

Documentation technique – Gestion de Visite

Ce projet a une architecture MVC (Modèle – Vue – Contrôleur).

Modèle :

Il existe 2 types de classe modèle différentes, les classes métier ainsi que les classes DAO.

Les classes DAO contiennent toutes les requêtes SQL qui sont liées à la base de donnée.

La classe « connexionDB » est très importante car c'est elle qui permet au programme de se connecter à la base de donnée.

```
public function connexionPDO(){
    try {
        $this->connect = new PDO( dsn: "mysql:host=".$this->serveur.";dbname=".$this->bdd, $this->login, $this->mdp);
        $this->connect->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
    } catch (PDOException $e) {
        print "Erreur de connexion PDO ";
        die();
    }
    if(!$this->connect){
        die('Erreur : '.mysqli_connect_error());
    }
    //echo 'Connexion réussie'. "\n";
}

//////////Prepare une requete //////////
public function PrepareRequete($requete){
    return $this->connect-> prepare($requete);
}
```

Il faut ensuite utiliser ses fonctions au début de chaque fonction de requête SQL.

Cette fonction contient une requête permettant de récupérer les informations de tous les médecins présents dans la base de donnée.

```

////////////////////////////////// Tous les medecins ////////////////////////////////////

public function getAllMedecins() {
    $resultat = array();
    try {
        $connexion = new connexionDB();
        $connexion->connexionPDO();
        $query = $connexion->PrepareRequete( requete: "Select * from medecin");
        $query->execute();
        $ligne = $query->fetch(PDO::FETCH_ASSOC);
        while ($ligne) {
            $resultat[] = $ligne;
            $ligne = $query->fetch(PDO::FETCH_ASSOC);
        }
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage();
        die();
    }
    return $resultat;
}

```

Contrôleur :

Les contrôleurs sert à synchroniser les modèles et la vue.

Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer.

Il peut donc appeler des fonctions des différents modèle et avertir la vue que des données ont changé pour que celle-ci se mette à jour.

Le contrôleur principale a pour but de diriger l'utilisateur quand il arrive sur le site. Pour qu'il aille sur le login ou le menu.

```

function mainController($action){
    if($action == "Login"){
        return "login_view.php";
    }
    elseif($action == "menu"){
        return "menu_view.php";
    }
}

```

Dans ce contrôleur, ce code va permettre de vérifier si l'action 'functionSee' est utilisé dans la vue. Si c'est le cas, le contrôleur va stocker les données de ces deux requêtes des modèles dans des variables et va ensuite appeler les vues nécessaires.

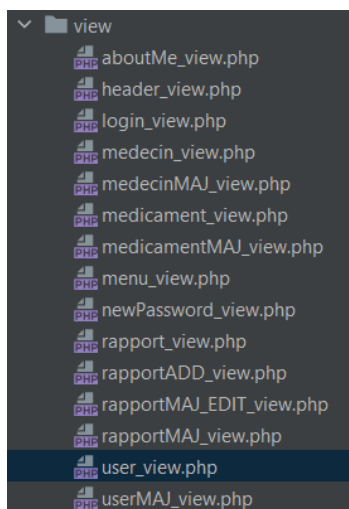
```
////////////////////Fonction Voir //////////////////////  
  
if(isset($_REQUEST['functionSee'])){  
    $view = true;  
  
    $medecinData = $medecinDATA->getInfosMedecins($_GET['functionSee']);  
    $rapportMedecin = $medecinDATA->getRapportMedecin($_GET['functionSee']);  
  
    require_once ("../view/header_view.php");  
    require_once ("../view/medecinMAJ_view.php");  
}
```

Vue :

La vue est très majoritairement codé en HTML, avec du php pour les actions.

Nous pouvons voir dans « medecin_view » 2 boutons qui ont chacun un nom permettant d'être appelé par les contrôleurs. Ils ont chacun la valeur « \$medecinData[\$i]['id'] » qui permet de récupérer les informations du médecin que l'on souhaite afficher ou modifier.

Il existe un fichier de vue pour chaque page, ou partie de page que nous souhaitons afficher.



Chaque nom de fichier termine par la fonction dont il est destiné :

- _controller,
- _modele ou DAO,
- _view.