

Documentation Analim

Mission 1 : Hôtellerie

Célia Baylet et Maxence Abric

1. Présentation du contexte

La Maison des Ligues de Lorraine (M2L) a pour mission de fournir des salles et des services aux différentes ligues sportives régionales et à d'autres structures hébergées.

La Maison des Ligues est un établissement du Conseil Régional. Elle est financée à 100% par le Conseil régional, bien qu'elle abrite un certain nombre de comités départementaux. Elle héberge la majorité des ligues sportives régionales, de la ligue de tennis (la plus grosse, qui emploie 8 personnes) à des ligues de sports qui n'ont pas d'employés permanents, comme le bowling ou la plongée sous-marine.

2. Objectifs de la mission

La maison des ligues de Lorraine organise en collaboration avec les autres régions françaises le [Congrès National des Maisons des Ligues].

Le congrès se déroule à Limoges [ANALIM] sur une semaine de 5 jours ouvrables. Il comporte différentes sessions d'une demi-journée chacune, consacrées à des conférences, débats sur un thème donné (exemple : "Tennis et stratégie", "Equipeement du bowling", "Tennis en fauteuil roulant", "Football féminin", etc.).

Pour donner plus de souplesse à l'organisation du congrès [ANALIM] il a été décidé que celle-ci doit être gérée grâce à une application web.

Mission 1 : Hôtellerie

Réalisation des tâches suivantes :

- T1.4 Gestion des hôtels : Cette tâche doit permettre la création, la modification, la suppression d'un hôtel à partir d'un formulaire. Il permet également d'obtenir la liste des hôtels.
- T2.6 Hôtellerie : Cette tâche doit permettre de loger un congressiste dans un hôtel en fonction du type d'hôtel souhaité, le congressiste peut demander à bénéficier du petit-déjeuner ce qui occasionne un supplément. Cette tâche n'est pas possible si la facture a déjà été imprimée.

Methode pour coder :

-utilisation d'un modèle MVC

-codé en php

-Visual Studio Code

MVC Hotel

Controleur Hotel

```
<?php  
  
include once "modele/dataBase.php";  
  
include once "modele/classHotel.php";  
  
include once "vue/entete.php";  
  
include once "vue/vueHotel.php";  
  
?>
```

On inclut le modele et vue de l'hotel.

Modele Hotel

```
class Hotel  
{  
  
    private String $id;  
  
    private String $nomHotel;  
  
    private int $prixParticipant;  
  
    private int $prixPetitDej;  
  
    private String $adresseHotel;  

```

```

private String $ville;

private String $codePostal;

private int $nbEtoiles;

private int $nbChambresPrises;

private int $nbChambresTotales;


public function construct(String $nomHotel = '', int
$prixParticipant = 0, int $prixPetitDej = 0, String $adresseHotel = '',
String $ville = '', String $codePostal = '', int $nbEtoiles = 0, int
$nbChambresPrises = 0, int $nbChambresTotales = 0)
{

    $this->nomHotel = $nomHotel;

    $this->prixParticipant = $prixParticipant;

    $this->prixPetitDej = $prixPetitDej;

    $this->adresseHotel = $adresseHotel;

    $this->ville = $ville;

    $this->codePostal = $codePostal;

    $this->nbEtoiles = $nbEtoiles;

    $this->nbChambresPrises = $nbChambresPrises;

    $this->nbChambresTotales = $nbChambresTotales;

}

```

On note les attributs de l'hôtel, puis on crée un constructeur qui permettra de créer un nouvel hôtel.

```

public function getIDHotel()

{

    return $this->id;
}

```

```

    }

    public function setIDHotel(int $idHotel)

    {

        $this->id = $idHotel;

    }

```

On crée les getters et setters de chaque attributs. Getter pour afficher, setter pour la modification.

```

public function getAllHotels($conn)

{

    include_once('dataBase.php');

    $conn = (new Database())->getConnexion();

    $result = $conn->prepare("SELECT * FROM hotel");

    $result->execute();

    return $result->fetchAll(PDO::FETCH_ASSOC);

}

```

Cette fonction sert à récupérer tous les hôtels pour les afficher.

```

public function getHotelById2($conn, $id)

{

    include_once('dataBase.php');

    $conn = (new Database())->getConnexion();

    $stmt = $conn->prepare("SELECT * FROM hotel WHERE id = ?");

    $stmt->bindValue(1, $id);

    $stmt->execute();

    return $stmt->fetch(PDO::FETCH_ASSOC);

}

```

Cette fonction sert à récupérer un hôtel en fonction de son id.

```
public function addHotel()

{

    include_once('dataBase.php');

    $conn = (new Database())->getConnexion();

    // Insérer un nouvel hôtel dans la base de données

    $sql = "INSERT INTO hotel (nomHotel, prixParticipant,
prixPetitDej, adresseHotel, ville, codePostal, nbEtoiles,
nbChambresTotales, nbChambresPrises) VALUES (?, ?, ?, ?, ?, ?, ?, ?,
?) ";

    $stmt = $conn->prepare($sql);

    $stmt->bindValue(1, $this->nomHotel);

    $stmt->bindValue(2, $this->prixParticipant);

    $stmt->bindValue(3, $this->prixPetitDej);

    $stmt->bindValue(4, $this->adresseHotel);

    $stmt->bindValue(5, $this->ville);

    $stmt->bindValue(6, $this->codePostal);

    $stmt->bindValue(7, $this->nbEtoiles);

    $stmt->bindValue(8, $this->nbChambresTotales);

    $stmt->bindValue(9, 0); // Valeur par défaut pour
nbChambresPrises

    if ($stmt->execute()) {

        return "Succès"; // Ou renvoyez un message de succès

    } else {

        return "Échec"; // Ou renvoyez un message d'erreur

    }

}
```

Cette fonction sert à insérer un hôtel dans la base de données. Avec une requête 'insert into' et des bindValue pour chaque attribut. On renvoie un message de succès ou d'erreur pour savoir si l'ajout a fonctionné.

```
public function updateHotel($id, $nomHotel, $prixParticipant,
    $prixPetitDej, $adresseHotel, $codePostal, $ville, $nbEtoiles)
{
    include_once('dataBase.php');

    $conn = (new Database())->getConnexion();

    $req = $conn->prepare("UPDATE hotel SET nomHotel = ?,
    prixParticipant = ?, prixPetitDej = ?, adresseHotel = ?, ville = ?,
    codePostal = ?, nbEtoiles = ? WHERE id = ?");

    $req->bindValue(1, $nomHotel);

    $req->bindValue(2, $prixParticipant);

    $req->bindValue(3, $prixPetitDej);

    $req->bindValue(4, $adresseHotel);

    $req->bindValue(5, $ville);

    $req->bindValue(6, $codePostal);

    $req->bindValue(7, $nbEtoiles);

    $req->bindValue(8, $id);

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

On modifie l'hôtel choisi en fonction de son id. L'id n'est pas modifié car c'est une clé primaire.

```
public function getLesHotels()
{
    include "bdd.php";
```

```

$sql = "SELECT * FROM hotel";

$prep = $conn->prepare($sql);

$prep->execute();

$lesHotels = $prep->fetchAll(PDO::FETCH_OBJ);

return $lesHotels;
}

```

Cette fonction sert à récupérer tous les hôtels pour que le congressiste puisse sélectionner son hôtel dans une liste déroulante.

```

public function deleteHotel($id)

{

    include_once('dataBase.php');

    $conn = (new Database())->getConnexion();

    $sql = "DELETE FROM hotel WHERE id = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bindValue(1, $id);

    if ($stmt->execute()) {

        return "Succès"; // Ou renvoyez un message de succès

    } else {

        return "Échec"; // Ou renvoyez un message d'erreur

    }

}

```

Cette fonction sert à supprimer un hôtel en fonction de son id. Un message de succès ou d'erreur est affiché.

```

public function getHotelAsOption() {

```

```

include_once("bdd.php"); // Connexion à la base de données

$sql = "SELECT * FROM hotel";

$stmt = $conn->prepare($sql);

$stmt->execute();

return $stmt->fetchAll(PDO::FETCH_OBJ); // Retourne un tableau
d'objets représentant les hôtels
}

```

Cette fonction sert à l'ajout d'un hôtel à un congressiste. Elle est appelée dans le contrôleur d'ajout d'un congressiste.

Cette fonction récupère tous les hôtels depuis la base de données et retourne leurs données sous forme d'un tableau d'objets, généralement pour les utiliser comme options dans un formulaire ou une liste déroulante.

Vue Hotel

Nom	Prix du Participant	Prix du Petit Déjeuner	Adresse	Code Postal	Ville	Nombre d'étoiles	Nombre de chambres prises	Nombre de chambres totales	Actions	
L'hotel Parfait	158	18	15 rue des Lilas	87200	Saint Junien	★ ★ ★	0	200	Supprimer	Modifier
test	15	15	12 rue test	87555	Bordeaux	★ ★ ★ ★	0	20000	Supprimer	Modifier
test2	605	20	12 rue test	87555	Bordeaux	★ ★ ★ ★ ★	0	20000	Supprimer	Modifier
L'hotel Parfait	160	46	15 rue test	87555	fdcsx	★ ★ ★ ★	0	20000	Supprimer	Modifier
test	596	41	ffchbv	87000	limogess	★ ★ ★	0	56	Supprimer	Modifier

Nom de l'hôtel

Prix participant

Prix petit déjeuner

Adresse

Code postal

Ville

Nombre d'étoiles

1 étoiles ▼

Nombre de chambres totales

Ajouter un hôtel

```
$hotelModel = new Hotel() ;
```

```
$conn = new Database() ;
```

// Création de la connexion à la base de données

// Injection de la connexion dans l'objet Hotel.

```
// Ajouter un hôtel
```

```
if (isset($_POST["ajouter"])) {
```

```
    $nomHotel = $_POST["nomHotel"];
```

```

    $prixParticipant = $_POST["prixParticipant"];

    $prixPetitDej = $_POST["prixPetitDej"];

    $adresseHotel = $_POST["adresseHotel"];

    $codePostal = $_POST["codePostal"];

    $ville = $_POST["ville"];

    $nbEtoiles = $_POST["nbEtoiles"];

    // Validation du nombre d'étoiles

    if (!is_numeric($nbEtoiles) || $nbEtoiles < 1 || $nbEtoiles > 5) {

        echo "Nombre d'étoiles invalide. Veuillez choisir un nombre
entre 1 et 5.";

        exit();

    }

    $nbChambresPrises = 0;

    $nbChambresTotales = $_POST["nbChambresTotales"];

    $hotel = new Hotel($nomHotel, $prixParticipant, $prixPetitDej,
$adresseHotel, $ville, $codePostal, $nbEtoiles, $nbChambresPrises,
$nbChambresTotales); // instance de la classe

    $hotel->addHotel();

}

```

Le nombre de chambres prises est initialisé à 0. On affecte les données à un hotel.

Ce code ajoute un nouvel hôtel à la base de données en validant les données du formulaire, notamment le nombre d'étoiles, et crée une instance de la classe **Hotel** pour enregistrer les informations.

L'instance de la classe utilise les valeurs saisies dans le formulaire pour initialiser l'objet, qui sera ensuite utilisé pour appeler la méthode **addHotel()** afin d'enregistrer les données.

```
// Supprimer un hôtel

if (isset($_POST["supprimer"]) && isset($_POST["hotelId"])) {

    $id = $_POST["hotelId"];

    $hotelModel->deleteHotel($id);

}
```

Ce code vérifie si un formulaire a soumis les données pour supprimer un hôtel (**supprimer** et **hotelId** sont définis) et appelle la méthode **deleteHotel(\$id)** de **\$hotelModel** pour supprimer l'hôtel correspondant à l'ID fourni.

```
// Modifier un hôtel

if (isset($_POST["modifier"]) && isset($_POST["hotelIdM"])) {

    $id = $_POST["hotelIdM"];

    $nomHotel = $_POST["nomHotel"];

    $prixParticipant = $_POST["prixParticipant"];

    $prixPetitDej = $_POST["prixPetitDej"];

    $adresseHotel = $_POST["adresseHotel"];

    $codePostal = $_POST["codePostal"];

    $ville = $_POST["ville"];

    $nbEtoiles = $_POST["nbEtoiles"];

    $hotelModel->updateHotel($id, $nomHotel, $prixParticipant,
    $prixPetitDej, $adresseHotel, $codePostal, $ville, $nbEtoiles);

}

$hotelList = $hotelModel->getAllHotels($conn);
```

```
?>
```

Ce code permet de modifier les informations d'un hôtel dans la base de données en fonction des données modifiées dans le formulaire, puis de récupérer et afficher la liste de tous les hôtels.

```
<form method="post">
```

On envoie des données au serveur en utilisant la méthode POST

```
<table>

    <tr>

        <th>Nom</th>

        <th>Prix du Participant</th>

        <th>Prix du Petit Déjeuner</th>

        <th>Adresse</th>

        <th>Code Postal</th>

        <th>Ville</th>

        <th>Nombre d'étoiles</th>

        <th>Nombre de chambres prises</th>

        <th>Nombre de chambres totales</th>

        <th>Actions</th>

    </tr>

    <?php foreach ($hotelList as $hotel) : ?>

        <tr>

            <td><?= $hotel['nomHotel'] ?></td>

            <td><?= $hotel['prixParticipant'] ?></td>
```

```

        <td><?= $hotel['prixPetitDej'] ?></td>

        <td><?= $hotel['adresseHotel'] ?></td>

        <td><?= $hotel['codePostal'] ?></td>

        <td><?= $hotel['ville'] ?></td>

        <td>

            <?php for ($i = 0; $i < $hotel['nbEtoiles']; $i++)
echo "★"; ?>

        </td>

        <td><?= $hotel['nbChambresPrises'] ?></td>

        <td><?= $hotel['nbChambresTotales'] ?></td>

        <td>

            <button type="submit" name="supprimer"
value="Supprimer">Supprimer</button>

            <input type="hidden" name="hotelId" value="<?=
$hotel['id'] ?>">

            <button type="submit" name="afficherDetails"
value="<?= $hotel['id'] ?>">Modifier</button>

        </td>

    </tr>

    <?php endforeach; ?>

</table>

```

C'est le code pour l'affichage de tous les hôtels. Ce code génère un tableau, affichant la liste des 'hôtels à partir des données de **\$hotelList**. Chaque ligne du tableau représente un hôtel, avec des colonnes pour ses détails comme son nom, prix, adresse, ville, nombre d'étoiles, et des informations sur les chambres. Dans la dernière colonne, des boutons permettent de supprimer ou modifier les informations de l'hôtel, en utilisant son identifiant caché pour les actions côté serveur.

```
<?php

if (isset($_POST["afficherDetails"])) {

    $id = $_POST["afficherDetails"];

    $hotelDetails = $hotelModel->getHotelById2($conn, $id);

    if ($hotelDetails) {
?>

        <h1>Modifier un hôtel</h1>

        <form action="" method="post">

            <input type="hidden" name="hotelIdM" value="<?=$hotelDetails['id'] ?>">

            <label for="nomHotel">Nom de l'hôtel</label>

            <input type="text" name="nomHotel" id="nomHotel" value="<?=$hotelDetails['nomHotel'] ?>"><br>

            <label for="prixParticipant">Prix participant</label>

            <input type="text" name="prixParticipant" id="prixParticipant" value="<?=$hotelDetails['prixParticipant'] ?>"><br>

            <label for="prixPetitDej">Prix petit déjeuner</label>

            <input type="text" name="prixPetitDej" id="prixPetitDej" value="<?=$hotelDetails['prixPetitDej'] ?>"><br>

            <label for="adresseHotel">Adresse</label>

            <input type="text" name="adresseHotel" id="adresseHotel" value="<?=$hotelDetails['adresseHotel'] ?>"><br>

            <label for="codePostal">Code postal</label>

            <input type="text" name="codePostal" id="codePostal" value="<?=$hotelDetails['codePostal'] ?>"><br>

            <label for="ville">Ville</label>
```

```

        <input type="text" name="ville" id="ville" value="<?=
$hotelDetails['ville'] ?>"><br>

        <label for="nbEtoiles">Nombre d'étoiles</label>

        <select name="nbEtoiles" id="nbEtoiles">

            <?php for ($i = 1; $i <= 5; $i++) : ?>

                <option value="<?= $i ?>" <?=
$hotelDetails['nbEtoiles'] == $i ? 'selected' : '' ?>><?= $i ?>
étoiles</option>

            <?php endfor; ?>

        </select><br>

        <input type="submit" name="modifier" value="Modifier un
hôtel">

    </form>

    <a href="index.php">Retour à la liste</a>

<?php
    }

} else {
?>

<h1>Ajouter un hôtel</h1>

<form action="" method="post">

    <label for="nomHotel">Nom de l'hôtel</label>

    <input type="text" name="nomHotel" id="nomHotel"><br>

    <label for="prixParticipant">Prix participant</label>

    <input type="text" name="prixParticipant"
id="prixParticipant"><br>

    <label for="prixPetitDej">Prix petit déjeuner</label>

```

```

        <input type="text" name="prixPetitDej" id="prixPetitDej"><br>

        <label for="adresseHotel">Adresse</label>

        <input type="text" name="adresseHotel" id="adresseHotel"><br>

        <label for="codePostal">Code postal</label>

        <input type="text" name="codePostal" id="codePostal"><br>

        <label for="ville">Ville</label>

        <input type="text" name="ville" id="ville"><br>

        <label for="nbEtoiles">Nombre d'étoiles</label>

        <select name="nbEtoiles" id="nbEtoiles">

            <?php for ($i = 1; $i <= 5; $i++) : ?>

                <option value="<?= $i ?>"><?= $i ?> étoiles</option>

            <?php endfor; ?>

        </select><br>

        <label for="nbChambresTotales">Nombre de chambres
        totales</label>

        <input type="text" name="nbChambresTotales"
        id="nbChambresTotales"><br>

        <input type="submit" name="ajouter" value="Ajouter un hôtel">

    </form>

<?php
}

```

Premier if : Vérifie si un identifiant d'hôtel est envoyé via le bouton "Modifier". Si oui, il récupère les détails de l'hôtel avec `getHotelById2` pour afficher un formulaire pré-rempli permettant de modifier ses informations.

Deuxième if : Si les détails de l'hôtel sont trouvés, il affiche un formulaire pré-rempli avec les données de l'hôtel sélectionné, permettant à l'utilisateur de les mettre à jour.

else : Si aucun hôtel n'est sélectionné pour modification, affiche un formulaire vierge pour permettre l'ajout d'un nouvel hôtel en saisissant ses informations.

MVC Hotel

Controleur Congressiste pour l'ajout

```
<?php
require "../modele/fonctionCongressiste.php";

require "../modele/classHotel.php";

include "../vue/entete.php";

$hotel = new Hotel();

$resultat = $hotel->getHotelAsOption();

include "../vue/vueAjoutCongressiste.php";

include "../vue/pied.php"

?>
```

On inclut le modele du congressiste et celui de l'hôtel. On inclut aussi la vue de l'ajout d'un congressiste à un hotel. Dans ce contrôleur, la fonction **getHotelAsOption()** est utilisée pour récupérer la liste des hôtels, afin de la transmettre à la vue **vueAjoutCongressiste.php**, où elle est affichée comme options dans un formulaire.

Contrôleur Congressiste pour la gestion(modification/suppression)

```
<?php

require "../modele/fonctionCongressiste.php";

require "../modele/classHotel.php";

$hotel = new Hotel();

$resultat = $hotel->getHotelAsOption();

include "../vue/entete.php";

include "../vue/vueGestionCongressiste.php";

include "../vue/pied.php"

?>
```

On inclut le modèle du congressiste et celui de l'hôtel. On inclut aussi la vue de la gestion d'un congressiste. Dans ce contrôleur, la fonction **getHotelAsOption()** est utilisée pour récupérer la liste des hôtels, afin de la transmettre à la vue **vueAjoutCongressiste.php**, où elle est affichée comme options dans un formulaire.

Contrôleur Congressiste pour l'affichage

```
<?php

require "../modele/fonctionCongressiste.php";

include "../vue/entete.php";

include "../vue/vueListeCongressiste.php";

include "../vue/pied.php"

?>
```

On inclut le modèle et la vue congressiste.

Model Congressiste (fonctions)

```
class Congressiste {  
  
    private string $nom;  
  
    private string $prenom;  
  
    private string $mail;  
  
    private string $adresse;  
  
    private string $ville;  
  
    private string $codePostal;  
  
    private string $tel;  
  
    private ?Hotel $unHotel; // Propriété nullable  
  
    private int $petitDej;  
  
    private int $nbEtoilesVoulues;  
  
  
    public function construct(  
  
        string $nom = '',  
  
        string $prenom = '',  
  
        string $mail = '',  
  
        string $adresse = '',  
  
        string $ville = '',  
  
        string $codePostal = '',  
  
        string $tel = '',  
  
        int $petitDej = 0,  
  
        int $nbEtoilesVoulues = 0,  
  
        Hotel $unHotel = null  
  
    ) {
```

```

        $this->nom = $nom;

        $this->prenom = $prenom;

        $this->mail = $mail;

        $this->adresse = $adresse;

        $this->ville = $ville;

        $this->codePostal = $codePostal;

        $this->tel = $tel;

        $this->petitDej = $petitDej;

        $this->nbEtoilesVoulues = $nbEtoilesVoulues;

        $this->unHotel = $unHotel;

    }

```

On a les attributs du congressiste avec ses informations personnelles (nom, adresse, etc.), ses préférences d'hôtel (petit déjeuner, nombre d'étoiles), et une association optionnelle à un objet **Hotel**. Le constructeur initialise ces propriétés avec des valeurs par défaut ou celles fournies lors de l'instanciation.

```

public function addUnCongressiste() {

    include "bdd.php";

    // Vérifier si l'hôtel est valide

    $hotel = $this->unHotel;

    if (empty($hotel->getIDHotel())) {

        throw new Exception("L'ID de l'hôtel ne peut pas être vide.");

    }

    // Vérifier si l'hôtel existe dans la base de données

    $sql = "SELECT COUNT(*) FROM hotel WHERE id = ?";

    $stmt = $conn->prepare($sql);

```

```

$stmt->bindValue(1, $hotel->getIDHotel());

$stmt->execute();

$hotelExist = $stmt->fetchColumn();

if ($hotelExist == 0) {

    throw new Exception("L'hôtel avec l'ID " . $hotel->getIDHotel() . " n'existe pas.");

}

// Insérer un congressiste dans la base de données

try {

    $sql = "INSERT INTO congressiste (nom, prenom, mail,
adresse, ville, codePostal, tel, id_Hotel, petitDej, nbEtoilesVoulues)

        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

    $stmt = $conn->prepare($sql);

    // Lier les valeurs aux paramètres

    $stmt->bindValue(1, $this->nom);

    $stmt->bindValue(2, $this->prenom);

    $stmt->bindValue(3, $this->mail);

    $stmt->bindValue(4, $this->adresse);

    $stmt->bindValue(5, $this->ville);

    $stmt->bindValue(6, $this->codePostal);

    $stmt->bindValue(7, $this->tel);

    $stmt->bindValue(8, $hotel->getIDHotel());

    $stmt->bindValue(9, $this->petitDej);

    $stmt->bindValue(10, $this->nbEtoilesVoulues);

    // Exécuter la requête

```

```

        $stmt->execute();

    } catch (Exception $e) {

        echo 'Erreur: ' . $e->getMessage();

    }

}

```

Connexion et validation de l'hôtel : Vérifie si l'ID de l'hôtel associé au congressiste est non vide, sinon lève une exception.

Vérification de l'existence de l'hôtel : Exécute une requête pour vérifier que l'hôtel avec l'ID fourni existe dans la base de données, sinon lève une exception.

Insertion du congressiste : Prépare une requête SQL pour insérer les informations du congressiste dans la table **congressiste**, incluant ses données personnelles et son hôtel associé.

Liaison des paramètres : Lie les propriétés de l'objet **Congressiste** et l'ID de l'hôtel aux paramètres de la requête préparée.

Exécution et gestion des erreurs : Exécute la requête d'insertion et affiche une erreur en cas d'exception.

```

public function ModifUnCongressiste($id) {

    include "bdd.php";

    // Requête de modification

    $sql = "UPDATE congressiste SET nom = ?, prenom = ?, mail = ?,
adresse = ?, ville = ?, codePostal = ?, tel = ?, id_Hotel = ?, petitDej
= ?, nbEtoilesVoulues = ? WHERE id = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bindValue(1, $this->nom);

    $stmt->bindValue(2, $this->prenom);

    $stmt->bindValue(3, $this->mail);

    $stmt->bindValue(4, $this->adresse);

```

```

        $stmt->bindValue(5, $this->ville);

        $stmt->bindValue(6, $this->codePostal);

        $stmt->bindValue(7, $this->tel);

        $stmt->bindValue(8, $this->unHotel ? $this->unHotel->getIDHotel() : null); // Vérifier si l'hôtel est défini

        $stmt->bindValue(9, $this->petitDej);

        $stmt->bindValue(10, $this->nbEtoilesVoulues);

        $stmt->bindValue(11, $id);

        $stmt->execute();

    }

```

Connexion et préparation de la requête : Connecte à la base de données et prépare une requête SQL pour mettre à jour les données d'un congressiste identifié par son ID.

Liaison des valeurs : Lie les propriétés du congressiste (nom, mail, hôtel, etc.) aux paramètres de la requête, en gérant l'hôtel comme optionnel.

Exécution de la requête : Exécute la requête pour mettre à jour les informations dans la base de données.

```

// Méthode pour mettre à jour l'hôtel du congressiste

public function updateHotelForCongressiste($congressisteId) {

    include "bdd.php";

    $hotel = $this->unHotel;

    $hotelId = $hotel ? $hotel->getIDHotel() : null;

    if (!$hotelId) {

        throw new Exception("Aucun hôtel n'est associé à ce congressiste.");
    }
}

```

```

    }

    $sql = "UPDATE congressiste SET id_Hotel = ? WHERE id = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bindValue(1, $hotelId);

    $stmt->bindValue(2, $congressisteId);

    $stmt->execute();

}

```

Validation de l'hôtel : Vérifie si un hôtel est associé au congressiste, sinon lève une exception.

Préparation et exécution de la requête : Prépare une requête SQL pour mettre à jour uniquement l'ID de l'hôtel associé à un congressiste spécifique, puis exécute la mise à jour.

```

public function SupprUnCongressiste($id) {

    include "bdd.php";

    $sql = "DELETE FROM congressiste WHERE id = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bindValue(1, $id);

    $stmt->execute();

}

```

Cette méthode supprime un congressiste de la base de données en utilisant son identifiant, en préparant une requête SQL.

```

public function recupUnCongressisteSansHotel() {

    include "bdd.php";

    $sql = "SELECT * FROM congressiste WHERE id_Hotel IS NULL";

    $stmt = $conn->prepare($sql);

```



```
$stmt->execute();

return $stmt->fetchAll(PDO::FETCH_OBJ);

}
```

Cette méthode récupère tous les congressistes sans hôtel associé (où `id_Hotel` est NULL) depuis la base de données, et retourne leurs données sous forme d'un tableau d'objets. Cette méthode sera utile pour ajouter un hôtel seulement aux congressistes qui n'en n'ont pas encore.

```
public function getUnCongressiste($id) {

    include "bdd.php";

    $sql = "SELECT * FROM congressiste WHERE id = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bindValue(1, $id);

    $stmt->execute();

    return $stmt->fetch(PDO::FETCH_OBJ);

}
```

Cette méthode récupère les informations d'un congressiste spécifique depuis la base de données, en utilisant son identifiant, et retourne les données sous forme d'objet.

```
public function getAllCongressistes() {

    include "bdd.php";

    $sql = "SELECT * FROM congressiste";

    $stmt = $conn->prepare($sql);

    $stmt->execute();

    return $stmt->fetchAll(PDO::FETCH_OBJ);

}
```

Cette méthode récupère tous les congressistes de la base de données et retourne leurs informations sous forme d'un tableau d'objets.

```

public function attribuerHotel($idCongressiste, $idHotel, $petitDej){

    include "bdd.php";

    // Vérifier si l'hôtel existe

    $sql = "SELECT COUNT(*) FROM hotel WHERE id = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bindValue(1, $idHotel);

    $stmt->execute();

    $hotelExist = $stmt->fetchColumn();

    if ($hotelExist == 0) {

        throw new Exception("L'hôtel avec l'ID " . $idHotel . "
n'existe pas.");

    }

    // Mise à jour de l'hôtel et du petit déjeuner du congressiste

    $sql = "UPDATE congressiste SET id_Hotel = ?, petitDej = ?
WHERE id = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bindValue(1, $idHotel);

    $stmt->bindValue(2, $petitDej);

    $stmt->bindValue(3, $idCongressiste);

    if ($stmt->execute()) {

        echo "Hôtel attribué avec succès.";

    } else {

        echo "Échec de l'attribution de l'hôtel.";

    }

}

```

```
}
```

Vérification de l'existence de l'hôtel : Vérifie si l'hôtel avec l'ID donné existe dans la base de données. Sinon, lève une exception.

Mise à jour des informations du congressiste : Prépare et exécute une requête pour attribuer l'hôtel et le petit déjeuner au congressiste, en mettant à jour ses informations.

Exécution et retour du résultat : Exécute la mise à jour et affiche un message de succès ou d'échec en fonction du résultat de l'opération.

Vue Congressiste pour l'ajout

Choisissez un Congressiste

BERNARD alex ▼

Afficher

Nom:

BERNARD

Prénom:

alex

Adresse:

rue des arbres

Code Postal:

87100

Ville:

limoges

Téléphone:

0777777997

Mail:

alex@gmail.com

Choisissez un Hôtel:

Lhotel Parfait - ★★ ★ ▼

Petit Déjeuner ☐

Attribuer Hôtel

```
// Récupération de tous les congressistes
```

```
$congressisteObj = new Congressiste();
```

```
$resultat = $congressisteObj->getAllCongressistes();
```

Récupération de tous les congressistes : Crée un objet **Congressiste**, puis appelle la méthode **getAllCongressistes()** pour récupérer tous les congressistes depuis la base de données.

```
// Vérifier si un congressiste a été sélectionné via le formulaire

$selectedCongressiste = null;

if (isset($_POST['afficher']) && !empty($_POST['id'])) {

    $id = $_POST['id'];

    $selectedCongressiste = $congressisteObj->getUnCongressiste($id);

}
```

Vérification d'un congressiste sélectionné : Si un **ID** est envoyé via le formulaire, on récupère les informations du congressiste correspondant avec la méthode **getUnCongressiste()**.

```
// Gestion de l'attribution d'un hôtel

if (isset($_POST['ajouterHotel'])) {

    $idCongressiste = $_POST['idCongressiste'];

    $idHotel = $_POST['idHotel'];

    $petitDej = isset($_POST['petitDej']) ? 1 : 0; // Si le petit
    déjeuner est sélectionné
```

Récupération des données du formulaire : Si le formulaire pour ajouter un hôtel est soumis, on récupère l'ID du congressiste, l'ID de l'hôtel et le petit déjeuner si il est sélectionné.

```
// Logique pour attribuer l'hôtel au congressiste

$congressisteObj->attribuerHotel($idCongressiste, $idHotel,
$petitDej);

}
```

Attribution de l'hôtel : Appelle la méthode **attribuerHotel()** de l'objet **Congressiste** pour associer l'hôtel et le petit déjeuner au congressiste sélectionné.

```
<!-- Formulaire de sélection de congressiste -->

<form method="post" action="">
```

```

<label for="id">Choisissez un Congressiste</label>

<select name="id" required>

    <?php

        // Récupération des congressistes sans hôtel

        $congressistes = $congressisteObj-
>recupUnCongressisteSansHotel();

        if ($congressistes) {

            foreach ($congressistes as $congressiste) {

                $selected = isset($selectedCongressiste) &&
                $selectedCongressiste->id == $congressiste->id ? 'selected' : '';

                echo "<option value='{$congressiste->id}'
                $selected>{$congressiste->nom} {$congressiste->prenom}</option>";

            }

        } else {

            echo "<option>Aucun congressiste trouvé</option>";

        }

    ?>

</select>

<button type="submit" name="afficher">Afficher</button>

</form>

```

Ce code génère un formulaire permettant de sélectionner un congressiste dans une liste déroulante. Il récupère les congressistes qui n'ont pas encore d'hôtel attribué en appelant la méthode **recupUnCongressisteSansHotel()**, puis affiche leur nom et prénom comme options. Si un congressiste est sélectionné, l'option correspondante est marquée comme "sélectionnée". Lorsque le bouton 'afficher' est cliqué on affiche les détails du congressiste sélectionné.

```

<!-- Affichage des détails du congressiste sélectionné -->

```

```
<?php if (isset($selectedCongressiste)) : ?>

    <form method="post" action="">

        <input type="hidden" name="idCongressiste" value="<?=
$selectedCongressiste->id ?>">

        <label for="nom">Nom:</label>

        <input type="text" id="nom" value="<?= $selectedCongressiste-
>nom ?>" readonly>

        <label for="prenom">Prénom:</label>

        <input type="text" id="prenom" value="<?=
$selectedCongressiste->prenom ?>" readonly>

        <label for="adresse">Adresse:</label>

        <input type="text" id="adresse" value="<?=
$selectedCongressiste->adresse ?>" readonly>

        <label for="codePostal">Code Postal:</label>

        <input type="text" id="codePostal" value="<?=
$selectedCongressiste->codePostal ?>" readonly>

        <label for="ville">Ville:</label>

        <input type="text" id="ville" value="<?= $selectedCongressiste-
>ville ?>" readonly>

        <label for="tel">Téléphone:</label>

        <input type="text" id="tel" value="<?= $selectedCongressiste-
>tel ?>" readonly>
```

```

<label for="mail">Mail:</label>

<input type="email" id="mail" value="<?= $selectedCongressiste-
>mail ?>" readonly>

```

Ce code affiche un formulaire en lecture seule avec les détails du congressiste sélectionné (nom, prénom, adresse, etc.) en remplissant les champs avec les informations correspondantes. Le formulaire inclut un champ caché contenant l'ID du congressiste pour un traitement ultérieur.

```

<label for="idHotel">Choisissez un Hôtel:</label>

<select name="idHotel" required>

    <?php

        // Récupération des hôtels depuis la base de données

        $hotelObj = new Hotel();

        $hotels = $hotelObj->getLesHotels(); // Récupération des
hôtels

        foreach ($hotels as $hotel) {

            echo "<option value='{ $hotel->id}'>" . $hotel->nomHotel
. " - " . str_repeat("★", $hotel->nbEtoiles) . "</option>";

        }

    ?>

</select>

```

Ce code génère un menu déroulant pour sélectionner un hôtel, en affichant le nom de chaque hôtel suivi du nombre d'étoiles sous forme d'icônes. Il récupère les hôtels depuis la base de données en utilisant la méthode `getLesHotels()` de la classe `Hotel`.

```

<label>

```



```

        Petit Déjeuner

        <input type="checkbox" name="petitDej" value="1" <?=$selectedCongressiste->petitDej ? 'checked' : '' ?>>

    </label>

```

Ce code crée une case à cocher pour indiquer si un petit déjeuner est inclus pour le congressiste. Si la propriété `petitDej` du congressiste sélectionné est vraie, la case sera cochée par défaut.

```

<button type="submit" name="ajouterHotel">Attribuer Hôtel</button>

```

On a un bouton "Attribuer Hôtel" qui soumet le formulaire pour attribuer un hôtel et le petit déjeuner au congressiste sélectionné

Vue Congressiste pour la gestion(modification/suppression)

Nom	Prénom	Adresse	Code Postal	Ville	Téléphone	Email	Petit Déjeuner	Actions
dupond	emma	rue des lilas	87150	limogess	0711111113	emmad@gmail.com	☒	Modifier Supprimer
dupond	marcel	rue des fleurs	87200	saint junien	0733333333	marcel.durand@gmail.com	☒	Modifier Supprimer
DUPONT	agathe	rue de saint jacques	87700	axe sur vienne	0744444444	agathe@gmail.com	☒	Modifier Supprimer
TEST	agathe	rue testttt	33000	limoges	0777777777	test@gmail.com	☒	Modifier Supprimer
BERNARD	alex	rue des arbres	87100	limoges	0777777997	alex@gmail.com	☐	Modifier Supprimer
MARTIN	Lilou	rue des gagnants	87200	limoges	0777777123	lilou@gmail.com	☐	Modifier Supprimer

```

if (isset($ POST['supprimer'])) {

    $id = $ POST['id'];

    $congressiste = new Congressiste();

    $congressiste->SupprUnCongressiste($id);

}

```

Ce code vérifie si le formulaire contient la requête de suppression d'un congressiste, puis crée une instance de la classe `Congressiste` et appelle la méthode `SupprUnCongressiste` pour supprimer le congressiste correspondant à l'ID spécifié.

```

if (isset($_POST['modifier'])) {

```

```
$id = $_POST['id'];

$nom = $_POST['nom'];

$prenom = $_POST['prenom'];

$mail = $_POST['mail'];

$adresse = $_POST['adresse'];

$ville = $_POST['ville'];

$codePostal = $_POST['CP']; // Correspond au modèle "codePostal"

$tel = $_POST['tel'];

$petitDej = isset($_POST['petitDejeuner']) ? 1 : 0; // Utilisation
de 1/0 pour respecter le modèle

$nbEtoilesVoulues = 0; // Si non modifiable ici, par défaut à 0 (ou
à définir selon vos besoins)

// Instanciation conforme au modèle

$congressiste = new Congressiste(

    $nom,

    $prenom,

    $mail,

    $adresse,

    $ville,

    $codePostal,

    $tel,

    $petitDej,

    $nbEtoilesVoulues

);

$congressiste->ModifUnCongressiste($id);

}
```



```

        <th>Actions</th>

    </tr>

</thead>

<tbody>

    <?php foreach ($resultat as $row) { ?>

        <form action="" method="post">

            <tr>

                <input type="hidden" name="id" value="<?php
echo $row->id; ?>">

                <td><input type="text" name="nom"
value="<?php echo $row->nom; ?>"></td>

                <td><input type="text" name="prenom"
value="<?php echo $row->prenom; ?>"></td>

                <td><input type="text" name="adresse"
value="<?php echo $row->adresse; ?>"></td>

                <td><input type="text" name="CP"
value="<?php echo $row->codePostal; ?>"></td> <!-- Changement en
"codePostal" -->

                <td><input type="text" name="ville"
value="<?php echo $row->ville; ?>"></td>

                <td><input type="text" name="tel"
value="<?php echo $row->tel; ?>"></td>

                <td><input type="text" name="mail"
value="<?php echo $row->mail; ?>"></td>

                <td>

                    <input type="checkbox"
name="petitDejeuner" value="1" <?php echo $row->petitDej == 1 ?
'checked="checked"' : ''; ?>>

                </td>

                <td>

```

```

                                <button type="submit"
name="modifier">Modifier</button>

                                <button type="submit"
name="supprimer">Supprimer</button>

                                </td>

                                </tr>

                                </form>

                                <?php } ?>

                                </tbody>

                                </table>

                                </div>

```

Table Head: Le tableau contient des en-têtes de colonnes pour afficher des informations sur les congressistes (Nom, Prénom, Adresse, etc.) ainsi que des actions (modifier et supprimer).

Formulaire & Table Rows: Chaque ligne du tableau correspond à un congressiste, et un formulaire est utilisé pour envoyer des données de modification ou de suppression. Chaque champ (Nom, Prénom, etc.) est prérempli avec les données actuelles du congressiste.

Checkbox Petit Déjeuner: Une case à cocher permet de spécifier si le congressiste a choisi un petit déjeuner, et elle est cochée si la valeur correspondante est 1.

Action Buttons: Des boutons "Modifier" et "Supprimer" sont fournis pour permettre de soumettre les modifications ou de supprimer un congressiste.

Vue Congressiste pour l'affichage

Nom	Prénom	Adresse	Code Postal	Ville	Téléphone	Email	Petit Déjeuner
dupond	emma	rue des lilas	87150	limogess	0711111113	emmadi@gmail.com	Oui
dupond	marcel	rue des fleurs	87200	saint junien	0733333333	marcel.durand@gmail.com	Oui
DUPONT	agathe	rue de saint jacques	87700	aixe sur vienne	0744444444	agathe@gmail.com	Oui
TEST	agathe	rue testtttt	33000	limoges	0777777777	test@gmail.com	Oui
BERNARD	alex	rue des arbres	87100	limoges	0777777997	alex@gmail.com	Non
MARTIN	Lilou	rue des gagnants	87200	limoges	0777777123	lilou@gmail.com	Non

```
$congressiste = new Congressiste();  
$resultat = $congressiste->getAllCongressistes();
```

Un objet **Congressiste** est instancié et assigné à la variable **\$congressiste**.

La méthode **getAllCongressistes()** est appelée pour récupérer tous les congressistes et les résultats sont stockés dans la variable **\$resultat**.

```
<table class="table-css">  
  
    <thead>  
  
        <tr>  
  
            <th>Nom</th>  
  
            <th>Prénom</th>  
  
            <th>Adresse</th>  
  
            <th>Code Postal</th>  
  
            <th>Ville</th>  
  
            <th>Téléphone</th>  
  
            <th>Email</th>  
  
            <th>Petit Déjeuner</th> <!-- Ajout de la colonne  
Petit Déjeuner -->  
  
        </tr>  
  
    </thead>  
  
    <tbody>  
  
        <?php foreach ($resultat as $row): ?>  
  
            <tr>  
  
                <td><?php echo $row->nom; ?></td>  
  
                <td><?php echo $row->prenom; ?></td>
```

```

        <td><?php echo $row->adresse; ?></td>

        <td><?php echo $row->codePostal; ?></td> <!--
Correction de "CP" en "codePostal" -->

        <td><?php echo $row->ville; ?></td>

        <td><?php echo $row->tel; ?></td>

        <td><?php echo $row->mail; ?></td>

        <td><?php echo $row->petitDej == 1 ? 'Oui' :
'Non'; ?></td> <!-- Ajout de l'affichage de l'état du petit déjeuner --
>

    </tr>

    <?php endforeach; ?>

</tbody>

</table>

```

Tableau HTML : Le tableau affiche les informations des congressistes, avec des colonnes pour le nom, prénom, adresse, etc., y compris une colonne pour le petit déjeuner.

Boucle PHP : Pour chaque congressiste dans `$resultat`, une ligne de tableau est générée avec ses informations et l'état du petit déjeuner (Oui/Non) basé sur la valeur de `petitDej`.