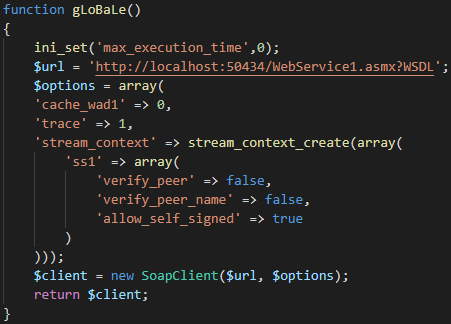


Une fois le clic effectué, le code de la vue accueil.php appelle aiguillageActivites.php ci-dessous.



Comme le montre ce code, on appelle le modèle à la ligne 2 pour utiliser les fonctions qu’il comporte. La fonction ici utilisée est la fonction getActivite() qui retourne un tableau contenant toutes les activités et stocké dans la variable $activites.

Le modèle :



Cette fonction interagit avec le webservice et appelle la fonction getAtivite().

Web Service :



La variable $activites contient donc alors un tableau de toutes les activités et il ne reste plus qu’à les afficher sur la page activité.php sur laquelle l’aiguillage nous a renvoyé.



On crée alors un tableau et dans les lignes de tableau, on affiche les différentes informations contenues dans $activites à l’aide d’une boucle foreach en php. Et voici le résultat:

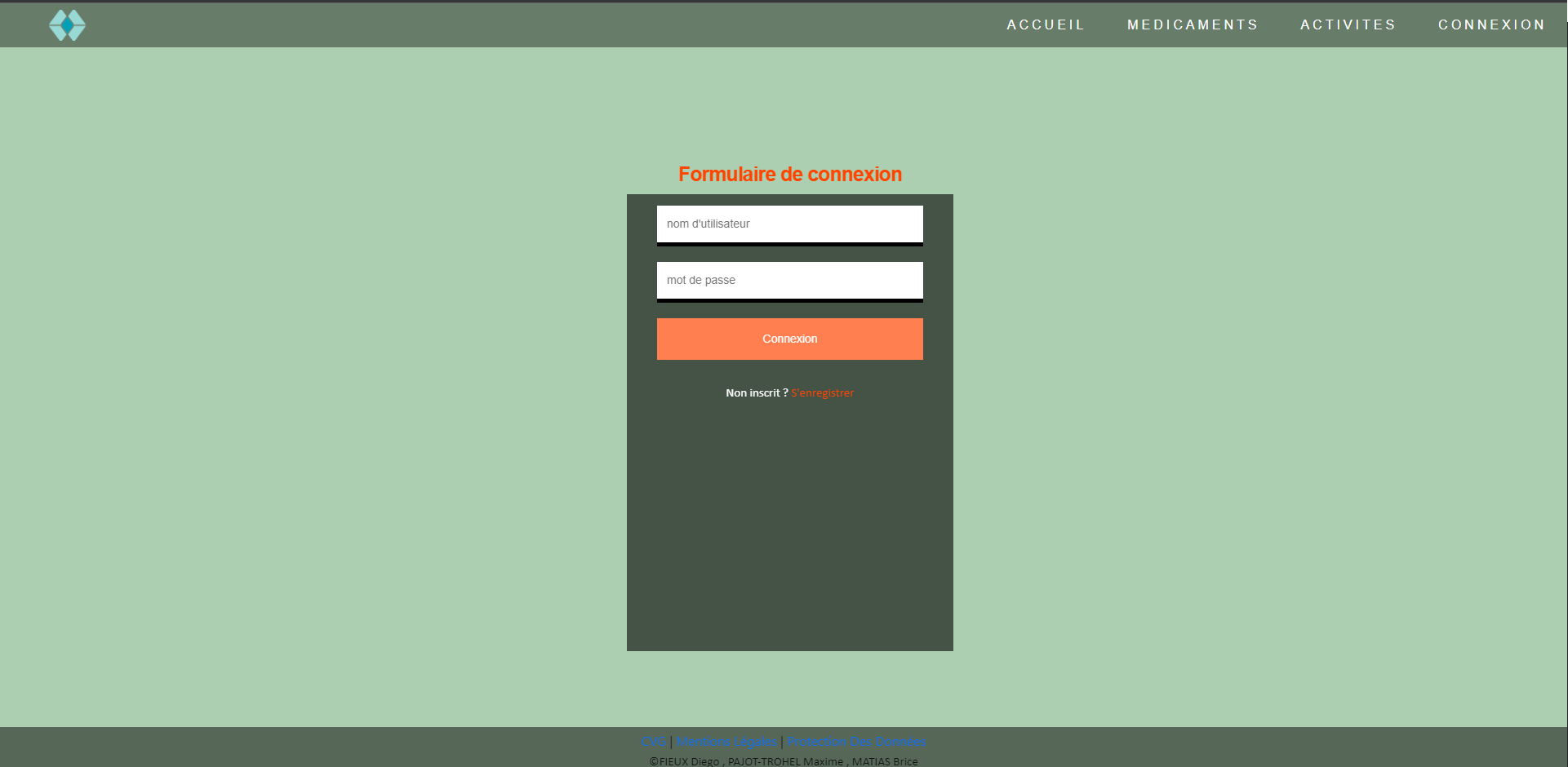


Les différentes pages

Accueil.php :

Inscription.php (sert de connexion aussi) :

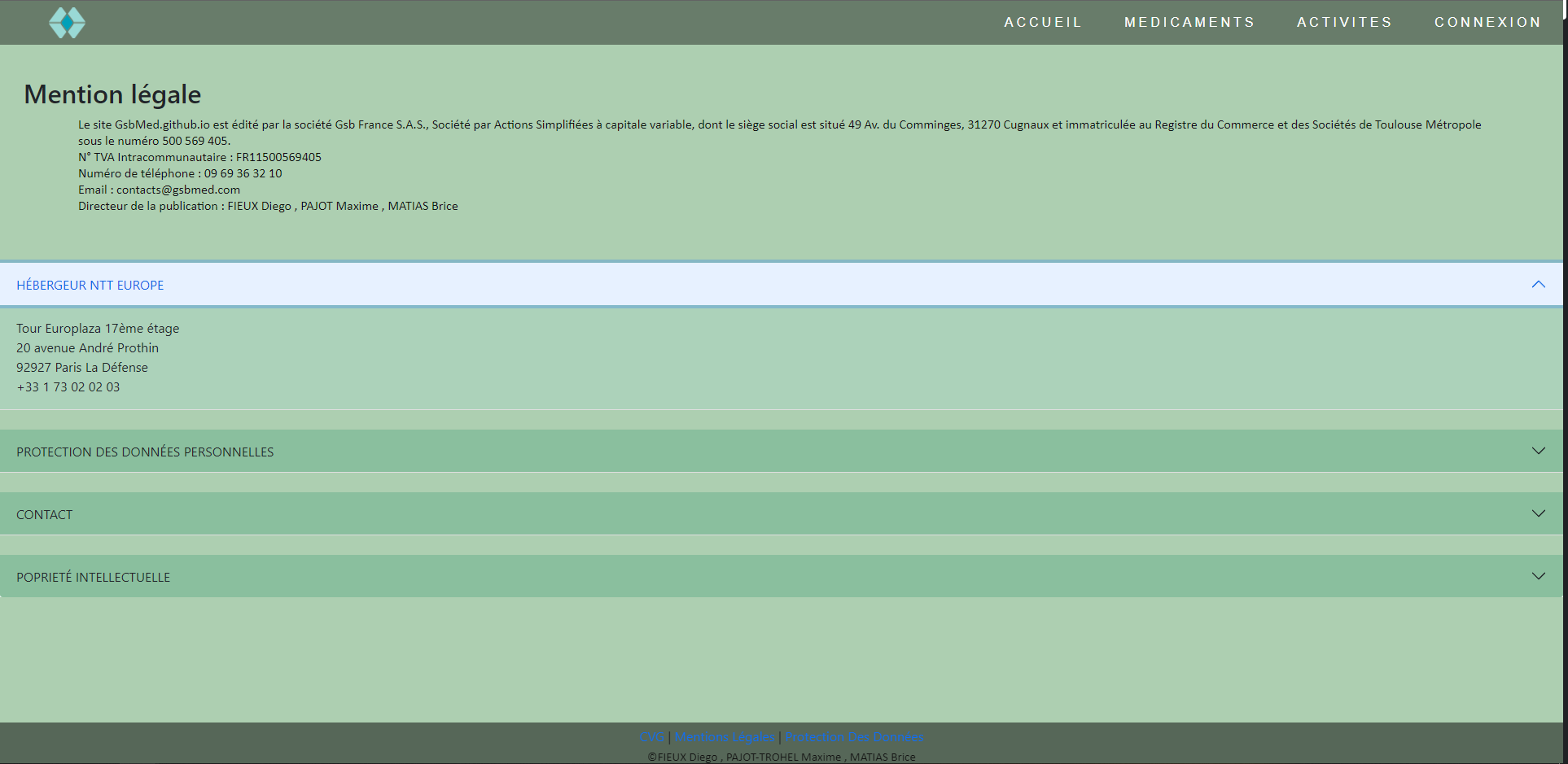




cvg.php :



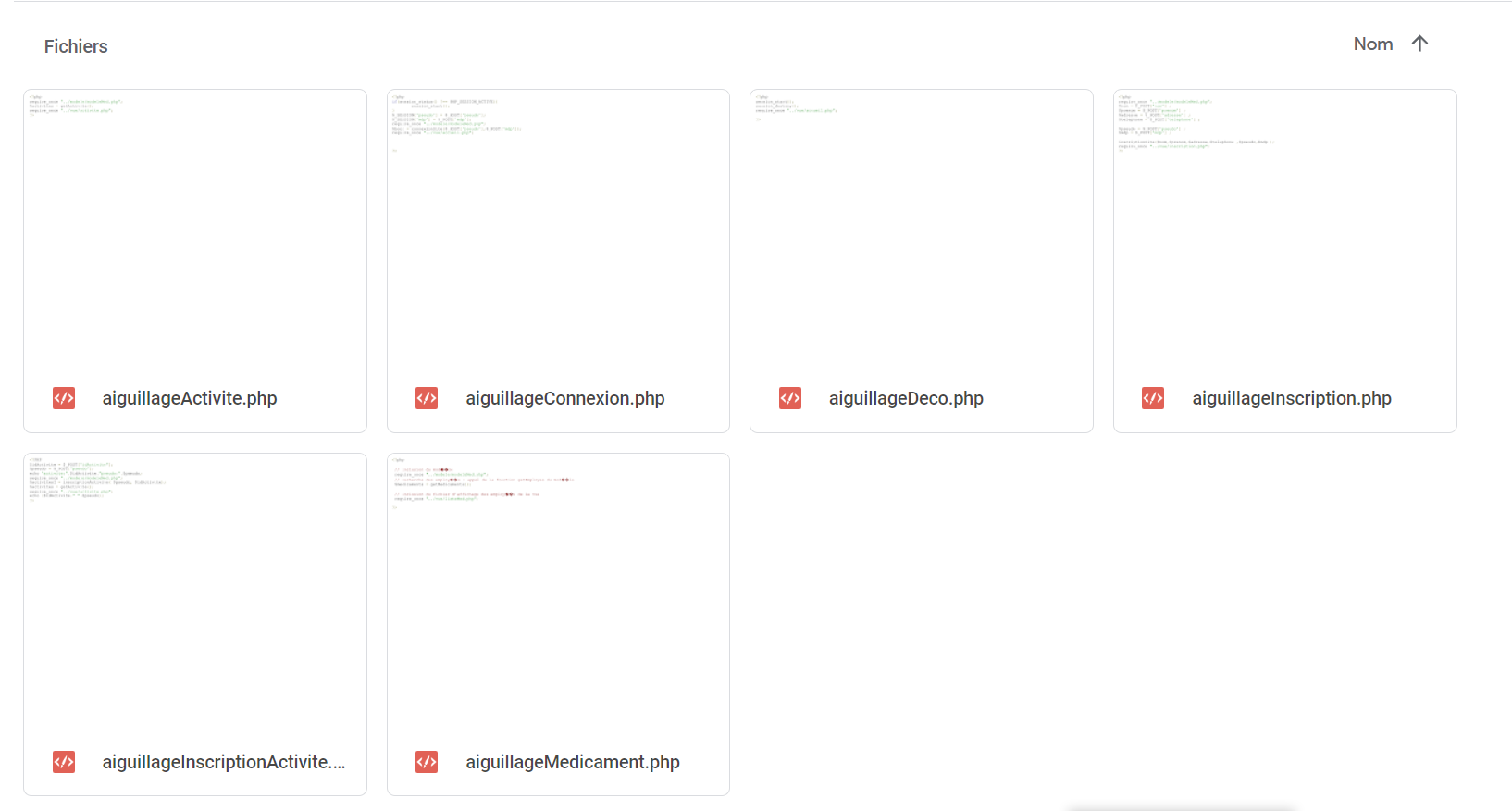
mentionLegale.php :



donneePersonnel.php :



Ensuite nous avons les **contrôleurs**, ces pages pages sont exclusivement en php.



Cette partie gère la logique du code, prend des décisions et renvoie le texte à afficher à la vue. C'est en quelque sorte l'intermédiaire entre le modèle et la vue.  
Contient exclusivement du php.  
  
Différents **controleurs** :

**AiguillageActivite** :

**<?php**

**require\_once "../modele/modeleMed.php";**

**$activites = getActivite();**

**require\_once "../vue/activite.php";**

**?>**

**AiguillageConnexion :**

**<?php**

**if(session\_status() !== PHP\_SESSION\_ACTIVE){**

**session\_start();**

**}**

**$\_SESSION['pseudo'] = $\_POST['pseudo'];**

**$\_SESSION['mdp'] = $\_POST['mdp'];**

**require\_once "../modele/modeleMed.php";**

**$bool = connexionSite($\_POST['pseudo'],$\_POST['mdp']);**

**require\_once "../vue/accueil.php";**

**?>**

**AiguillageDeco :**

**<?php**

**session\_start();**

**session\_destroy();**

**require\_once "../vue/accueil.php";**

**?>**

**AiguillageInscription :**

**<?php**

**require\_once "../modele/modeleMed.php";**

**$nom = $\_POST['nom'] ;**

**$prenom = $\_POST['prenom'] ;**

**$adresse = $\_POST['adresse'] ;**

**$telephone = $\_POST['telephone'] ;**

**$pseudo = $\_POST['pseudo'] ;**

**$mdp = $\_POST['mdp'] ;**

**inscriptionSite($nom,$prenom,$adresse,$telephone ,$pseudo,$mdp );**

**require\_once "../vue/inscription.php";**

**?>**

**AiguillageInscriptionActivite:**

**<?PHP**

**$idActivite = $\_POST["idActivite"];**

**$pseudo = $\_POST["pseudo"];**

**echo "activite:".$idActivite."pseudo:".$pseudo;**

**require\_once "../modele/modeleMed.php";**

**$activitesI = inscriptionActivite( $pseudo, $idActivite);**

**$activites = getActivite();**

**require\_once "../vue/activite.php";**

**echo ($idActivite." ".$pseudo);**

**?>**

**AiguillageMedicaments**

**<?php**

**// inclusion du modèle**

**require\_once "../modele/modeleMed.php";**

**// recherche des employés : appel de la fonction getEmployes du modèle**

**$medicaments = getMedicaments();**

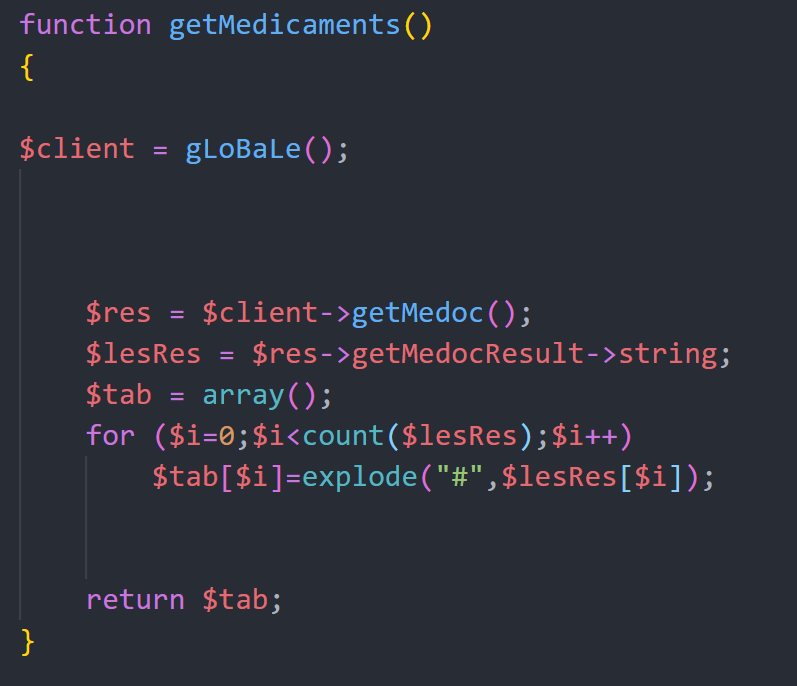
**// inclusion du fichier d'affichage des employés de la vue**

**require\_once "../vue/listeMed.php";**

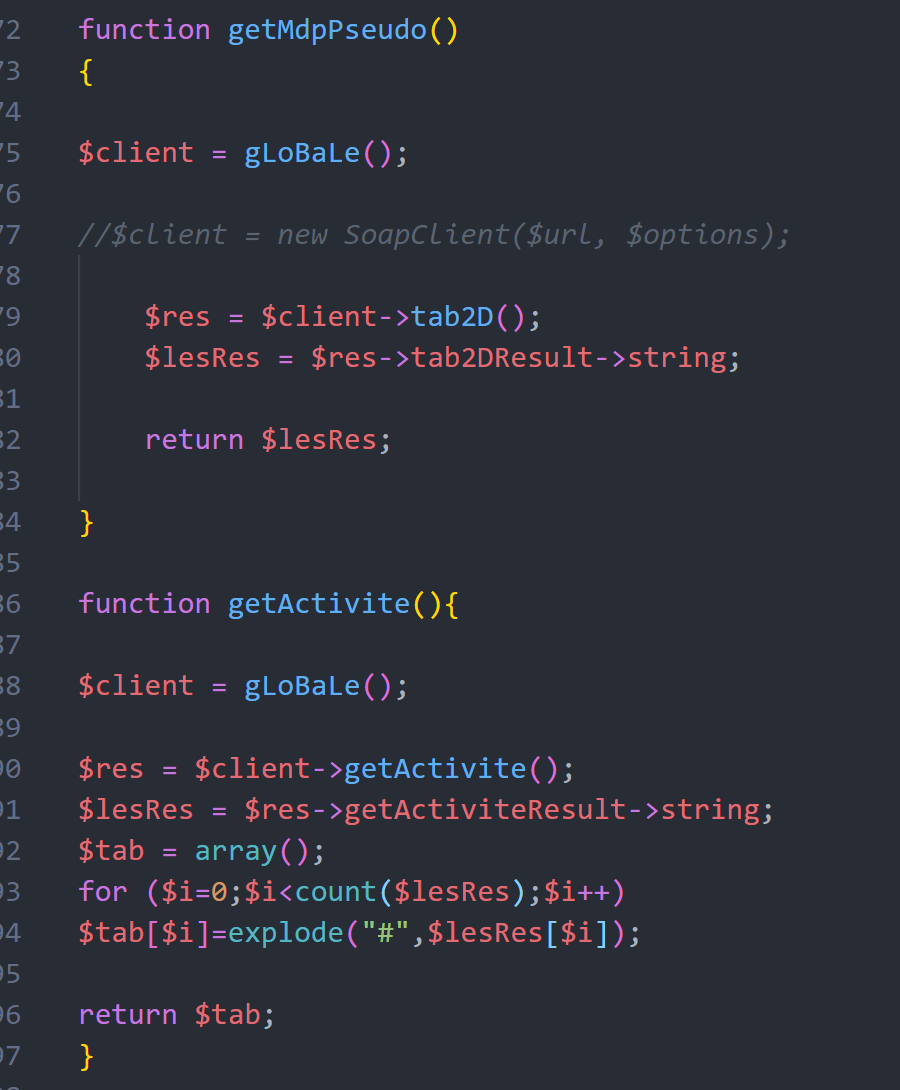
**?>**

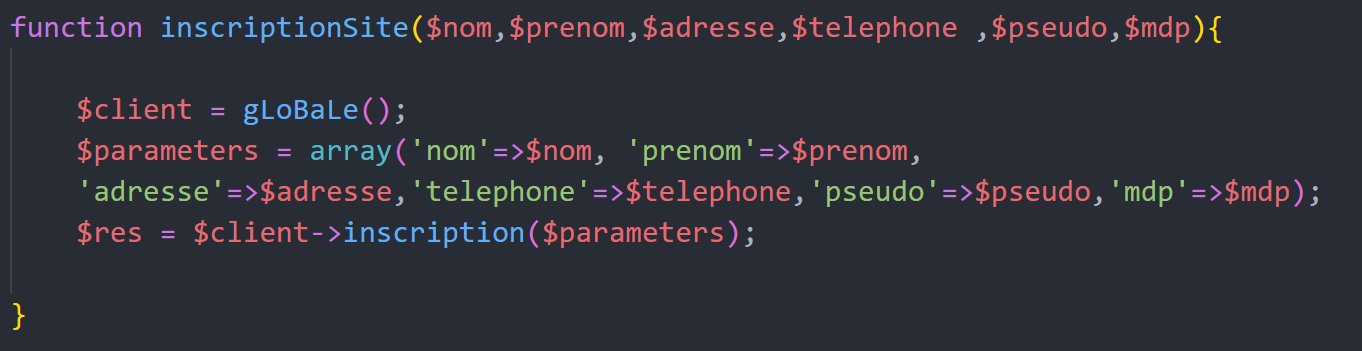
Et pour finir nous avons le **modéle**, qui est en relation avec le web service ou on y retrouve les fonctions du site

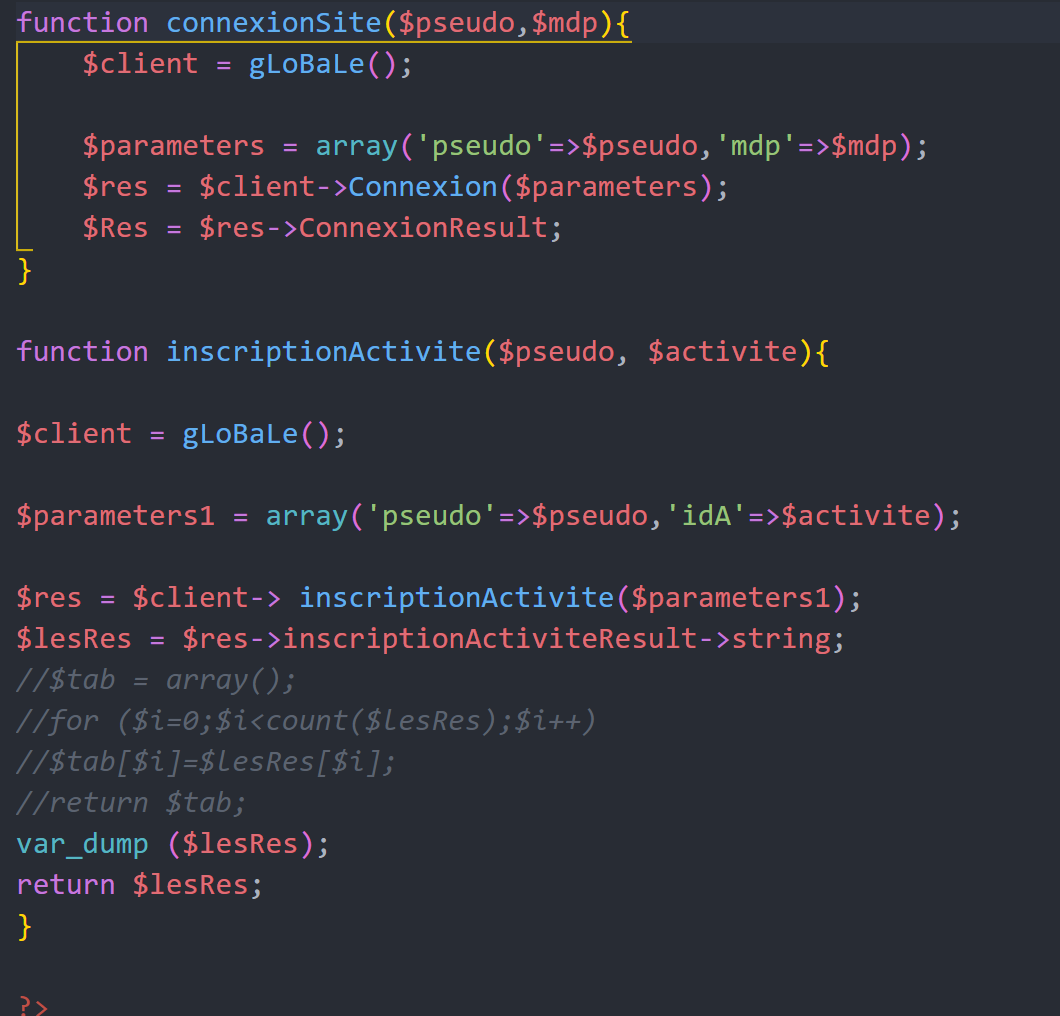






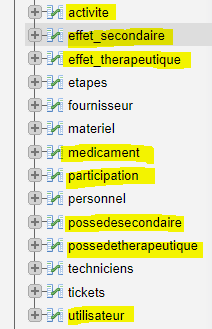




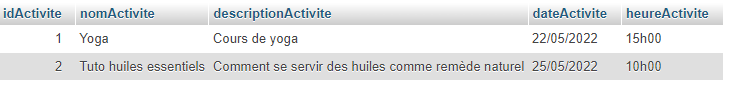


Base de données :

Liste des tables :

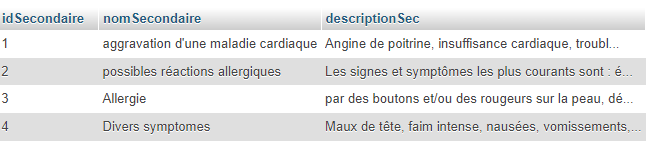


Activite :



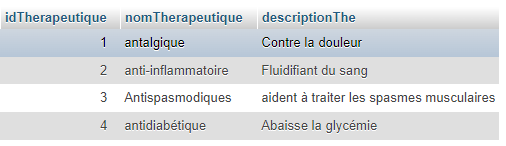
Effet\_secondaire :

Liste des effets secondaires qui sont ensuite liées à un médicament

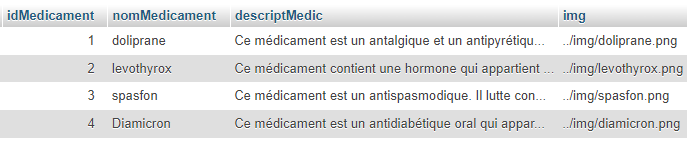


Effet\_therapeutique :

Liste des effets thérapeutiques qui sont ensuite liées à un médicament



Medicament :



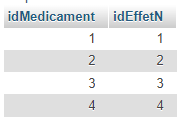
participation :

Liste des ids d’un utilisateur et de l’activité à laquelle il s’est inscrit



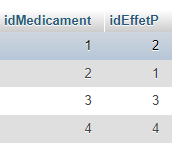
possedesecondaire :

Table qui lie un effet secondaire à un médicament



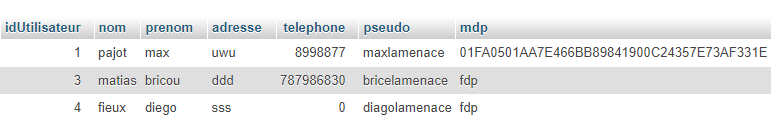
possedetherapeutique :

Table qui lie un effet thérapeutique à un médicament



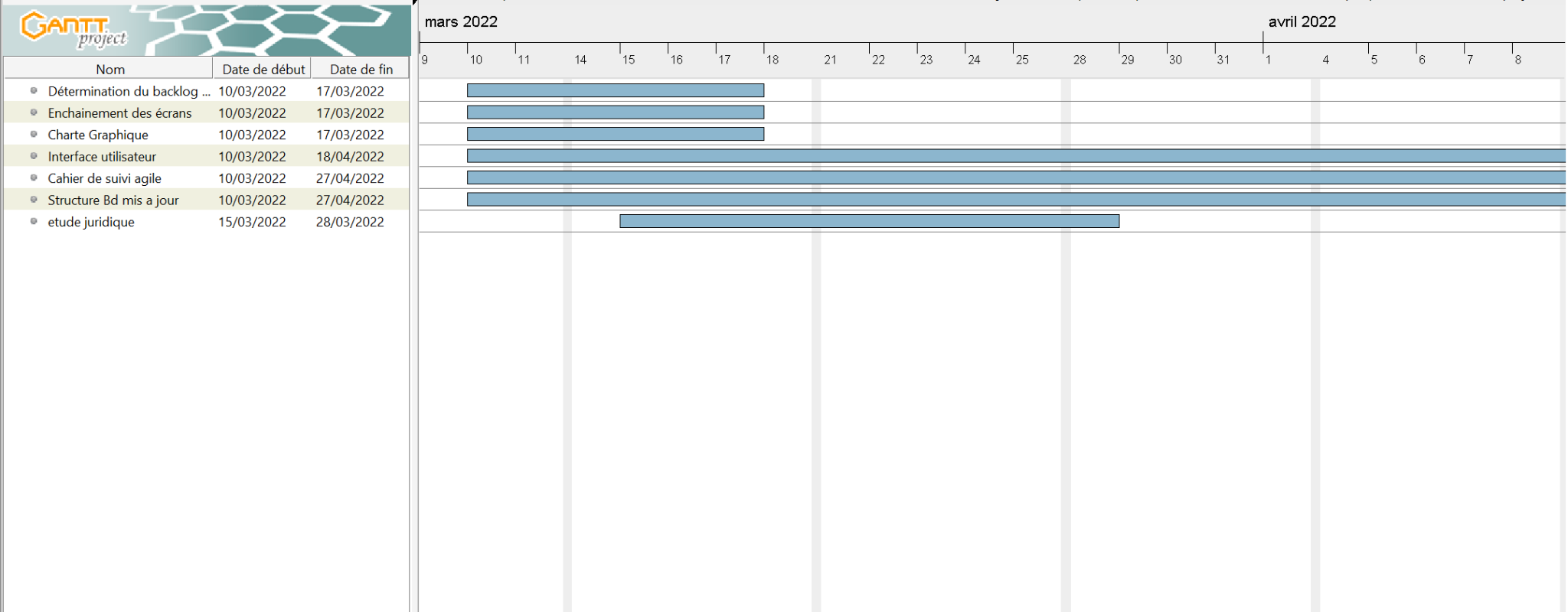
utilisateur :

Liste des utilisateurs



Gestion de projet :

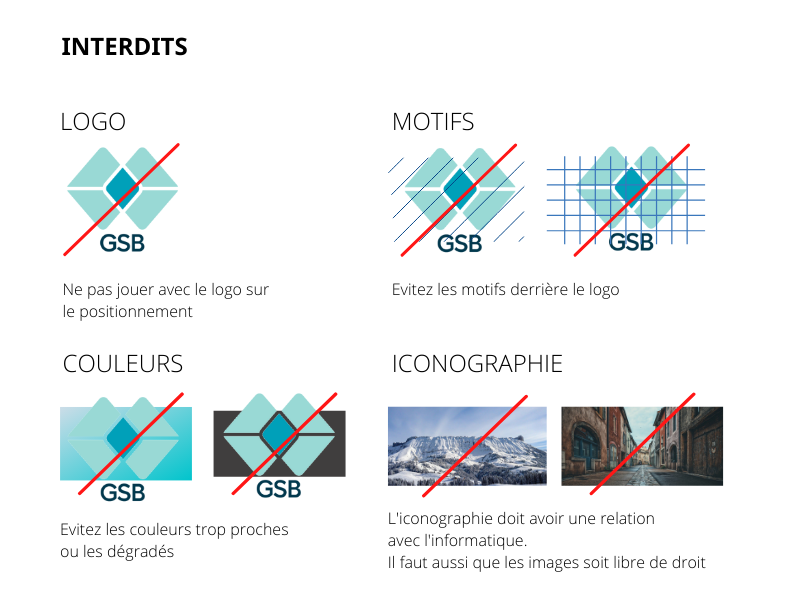
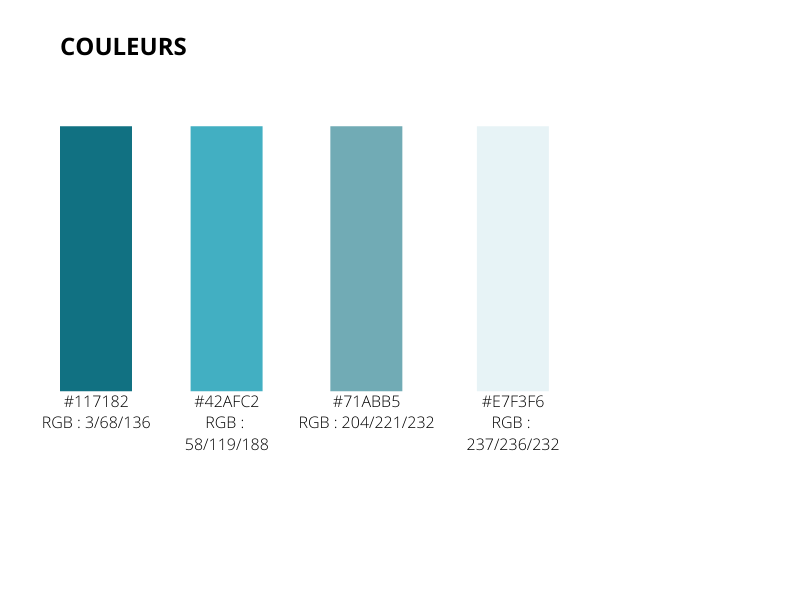
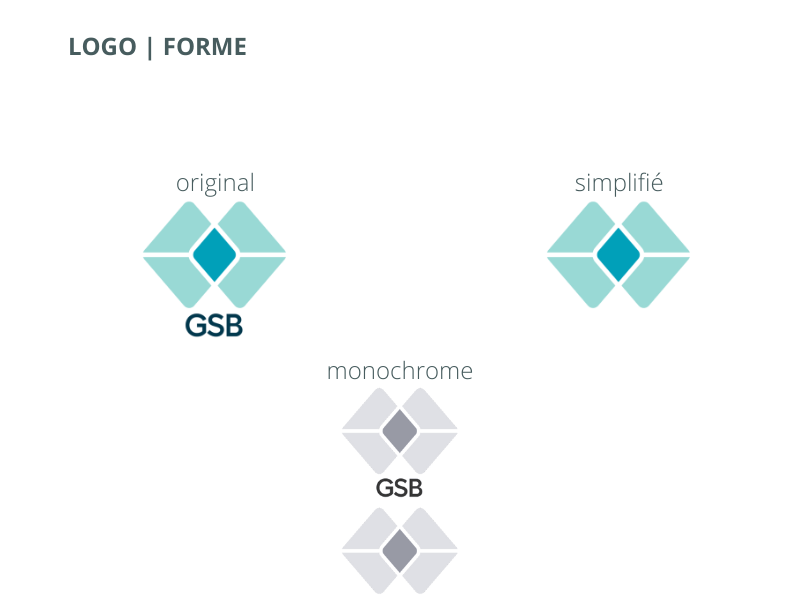
En amont nous avons fait de la gestion sur le projet afin d’estimer le temps qu’il nous faudra pour réaliser le projet mais aussi afin de déterminer quelles sont les tâches prioritaires et les tâches dont nous pouvons nous laisser de la marge.



Charte graphique :

La charte graphique contient toutes les consignes concernant le design du site, particulièrement dans les couleurs, les typographie et l’utilisation du logo afin de garder des sites cohérents.





Cahier d’activités :

Le projet débute le 10 mars 2022 :

On débute par la prise de connaissance du projet et on commence à travailler dessus. On crée les premières ébauches de la base de données, débute et termine la charte graphique nécessaire au bon déroulement du projet, on crée l'enchaînement d'images et le diagramme de Gantt.

Mardi 15 mars 2022 :

Étude juridique autour du projet, début du code des services web et mise en place des sprint et backlog.

Jeudi 17 mars 2022 :

Continuité des services web, début du code des différentes parties du modèle MVC et avancement de l’interface utilisateur.

Mardi 22 mars 2022 :

Continuité des services web, avancement du MVC et de l’interface utilisateur.

Jeudi 24 mars 2022 :

Continuité des services web, avancement du MVC et de l’interface utilisateur.

Mardi 29 mars 2022 :

Fin de l’interface du site et modification du MVC

Jeudi 31 mars 2022 :

Création de différentes pages (médicament), début de la gestion de la page de connexion et de l’inscription à une activité.

Mardi 12 avril 2022 :

Gestion des inscriptions aux activités.

Jeudi 14 avril 2022 :

Rédaction du rapport de projet et gestion des erreurs des web services.

Mardi 19 avril 2022 :

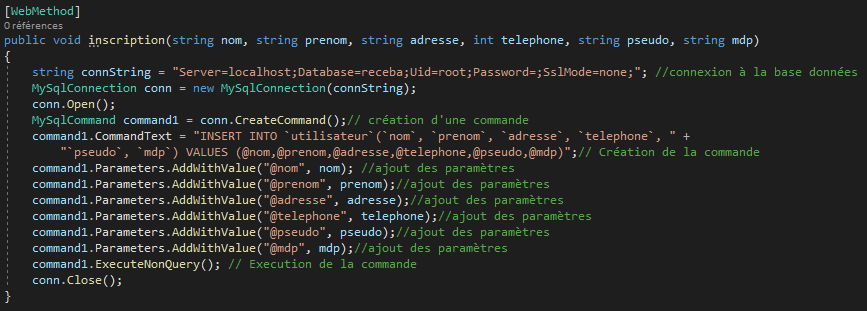
Finalisation du site et de l’inscription aux activités.

Web services

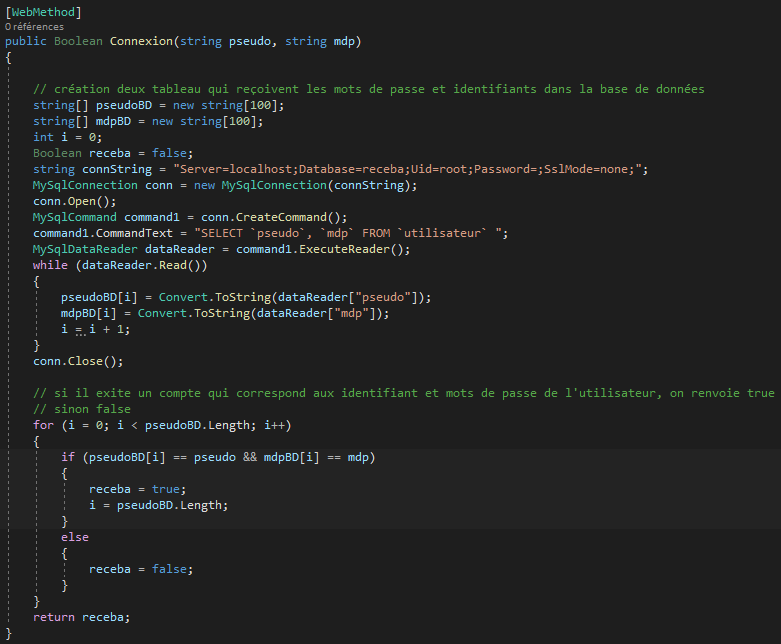
Les web services sont en C# au format asmx.

Ce web service va recevoir les informations nécessaires à l’inscription d’un utilisateur puis les ajouter dans la base de données.

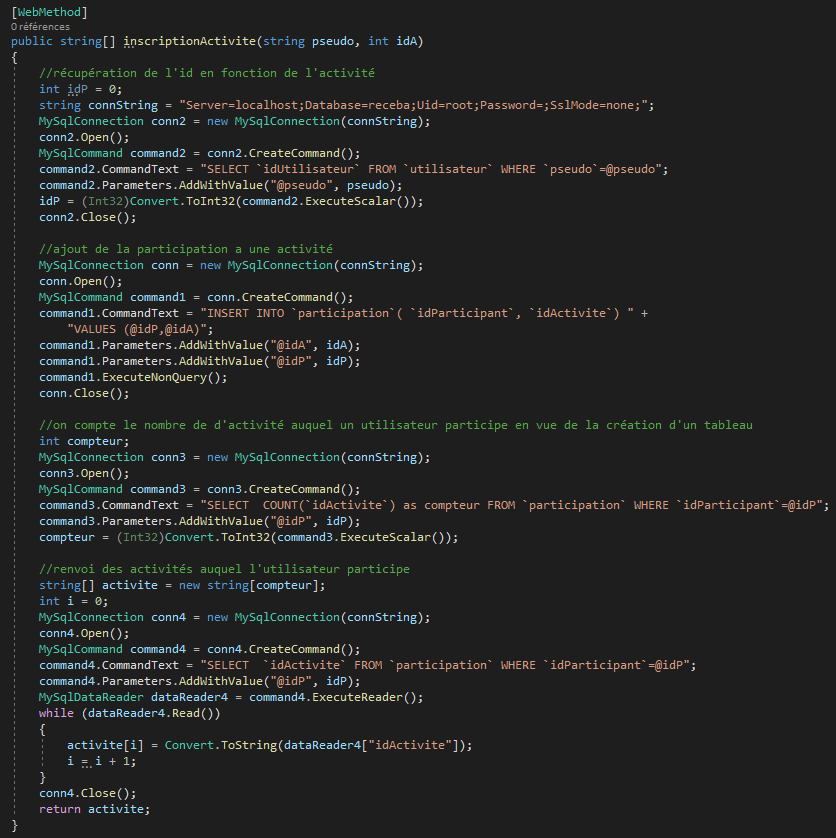
Pour ce faire, on crée la connexion avec la BD avec le connstring qui comporte les information nécessaire pour s’y connecter. Ensuite on ouvre une connexion avec conn.open. On crée une commande sql puis on ajoute les paramètres et on exécute la commande. Ensuite, on ferme la connexion.



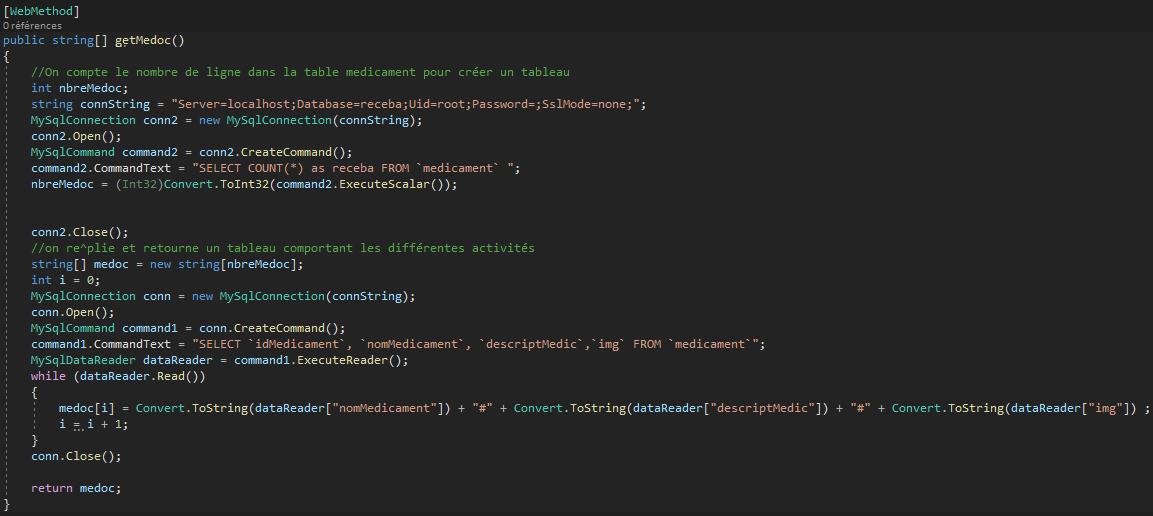
On récupère les pseudo et mot de passe dans la BD puis on les compare avec ce que l’utilisateur a rempli dans le formulaire de connexion pour valider ou non la connexion.



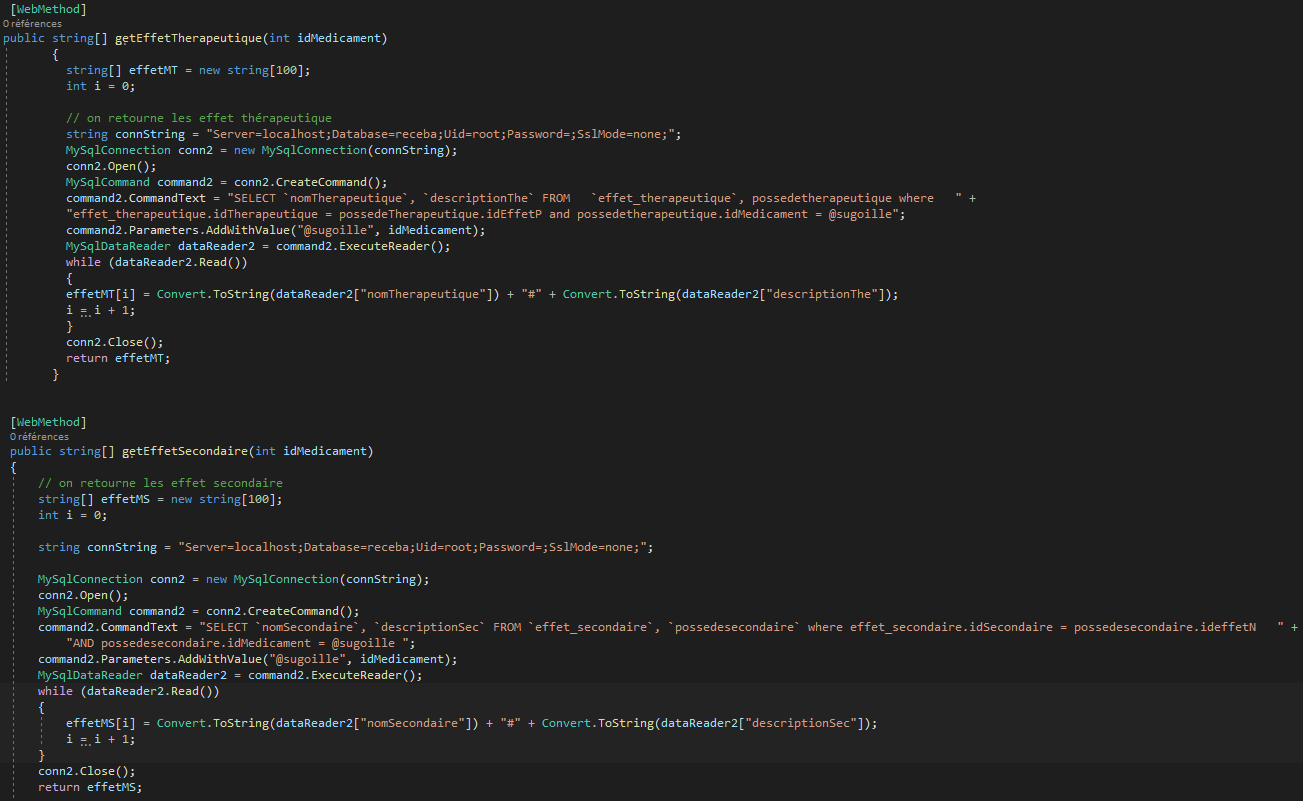
On récupère l’id correspondante au pseudo rentré. Puis on l’inscrit à l’activité sélectionnée avec l’id de l’activité et de l’utilisateur. Ensuite on récupère les activités auxquelles il est inscrit et on les retourne.



On récupère les médicaments et on les retournes.



On retourne un tableau avec les effet thérapeutique et secondaire.



On retourne toutes les activités.

