

Documentation technique

Architecture de l'application

L'application a été développée sous l'architecture MVC (Modèle-Vue-Contrôleur) afin d'optimiser l'organisation du code et son développement. Cette architecture permet notamment de bien séparer les données, la présentation et les traitements.

```
▼ GSB_CR_2
  ▼ controleurs
    c_authentification.php
    c_deconnexion.php
    c_estConnecte.php
    c_medicaments.php
    c_pages.php
    c_praticiens.php
    c_rapports.php
    c_saisie_rapport.php
  ▼ modeles
    m_authentification.php
    m_connexion.php
    m_medicaments.php
    m_praticiens.php
    m_rapports.php
    m_saisie_rapport.php
  ► style
  ▼ vues
    v_accueil.php
    v_authentification.php
    v_erreur.php
    v_footer.php
    v_medicaments.php
    v_menu.php
    v_navig.php
    v_praticiens.php
    v_rapports.php
    v_saisie_rapport.php
  index.php
```

Modèles :

Ce dossier contient tous les fichiers modèles, c'est-à-dire les fichiers ayant accès à la base de données. Ils contiennent toutes les fonctions nécessaires au développement de l'application. Les fichiers sont séparés selon la page sur laquelle ils interviennent.

Vues :

Ce dossier comporte les fichiers vues, c'est-à-dire les fichiers qui affichent du contenu. La vue se contente d'afficher les résultats renvoyés par le modèle et d'interagir avec l'utilisateur.

Contrôleurs :

Ce dossier rassemble les fichiers contrôleurs permettant de traiter les données récupérées grâce aux modèles. Ils renvoient ensuite le texte à afficher aux vues.

Présentation des méthodes utilisées

- **Modèle connexion :**

```
function connect()
```

Cette fonction permet la connexion à la base de données.

- **Modèle authentification**

```
function verifAuthentification($login, $mdp)
```

Cette fonction permet de savoir si un utilisateur est présent dans la base de données. Cette méthode prend en paramètre le login et le mot de passe saisi par l'utilisateur.

```
function getInfosUser($login, $mdp)
```

Cette fonction permet de récupérer le nom et le prénom de l'utilisateur connecté. Cette méthode prend en paramètre le login et le mot de passe saisi par l'utilisateur.

- **Modèle médicaments :**

```
function getMedicaments()
```

Cette fonction permet de récupérer la liste des médicaments.

```
function getInfosMedicament($codeMedic)
```

Cette fonction permet de récupérer les informations relatives à un médicament. Cette méthode prend en paramètre un code médicament.

- **Modèle praticiens :**

```
function getPraticiens()
```

Cette fonction permet de récupérer la liste des praticiens.

```
function getInfosPraticien($idPrat)
```

Cette fonction permet de récupérer les informations relatives à un praticien. Cette méthode prend en paramètre un id praticien.

- **Modèle rapports :**

```
function getRapports($idVisiteur)
```

Cette fonction permet de récupérer la liste des rapports rédigés par le visiteur connecté. Cette méthode prend en paramètre l'id du visiteur.

```
function getInfosRapports($numRapport)
```

Cette fonction permet de récupérer certaines informations relatives à un rapport, à savoir : le numéro, le nom et prénom du praticien, la date de visite, le coefficient de confiance, le motif, le bilan, la documentation et l'id du remplaçant. Cette méthode prend en paramètre le numéro du rapport.

```
function getInfosRapportsRempl($numRapport)
```

Cette fonction permet de récupérer les informations du remplaçant présent lors de la visite, s'il existe. Cette méthode prend en paramètre le numéro du rapport.

```
function getInfosRapportProduits($numRapport)
```

Cette fonction permet de récupérer la liste des produits présentés lors de la visite sélectionnée. Cette méthode prend en paramètre le numéro du rapport.

```
function getInfosRapportEchantillons($numRapport)
```

Cette fonction permet de récupérer la liste des échantillons offerts lors de la visite sélectionnée ainsi que la quantité. Cette méthode prend en paramètre le numéro du rapport.

• **Modèle saisie rapport :**

```
function getNumRapportMax()
```

Cette fonction permet de récupérer le numéro du dernier rapport enregistré + 1.

```
function addRapport($numR, $dateV, $coef, $motif, $bilan, $doc, $dateR, $idRemp, $idPrat, $pdt1, $pdt2, $idVisiteur)
```

Cette procédure permet d'ajouter un nouveau rapport à la base de données. Cette méthode prend en paramètre toutes les informations relatives à un rapport, excepté les échantillons.

```
function addEchantillon($codeMedic, $numR, $qte)
```

Cette procédure permet d'ajouter un nouvel échantillon à la base de données. Cette méthode prend en paramètre le numéro du rapport, le code du médicament et la quantité.

```
function updateCoefPrat($coef, $idPrat)
```

Cette procédure permet de mettre à jour le coefficient de confiance du praticien. Cette méthode prend en paramètre le nouveau coefficient et l'id du praticien.