

# **GSB-Visite - Documentation technique**

## **Chaibi Yanis**



## **Documentation Technique**

GSB-Visite - AP 2021



# GSB-Visite - Documentation technique

## Chaibi Yanis

### Informations

<b>Nom du Projet</b>	GSB-Visite
<b>Type de document</b>	Documentation technique
<b>Date</b>	17/11/2021
<b>Mots-clés</b>	
<b>Auteur</b>	Yanis Chaibi

### Table des matières

- **1- L'application à réaliser**
  - 1.1-Objectif
  - 1.2-Resultat
  - 1.3-Objectif à réaliser ou améliorer
- **2-Technologie utilisée**
- **3-Modèle de données**
- **4- Arborescence des fichiers**
- **5- Architecture**
- **6- Explication du code**

## 1-L'application à réaliser

### 1.1-Objectif:

L'application à réaliser pour l'entreprise Galaxy Swiss Bourdin (GSB) a pour but de permettre aux visiteurs de gérer ses visites. L'application doit permettre aux visiteurs de **se connecter** puis par la suite de **gérer ses rapports de visite** et de pouvoir **gérer ses médecins**.

### 1.2-Resultat:

L'application actuellement permet aux visiteurs de se connecter et de gérer ses rapports de visite.

### 1.3-Objectif à réaliser ou améliore

L'application réaliser ne permet pas entièrement aux visiteur de gérer les médecins.

## 2-Technologie et logiciel Utilisé

### **HTML**



HTML 5 a été utilisé afin de structurer, mettre en forme les pages web

### **CSS**



CSS 3 a été utilisé pour le design des pages web



PHP a été utilisé afin de traiter les données

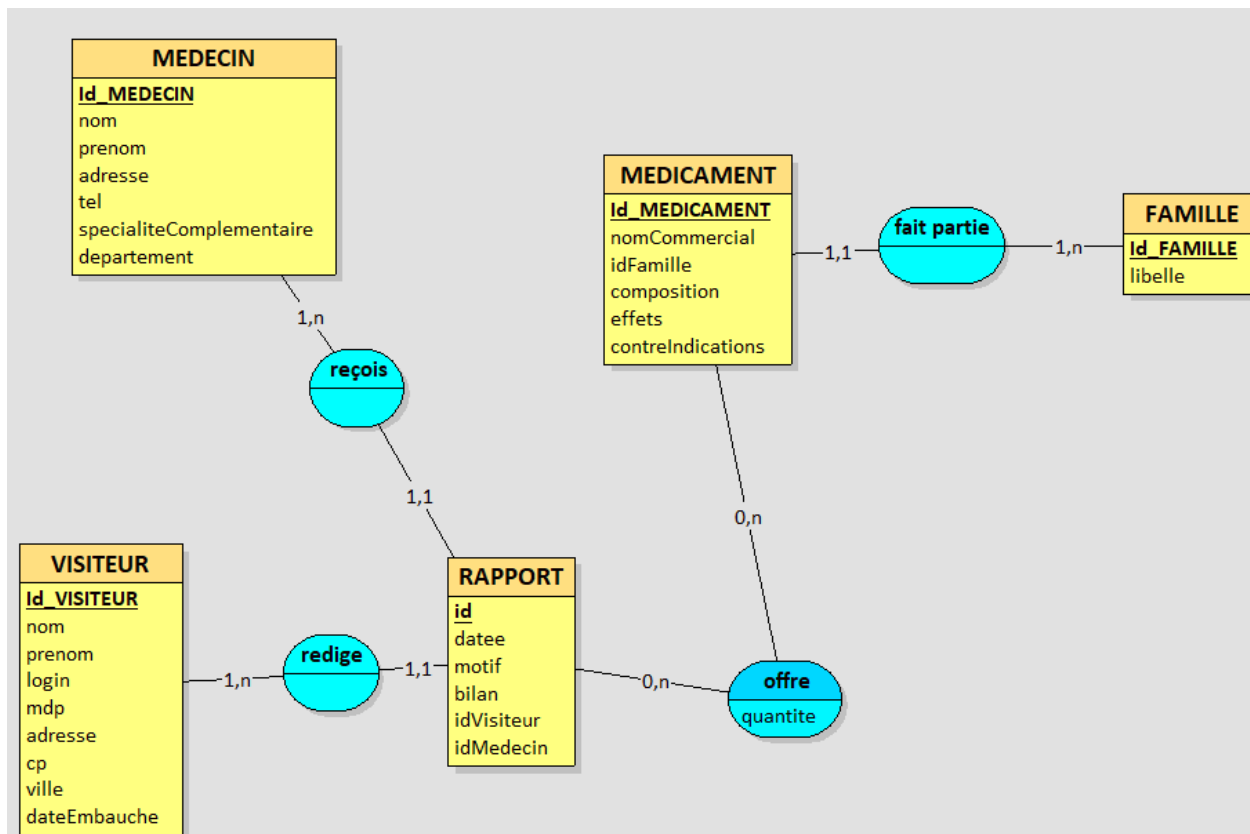


Bootstrap a permis de rendre plus facilement le site responsive

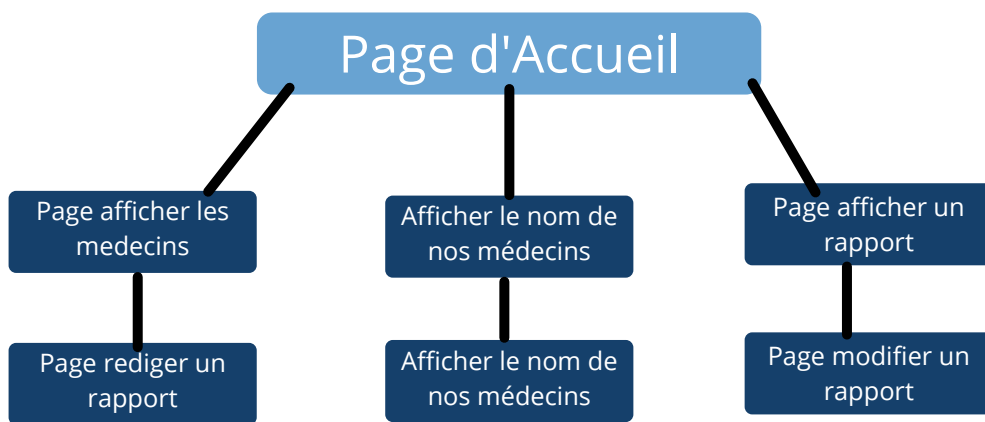
# GSB-Visite - Documentation technique

## Chaibi Yanis

### 3-Modèle de données

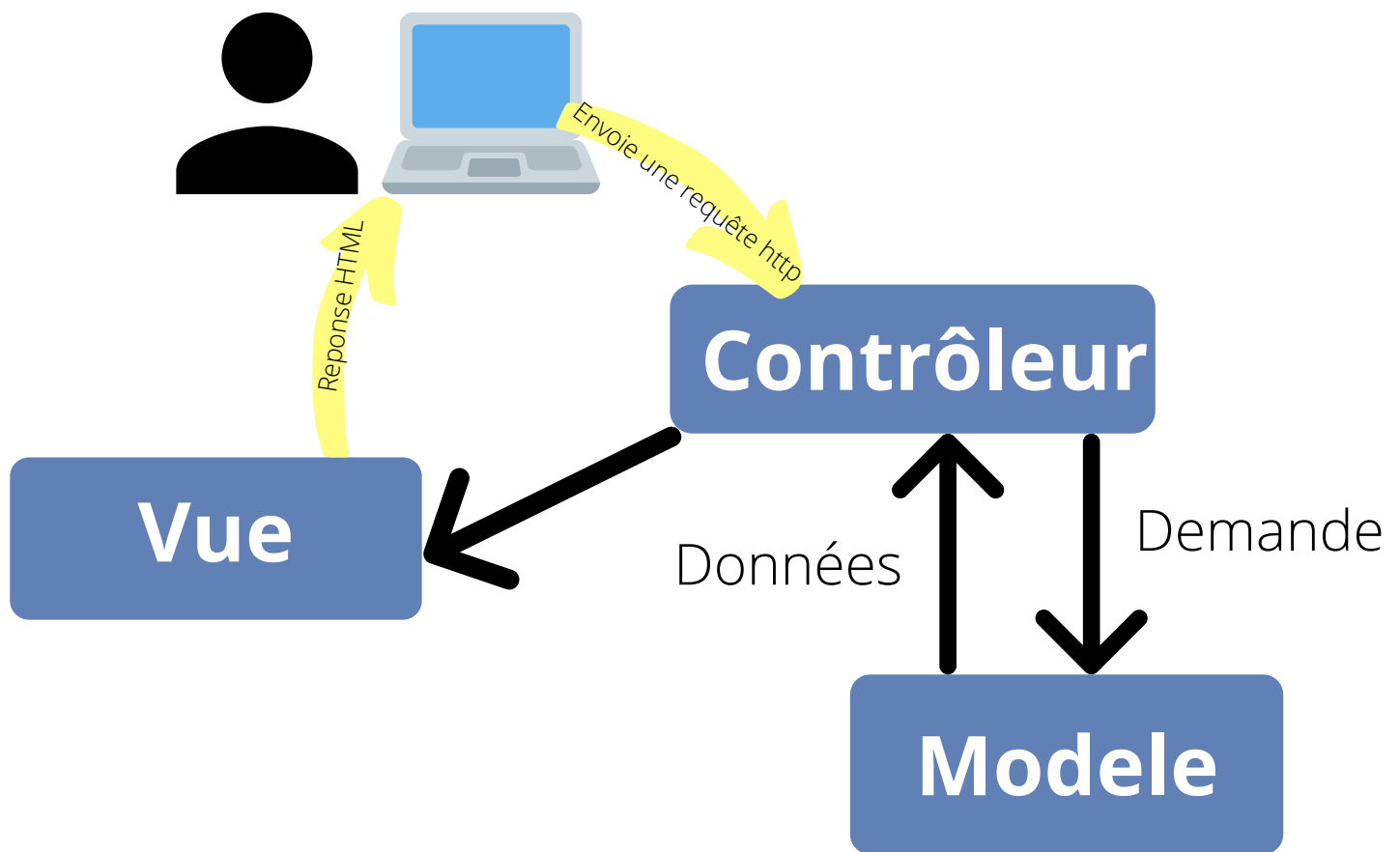


## 4-Arborescences des fichiers



## 5-Architecture

L'architecture de l'application a été faite en mvc



### 6- Description du code

```
public function login(){
    //Sanitize POST data
    $_POST = filter_input_array(INPUT_POST, FILTER_SANITIZE_STRING);

    //Init data
    $data=[
        'login' => trim($_POST['login']),
        'usersPwd' => trim($_POST['usersPwd'])
    ];

    if(empty($data['login']) || empty($data['usersPwd'])){
        flash("login", "Please fill out all inputs");
        header("location: ../login.php");
        exit();
    }

    //Check for login
    if($this->userModel->findVisitorByLogin($data['login'], $data['login'])){
        //User Found
        $loggedInUser = $this->userModel->login($data['login'], $data['usersPwd']);
        if($loggedInUser){
            //Create session
            $this->createUserSession($loggedInUser);
        }else{
            flash("login", "Password Incorrect");
            redirect("../login.php");
        }
    }else{
        flash("login", "No user found");
        redirect("../login.php");
    }
}
```



# GSB-Visite - Documentation technique

## Chaibi Yanis

### 6- Description du code affichant tout les médecins

```
<?php

require_once '../models/Doctor.php';
require_once '../helpers/session_helper.php';

class Doctors {

    private $doctorModel;

    public function __construct(){
        $this->doctorModel = new Doctor;
    }

    public function get(){
        $doctors = $this->doctorModel->findAllDoctors();
        $_SESSION['doctorsLists'] = $doctors[0]->prenom;

        $returnedDocs = array();
        foreach($doctors as $doc){
            array_push($returnedDocs, $doc);
        }
        $_SESSION['doctorsList'] = $returnedDocs;
        redirect("../medecin.php");
    }

}
```

la fonction get nous permet d'obtenir la liste de tout les médecins tout d'abord on crée une variable qui est égale a la requête nous permettant d'obtenir tout les médecins (findAllDoctors) ensuite on crée une liste puis on parcourt cette liste et a chaque medecin trouver on ajoute une ligne

```
//get all Doctors
public function getAllDoctors(){
    $this->db->query('SELECT * FROM medecin');

    $row = $this->db->resultSet();

    //Check row
    if($this->db->rowCount() > 0){
        return $row;
    }else{
        return false;
    }
}

?>
```

ici nous nous trouvons dans le model .

La fonction getAllDoctors contient ici la requête permettant d'obtenir tout les medecins avec toute leurs informations

# GSB-Visite - Documentation technique

## Chaibi Yanis

### 7- Description du code affichant tout les médecins

#### View de l'affichage des médecins

```
<?php
foreach($_SESSION['doctorsList'] as $doctor){

?>
<tr>
<td scope="row"><?php echo utf8_encode($doctor->id); ?></td>
<td><?php echo utf8_encode($doctor->prenom); ?></td>
<td>
<form method="post" action="rapport.php">
<input type="hidden" name="type" value="get">
<button type="submit" name="submit" type="button" class="btn btn-primary" href="rapport.php">
<?php echo utf8_encode($doctor->nom); ?> </button> </form></td>
</tr>

<?php
}
?>
```

# GSB-Visite - Documentation technique

## Chaibi Yanis

### 8- Description du code enregistrant les rapports

#### Model de l'enregistrement des rapports

```
Class Rapport{  
    private $db;  
  
    public function __construct() {  
        $this->db = new Database;  
    }  
  
    //register Data  
    public function register($data){  
  
        $this->db->query('INSERT INTO rapport ( date, motif, bilan, idVisiteur, idMedecin)  
VALUES (:date, :motif, :bilan, :idVisiteur, :idMedecin)');  
        //Bind values  
        $this->db->bind(':date', $data['date']);  
        $this->db->bind(':motif', $data['motif']);  
        $this->db->bind(':bilan', $data['bilan']);  
        $this->db->bind(':idVisiteur', $data['idVisiteur']);  
        $this->db->bind(':idMedecin', $data['idMedecin']);  
  
        //Execute  
        if($this->db->execute()){  
            return true;  
        }else{  
            return false;  
        }  
    }  
}
```

Ici on se trouve dans le model  
l'enregistrement de rapport  
se fera grâce à la fonction  
register qui prend en  
paramètre data qu'on associe  
au valeur correspondant a  
ceux de la requête

#### Contrôleur pour l'enregistrement des rapports

```
require_once '../models/User.php';  
require_once '../models/Rapport.php';  
require_once '../helpers/session_helper.php';  
  
Class Rapports {  
    private $rapportModel;  
  
    public function __construct(){  
        $this->rapportModel = new Rapport;  
    }  
  
    public function register(){  
        //Process form  
        echo "<script type='text/javascript'>alert('BLABLA');</script>";  
        //Sanitize POST data  
        $_POST = filter_input_array(INPUT_POST, FILTER_SANITIZE_STRING);  
  
        //Init data  
        $data = [  
            'date' => $_POST['date'],  
            'motif' => $_POST['motif'],  
            'bilan' => $_POST['bilan'],  
            'idVisiteur' => $_POST['idVisiteur'],  
            'idMedecin' => $_POST['idMedecin'],  
        ];  
  
        //Validate inputs  
        if(empty($data['date']) || empty($data['motif']) || empty($data['bilan']) || empty($data['idVisiteur']) || empty($data['idMedecin'])){  
            flash("register", "Please fill out all inputs");  
            redirect("../rapport.php");  
        }  
  
        //Enregistre le rapport  
        if($this->rapportModel->register($data)){  
            redirect("../rapport.php");  
        }else{  
            die("Something went wrong");  
        }  
    }  
}
```

la fonction register  
initialise les données ensuite  
on vérifie que tout les champs  
ont était remplis puis on fait  
appel a la fonction register du  
model

# GSB-Visite - Documentation technique

## Chaibi Yanis

### 8- Description du code affichant les rapports

#### Modele de l'affichage des rapports

```
class Edit {
    private $db;

    public function __construct(){
        $this->db = new Database;
    }

    //Find user by email or username
    public function getAllReports($id){
        $this->db->query('SELECT * FROM rapport WHERE idVisiteur = :id');

        $this->db->bind(':id', $id);

        $row = $this->db->resultSet();

        //Check row
        if($this->db->rowCount() > 0){
            return $row;
        }else{
            return false;
        }
    }
}
```

la fonction getAllReports prend en paramètre (id) qui est égal à l'id puis on fait notre requête puis on affecte l'id du visiteur à \$id

#### Contrôleur de l'affichage des rapports

```
class Edits {
    private $editModel;

    public function __construct(){
        $this->editModel = new Edit;
    }

    public function getAllReport(){

        $edits = $this->editModel->getAllReports($_SESSION['usersId']);

        $returnedEdits = array();
        foreach($edits as $ed){
            array_push($returnedEdits, $ed);
        }
        $_SESSION['reportslist'] = $returnedEdits;
        redirect("../allrapport.php");
    }
}
```

La fonction getAllreports crée une variable \$edot qui est = à l'id du visiteur afin de lui afficher uniquement ses rapports puis ensuite on crée une liste \$returnedEdits

# GSB-Visite - Documentation technique

## Chaibi Yanis

### 9- Description du code recherchant les rapports

#### Contrôleur de la recherche des rapports

```
public function searchReports()  
{  
    $edites = $this->editModel->getAllReports($_SESSION['usersId']);  
  
    $criteria = $_POST['find'];  
  
    $result = $this->editModel->findReports($criteria);  
  
    $returnSearch = array();  
    foreach($result as $rs){  
        array_push($returnSearch,$rs);  
    }  
  
    $_SESSION['reportsList'] = $returnSearch;  
    redirect("../allrapport.php");  
}
```

La fonction searchReports va tout d'abord crée une variable \$criteria qui est = au nom de l'input (find) permettant de faire des recherches ensuite on crée une variable \$result qui est = au résultat de la requête puis ensuite on crée une Liste qui contiendra le resultat de notre recherche qu'on devra retourner

#### Modele de la recherche des rapports

```
public function findReports($criteria){  
    $this->db->query('SELECT * FROM rapport WHERE date = :date ');  
    //Bind values  
    $this->db->bind(':date', $criteria);  
  
    $row = $this->db->resultSet();  
    //Execute  
    if($this->db->rowCount() > 0){  
        return $row;  
    }else{  
        return false;  
    }  
}
```

la fonction findReports prend un paramètre (\$criteria) qui sera un critère ensuite on rédige notre requête puis on définis notre critère qui sera une date

# GSB-Visite - Documentation technique

## Chaibi Yanis

### 9- Description du code recherchant les rapports

#### Contrôleur de la mise a jour des rapports

```
public function updateReport(){
    //Process form
    //Sanitize POST data
    $_POST = filter_input_array(INPUT_POST, FILTER_SANITIZE_STRING);

    //Init data
    $data = [
        'id' => $_POST['id'],
        'date' => $_POST['date'],
        'motif' => $_POST['motif'],
        'bilan' => $_POST['bilan']
    ];

    //Validate inputs
    if(empty($data['date']) || empty($data['motif']) || empty($data['bilan']) ){
        flash("updateReport", "Veuillez remplir tout les champs");
        redirect("../update.php");
    }

    //Modifie le rapport
    if($this->editModel->updateReports($data){
        redirect("../update.php");
    }else{
        die("Something went wrong");
    }
}
```

#### Modele de la mise a jour des rapports

```
public function updateReports($data){
    $this->db->query('UPDATE rapport SET date= :date, motif = :motif, bilan = :bilan WHERE id = :id' );
    //Bind values
    $this->db->bind(':id', $data['id']);
    $this->db->bind(':date', $data['date']);
    $this->db->bind(':motif', $data['motif']);
    $this->db->bind(':bilan', $data['bilan']);

    //Execute
    if($this->db->execute()){
        return true;
    }else{
        return false;
    }
}
```