

# 1. Architecture applicative de l'application Web

## Principes d'organisation de l'application PHP Gsb-AppliFrais

Nous ne présentons pas ici les avantages de la structuration du code relevant de l'architecture Modèle-Vue-Contrôleur ; de nombreux documents se penchent sur la question.

Des frameworks (Zend, Symfony, Laravel) fournissent les classes mettant en œuvre cette technologie. Nous avons fait le choix ici de faire le travail « à la main »

Le langage HTML respectera la norme HTML5. Les pages seront validées à l'aide du validateur du W3C : <http://validator.w3.org/check>.

Les règles de style respecteront la norme CSS3. Elles seront validées auprès du validateur <http://jigsaw.w3.org/css-validator/>.

## Un développement guidé par les cas d'utilisation

C'est le propre de l'architecture MVC ; le système (l'application) doit répondre aux sollicitations de l'utilisateur. Les cas d'utilisation sont les moyens textuels de décrire ces sollicitations et les réponses.

Prenons l'exemple du cas d'utilisation suivant :

<b>PROJET</b> : Application web de gestion des frais	<b>Description cas d'utilisation</b>
<b>Nom cas d'utilisation</b> : Renseigner fiche de frais	
<b>Acteur déclencheur</b> : Visiteur médical	
<b>Pré conditions</b> : Visiteur médical authentifié	
<b>Post conditions</b> : néant	
<b>Scénario nominal</b> : <ul style="list-style-type: none"><li>○ <b>L'utilisateur demande à saisir un ou plusieurs frais pour le mois courant.</b></li><li>○ Le système retourne les frais actuellement saisis – éléments forfaitisés et hors forfait - pour le mois courant.</li><li>○ <b>L'utilisateur modifie une ou des valeurs des frais au forfait et demande la validation.</b></li><li>○ Le système enregistre cette ou ces modifications et retourne ces valeurs à jour.</li><li>○ <b>L'utilisateur ajoute un nouveau frais hors forfait en renseignant les différents champs – date d'engagement, libellé, montant - et valide.</b></li><li>○ Le système enregistre la ligne de frais hors forfait.</li></ul>	
<b>Exceptions</b> :  2.a- C'est la première saisie pour le mois courant. Si ce n'est pas encore fait, le système clôt la fiche du mois précédent et crée une nouvelle fiche de frais avec des valeurs initialisées à 0. Retour à 3. 4.a. Une valeur modifiée n'est pas numérique : le système indique 'Valeur numérique attendue '. Retour à 3. 6.a Un des champs n'est pas renseigné : le système indique : 'Le champ date (ou libellé ou montant) doit être renseigné'. 6.b La date d'engagement des frais hors forfait est invalide : le système indique 'La date d'engagement doit être valide'. Retour à 5. 6.c La date d'engagement des frais hors forfait date de plus d'un an. Le système indique 'La date d'engagement doit se situer dans l'année écoulée'. Retour à 5. <b>7. L'utilisateur sélectionne un frais hors forfait pour suppression.</b> 8. Le système enregistre cette suppression après une demande de confirmation.	
<b>Contraintes</b> :	

L'utilisateur sollicite à 4 reprises le système (points 1, 3, 5 et 7 en italique gras). Le contrôleur (fichier spécifique) doit donc répondre à ces 4 sollicitations :

```

21 $action = filter_input(INPUT_GET, 'action', FILTER_SANITIZE_STRING);
22 switch ($action) {
23 + case 'saisirFrais': {...5 lines }
28 + case 'validerMajFraisForfait': {...9 lines }
37 + case 'validerCreationFrais': {...17 lines }
54 + case 'supprimerFrais': {...4 lines }
58 }
59 $lesFraisHorsForfait = $pdo->getLesFraisHorsForfait($idVisiteur, $mois);
60 $lesFraisForfait = $pdo->getLesFraisForfait($idVisiteur, $mois);
61 require 'vues/v_listeFraisForfait.php';
62 require 'vues/v_listeFraisHorsForfait.php';

```

Remarque : le code des cases a été plié ici pour se concentrer sur l'essentiel.

Pour chacune des sollicitations, le système réagit et agit en conséquence, par exemple pour la demande de saisie des frais :

```

21 $action = filter_input(INPUT_GET, 'action', FILTER_SANITIZE_STRING);
22 switch ($action) {
23 - case 'saisirFrais':
24 -     if ($pdo->estPremierFraisMois($idVisiteur, $mois)) {
25 -         $pdo->creeNouvellesLignesFrais($idVisiteur, $mois);
26 -     }
27 -     break;
28 + case 'validerMajFraisForfait': {...9 lines }
37 + case 'validerCreationFrais': {...17 lines }
54 + case 'supprimerFrais': {...4 lines }
58 }
59 $lesFraisHorsForfait = $pdo->getLesFraisHorsForfait($idVisiteur, $mois);
60 $lesFraisForfait = $pdo->getLesFraisForfait($idVisiteur, $mois);
61 require 'vues/v_listeFraisForfait.php';
62 require 'vues/v_listeFraisHorsForfait.php';

```

Le système teste (ligne 10) si c'est la première fois que l'utilisateur accède à cette demande de saisie de frais –cf extension 2.a- et va chercher en base (lignes 45-46) les données concernant les frais forfaitisés et non forfaitisés afin d'afficher les deux vues demandées (lignes 47-48). Ici, ces affichages sont communs aux autres cases.

Les deux vues affichées sont ici :

Renseigner ma fiche de frais du mois 08-2022

Eléments forfaitisés

Forfait Etape

12

Frais Kilométrique

562

Nuitée Hôtel

6

Repas Restaurant

25

Ajouter

Effacer

Vue « Liste des frais au forfait »

Descriptif des éléments hors forfait			
Date	Libellé	Montant	
12/08/2022	Achat de fleurs	29.00	<a href="#">Supprimer ce frais</a>
14/08/2022	Taxi / VTC	32.00	<a href="#">Supprimer ce frais</a>

#### Nouvel élément hors forfait

Date (jj/mm/aaaa):

Libellé

Montant :

[Ajouter](#) [Effacer](#)

Vue « Liste des frais hors forfait »

Nous avons fait le choix de présenter deux vues distinctes –nous aurions pu bien sûr mettre ce code dans un seul fichier– pour éventuellement réutiliser une de ces vues dans un autre cas d’utilisation. Dans cette architecture, l’affichage des vues est provoqué par un ordre include (ou require) nomVue.

Pour respecter l’indépendance des couches (vue, modèle), le modèle (fichier php) retourne des tableaux :

```

171 public function getLesFraisForfait($idVisiteur, $mois): array
172 {
173     $requetePrepare = $this->connexion->prepare(
174         'SELECT fraisforfait.id as idfrais, '
175         . 'fraisforfait.libelle as libelle, '
176         . 'lignefraisforfait.quantite as quantite '
177         . 'FROM lignefraisforfait '
178         . 'INNER JOIN fraisforfait '
179         . 'ON fraisforfait.id = lignefraisforfait.idfraisforfait '
180         . 'WHERE lignefraisforfait.idvisiteur = :unIdVisiteur '
181         . 'AND lignefraisforfait.mois = :unMois '
182         . 'ORDER BY lignefraisforfait.idfraisforfait'
183     );
184     $requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
185     $requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
186     $requetePrepare->execute();
187     return $requetePrepare->fetchAll();
188 }

```

La vue construit le code HTML à partir du tableau retourné :

```

19 <div class="row">
20     <h2>Renseigner ma fiche de frais du mois
21     <?php echo $numMois . '-' . $numAnnee ?>
22     </h2>
23     <h3>Eléments forfaitisés</h3>
24     <div class="col-md-4">
25         <form method="post"
26             action="index.php?uc=gererFrais&action=validerMajFraisForfait"
27             role="form">
28             <fieldset>
29                 <?php
30                 foreach ($lesFraisForfait as $unFrais) {
31                     $idFrais = $unFrais['idfrais'];
32                     $libelle = htmlspecialchars($unFrais['libelle']);

```

```

33     $quantite = $unFrais['quantite']; ?>
34     <div class="form-group">
35         <label for="idFrais"><?php echo $libelle ?></label>
36         <input type="text" id="idFrais"
37             name="lesFrais[<?php echo $idFrais ?>]"
38             size="10" maxlength="5"
39             value="<?php echo $quantite ?>"
40             class="form-control">
41     </div>
42     <?php
43     }
44     ?>
45     <button class="btn btn-success" type="submit">Ajouter</button>
46     <button class="btn btn-danger" type="reset">Effacer</button>
47 </fieldset>
48 </form>
49 </div>
50 </div>

```

## Fonctionnement de l'application

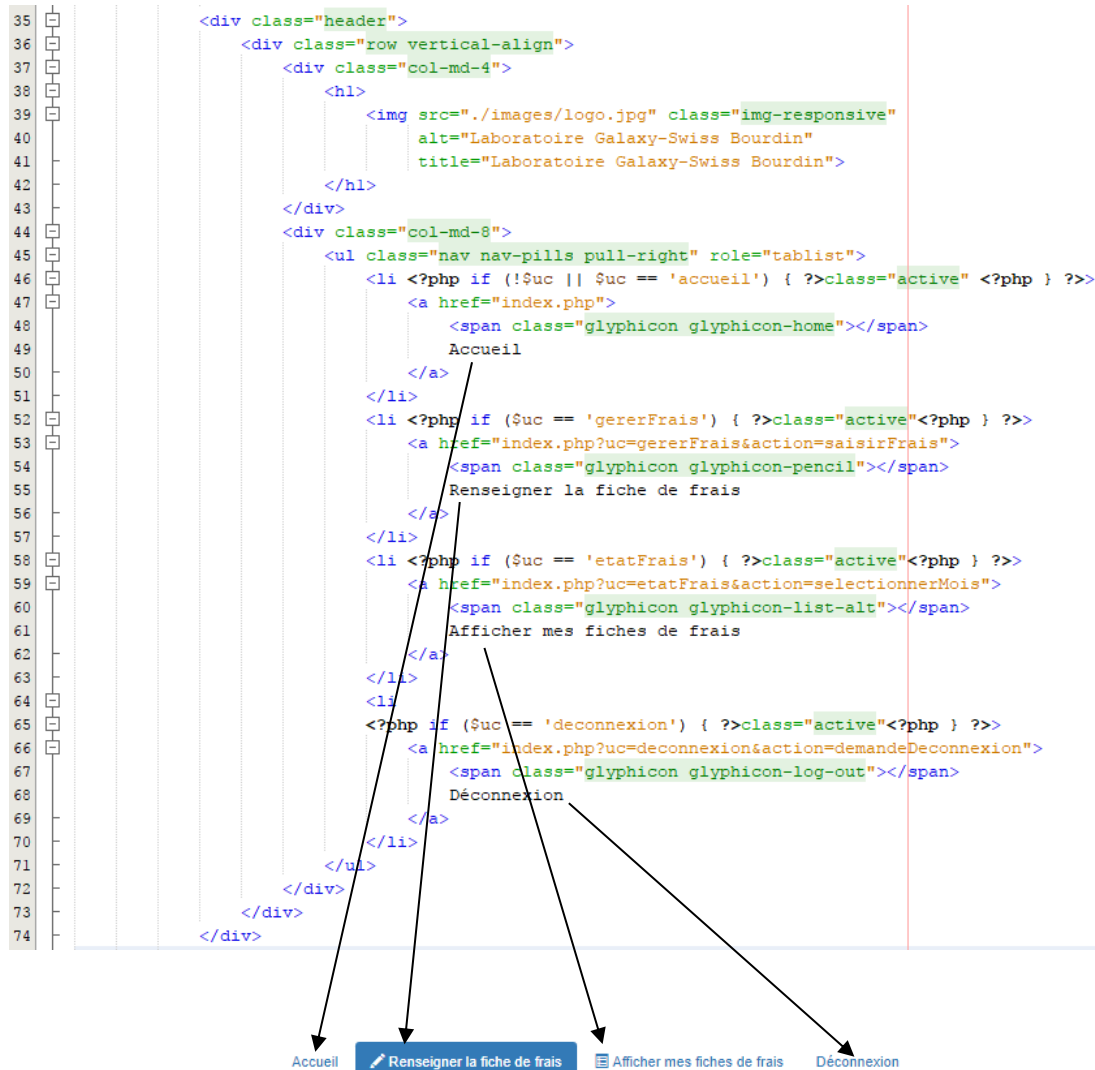
C'est la page index qui sert d'aiguilleur principal et oriente vers un contrôleur de cas d'utilisation :

```

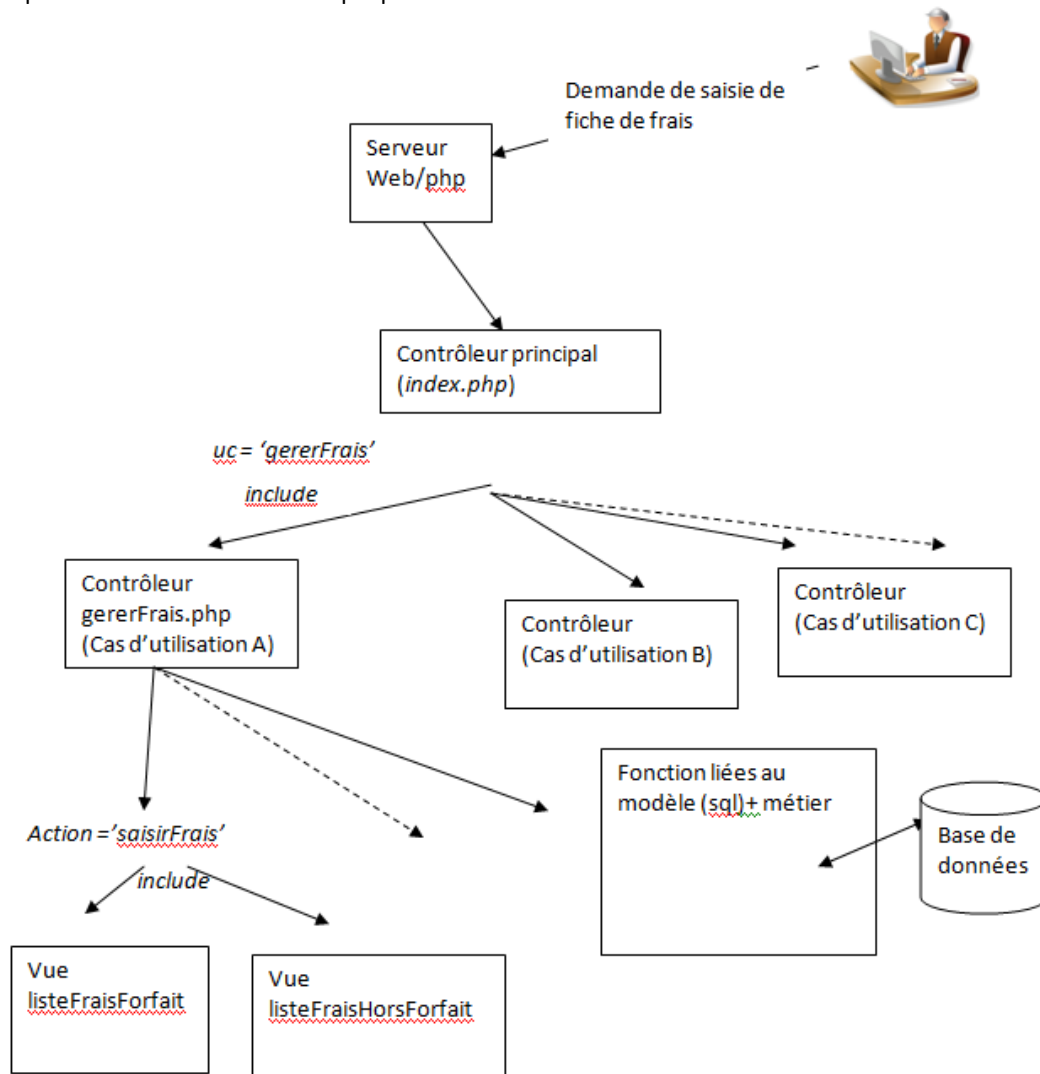
18 use Modeles\PdoGsb;
19 use Outils\Utilitaires;
20
21 require '../vendor/autoload.php';
22 require '../config/define.php';
23
24 session_start();
25
26 $pdo = PdoGsb::getPdoGsb();
27 $estConnecte = Utilitaires::estConnecte();
28
29 require PATH_VIEWS . 'v_entete.php';
30
31 $uc = filter_input(INPUT_GET, 'uc', FILTER_SANITIZE_FULL_SPECIAL_CHARS);
32
33 if ($uc && !$estConnecte) {
34     $uc = 'connexion';
35 } elseif (empty($uc)) {
36     $uc = 'accueil';
37 }
38
39 switch ($uc) {
40     case 'connexion':
41         include PATH_CTRL . 'c_connexion.php';
42         break;
43     case 'accueil':
44         include PATH_CTRL . 'c_accueil.php';
45         break;
46     case 'gererFrais':
47         include PATH_CTRL . 'c_gererFrais.php';
48         break;
49     case 'etatFrais':
50         include PATH_CTRL . 'c_etatFrais.php';
51         break;
52     case 'deconnexion':
53         include PATH_CTRL . 'c_deconnexion.php';
54         break;
55     default:
56         Utilitaires::ajouterErreur('Page non trouvée, veuillez vérifier votre lien...');
57         include PATH_VIEWS . 'v_erreurs.php';
58         break;
59 }
60 require PATH_VIEWS . 'v_pied.php';

```

Ceci est à associé à ce que l'utilisateur sélectionne dans le sommaire :

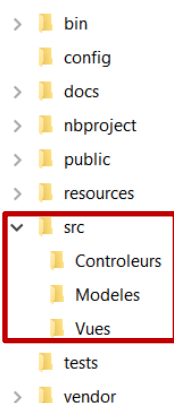


On peut résumer cette cinématique par un schéma :



Ainsi chaque page reçue est construite à partir de l'index comme une succession de fichiers include selon le cas d'utilisation. L'action demandée entraîne un traitement, à partir de la base de données et des règles métier (responsabilité de la couche Modèle) et expose les vues associées.

L'arborescence du site reflète cette architecture :



→ L'organisation des répertoires correspond aux bonnes pratiques PHP que l'on peut retrouver ici : [https://phptherightway.com/#common\\_directory\\_structure](https://phptherightway.com/#common_directory_structure).

## Utilisation d'une bibliothèque d'accès aux données

Php contient en standard une classe PDO, d'accès aux données ; elle a l'avantage d'être générique-indépendante du SGBD- et propose des services performants qui évitent la plupart du temps de faire soi-même les boucles de parcours des jeux d'enregistrements :


```
42 use PDO;
43 use PDOStatement;
44 use Outils\Utilitaires;
45
46 require '../config/bdd.php';
47
48 class PdoGsb
49 {
50     protected $connexion;
51     private static $instance = null;
52
53     /**
54      * Constructeur privé, crée l'instance de PDO qui sera sollicitée
55      * pour toutes les méthodes de la classe
56      */
57     private function __construct()
58     {
59         $this->connexion = new PDO(DB_DSN, DB_USER, DB_PWD);
60         $this->connexion->query('SET CHARACTER SET utf8');
61     }
62 }
```

Cette classe n'est pas associée au modèle MVC, mais elle allège grandement la gestion des données.

Dans l'application GSB, la classe PDO est encapsulée dans une classe PdoGsb.

## 2. Ebauches des formulaires

### Saisie des frais



[Accueil](#) [Renseigner la fiche de frais](#) [Afficher mes fiches de frais](#) [Déconnexion](#)

#### Renseigner ma fiche de frais du mois 08-2022

##### Eléments forfaitisés

**Forfait Etape**

**Frais Kilométrique**

**Nuitée Hôtel**


**Repas Restaurant**

[Ajouter](#) [Effacer](#)

---

Descriptif des éléments hors forfait			
Date	Libellé	Montant	
12/08/2022	Achat de fleurs	29.00	<a href="#">Supprimer ce frais</a>
14/08/2022	Taxi / VTC	32.00	<a href="#">Supprimer ce frais</a>

##### Nouvel élément hors forfait

**Date (jj/mm/aaaa):**  
 

**Libellé**

**Montant :**

[Ajouter](#) [Effacer](#)

Partie 4  
Documents

Page 39

AP GSB 2024



## Consultation des frais

[Accueil](#)[Renseigner la fiche de frais](#)[Afficher mes fiches de frais](#)[Déconnexion](#)

### Mes fiches de frais

Sélectionner un mois :

Mois :

08/2022

Proposer un calendrier pour la période

Valider

Effacer

Fiche de frais du mois 08-2022 :

Etat : Fiche créée, saisie en cours depuis le 29/08/2022  
Montant validé : 0.00

Date d'enregistrement de la dernière  
opération correspondant à la ligne

Éléments forfaitisés

Forfait Etape	Frais Kilométrique	Nuitées Hotel	Repas Restaurant
12	562	6	25


Descriptif des éléments hors forfait - 0 justificatifs reçus

Date	Libellé	Montant
12/08/2022	Achat de fleurs	29.00
14/08/2022	Taxi / VTC	32.00

Lignes remplies en fonction  
des données de la base

## Validation des frais (partie comptable avec un thème « orange »)

Accueil **✓ Valider les fiches de frais** Suivre le paiement des fiches de frais Déconnexion



Choisir le visiteur : Villechalane Louis Mois : 08/2022

**Valider la fiche de frais**

Éléments forfaitisés

Forfait Etape : 12

Frais Kilométrique : 562

Nuitée Hôtel : 6

Repas Restaurant : 25

Corriger Réinitialiser

Rempli automatiquement d'après les saisies des visiteurs et le mois choisi. Modifiable s'il y a contestation par le service comptable (qui pourra vérifier le nombre de rapports et les lieux de déplacement)

Descriptif des éléments hors forfait

Date	Libellé	Montant	
12/08/2022	Achat de fleurs	29.90	Corriger Réinitialiser
14/08/2022	Taxi	32.50	Corriger Réinitialiser

Nombre de justificatifs : 2

Valider Réinitialiser

Rempli automatiquement d'après les saisies des visiteurs et le mois choisi. Il y aura autant de lignes que de saisies "hors forfait" par le visiteur

La validation nécessite l'enregistrement de la date de l'opération.