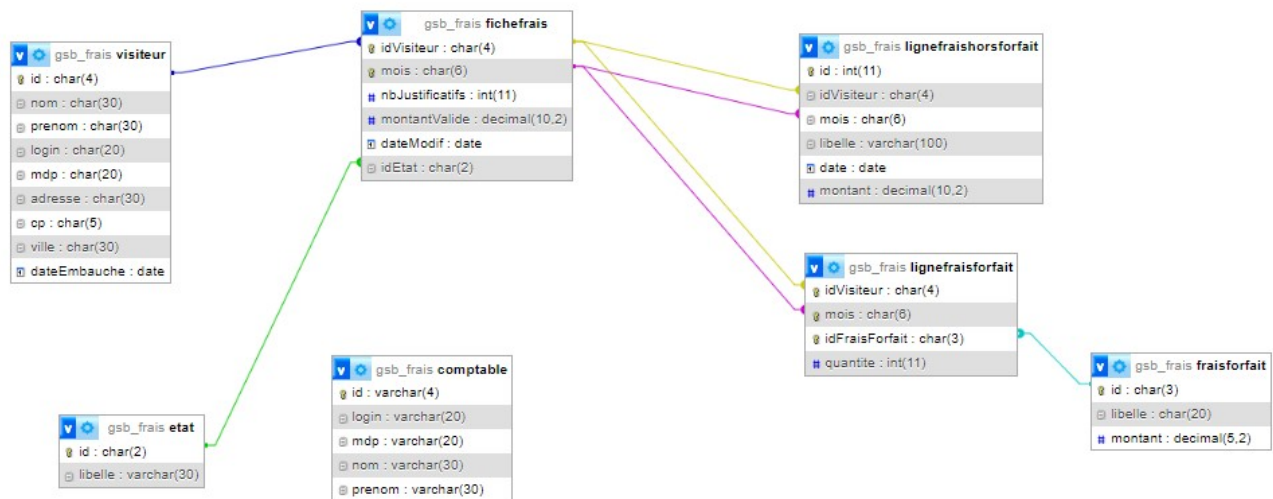


I. Présentation du contexte GSB

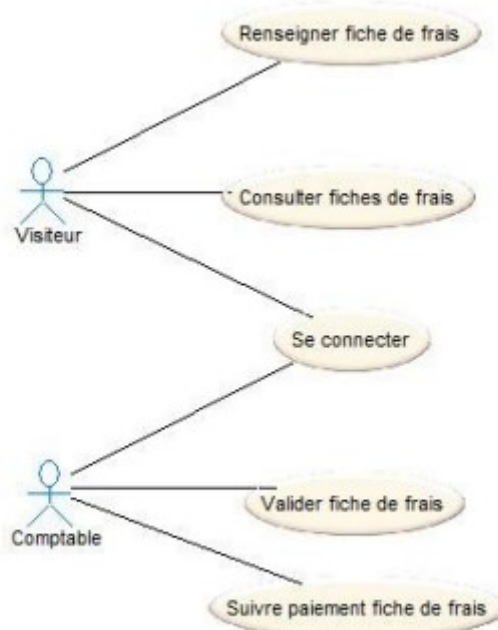
Le laboratoire Galaxy Swiss Bourdin (GSB) est issu de la fusion entre le géant américain Galaxy (spécialisé dans le secteur des maladies virales dont le SIDA et les hépatites) et le conglomérat européen Swiss Bourdin (travaillant sur des médicaments plus conventionnels), lui même déjà union de trois petits laboratoires. En 2009, les deux géants pharmaceutiques ont uni leurs forces pour créer un leader de ce secteur industriel. L'entité Galaxy Swiss Bourdin Europe a établi son siège administratif à Paris. Le siège social de la multinationale est situé à Philadelphie, Pennsylvanie, aux États-Unis.

II. Analyse

A. Modélisation des données



B. Les traitements



1. Connexion à l'application

Portail de connexion :



Identification utilisateur


 Login

 Mot de passe

poste au sein de l'entreprise : ☒ **Visiteur** ☐ **Comptable**

Se connecter


Accueil Comptable :




[Accueil](#) [Valider une fiche de frais](#) [Déconnexion](#)

Gestion des frais - Comptable : Jean Dupont

Navigation

 Valider une fiche de frais

Accueil Visiteur :




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

Gestion des frais - Visiteur : David Andre

Navigation

 Renseigner la fiche de frais

 Afficher mes fiches de frais

2. Renseignement d'une fiche de frais

Cas d'utilisations :

Page de saisie :



[Accueil](#)

[Renseigner la fiche de frais](#)

[Afficher mes fiches de frais](#)

[Déconnexion](#)

Renseigner ma fiche de frais du mois 09-2024

Eléments forfaitisés

Forfait Etape

Frais Kilométrique

Nuitée Hôtel

Repas Restaurant

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

Descriptif des éléments hors forfait

Date	Libellé	Montant	
------	---------	---------	--

Nouvel élément hors forfait

Date (jj/mm/aaaa):

Libellé

Montant :

€


Pièce justificatives :

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

3. Modification d'un compte rendu

Cas d'utilisation :

Page de choix du visiteur



[Accueil](#) [Valider une fiche de frais](#) [Déconnexion](#)

Sélectionner un visiteur :

Villechalane Louis

Villechalane Louis

Andre David

Bedos Christian

Tusseau Louis

Bentot Pascal

Bioret Luc

Bunisset Francis

Bunisset Denise

Cacheux Bernard

Cadic Eric

Charoze Catherine

Clepkens Christophe

Cottin Vincenne

Daburon François

De Philippe

Debelle Michel


Debelle Jeanne

Debroise Michel

Desmarquest Nathalie

Desnost Pierre

Page de choix du mois :



[Accueil](#) [Valider une fiche de frais](#) [Déconnexion](#)

Mes fiches de frais

Villechalane Louis

Sélectionner un mois :

Mois :

08/2024

Valider

Annuler

Page de saisie :

Eléments forfaitisés

Forfait Etape	Frais Kilométrique	Nuitée Hôtel	Repas Restaurant
8	730	13	7

Confirmer les modifications

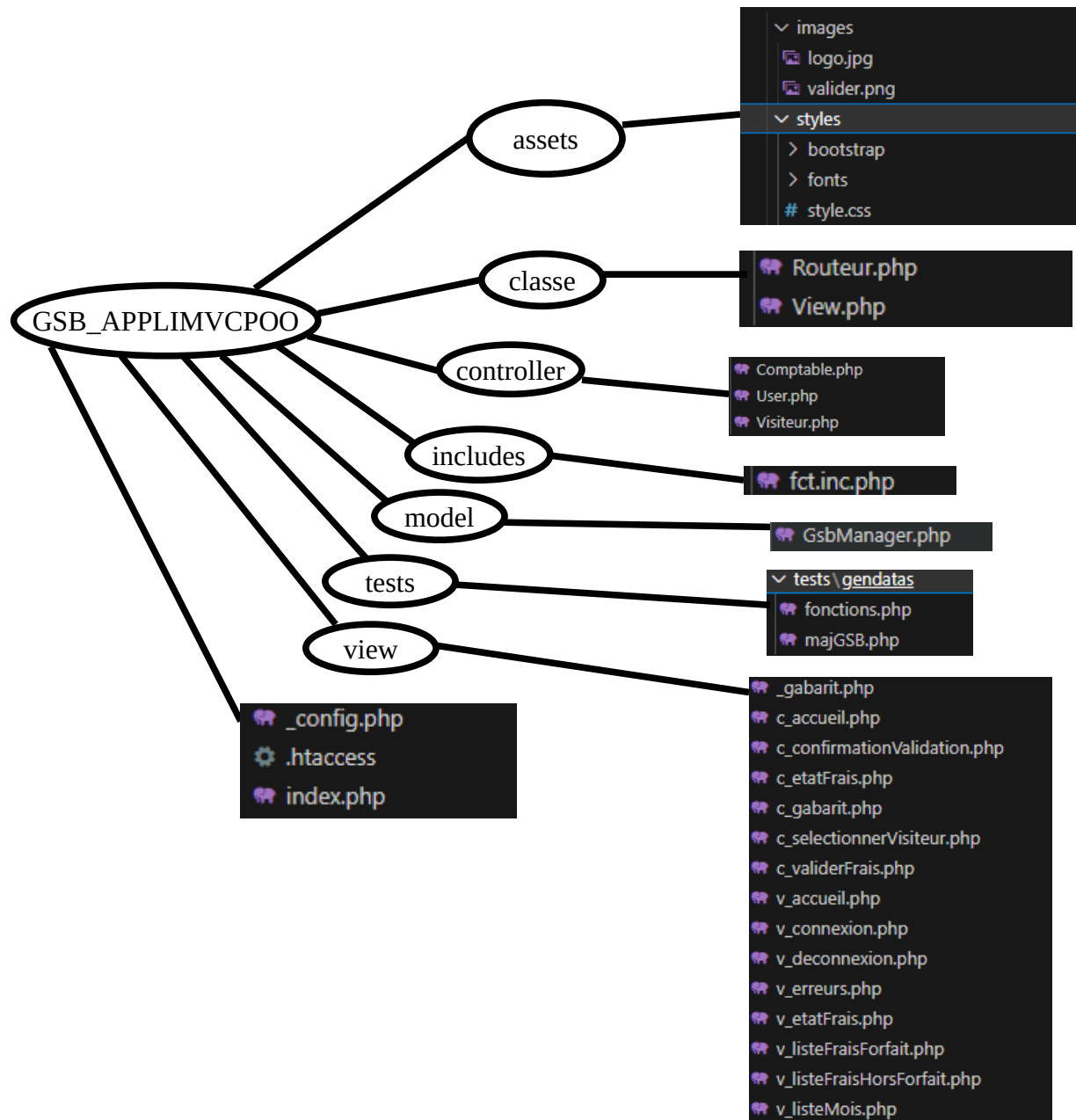
Descriptif des éléments hors forfait - 10 justificatifs reçus

Date	Libellé	Montant	
12/08/2024	Voyage SNCF	133.00	<div>Reporter</div> <div>Refuser</div>
22/08/2024	Repas avec praticien	32.00	<div>Reporter</div> <div>Refuser</div>
08/08/2024	Rémunération intervenant/spécialiste	897.00	<div>Reporter</div> <div>Refuser</div>
11/08/2024	Achat de matériel de papèterie	44.00	<div>Reporter</div> <div>Refuser</div>
20/08/2024	Taxi	42.00	<div>Reporter</div> <div>Refuser</div>
26/08/2024	Location salle conférence	399.00	<div>Reporter</div> <div>Refuser</div>

Valider la fiche de frais

III. Réalisation technique

A. Architecture de l'application



B. L'accès à la base de données

fichier GsbManager.php

```
<?php
```

```
/**  
*
```

* Classe qui gère l'accès à la BD et les différentes actions CRUD sur la BD

gsb_frais

*

*/

class GsbManager

{

private static \$serveur = 'mysql:host=localhost';

private static \$bdd = 'dbname=gsb_frais';

private static \$user = 'root';

private static \$mdp = '';

private static \$monPdo;

private static \$monPdoGsb = null;

/**

* Constructeur privé, crée l'instance de PDO qui sera sollicitée

* pour toutes les méthodes de la classe

*/

private function __construct()

{

GsbManager::\$monPdo = new PDO(

GsbManager::\$serveur . ';' . GsbManager::\$bdd,

GsbManager::\$user,

GsbManager::\$mdp

);

GsbManager::\$monPdo->query('SET CHARACTER SET utf8');

}

/**

* Méthode destructeur appelée dès qu'il n'y a plus de référence sur un

* objet donné, ou dans n'importe quel ordre pendant la séquence d'arrêt.

*/

public function __destruct()

{

GsbManager::\$monPdo = null;

}

/**

* Fonction statique qui crée l'unique instance de la classe

* Appel : \$instancePdoGsb = PdoGsb::getPdoGsb();

*

* @return l'unique objet de la classe PdoGsb

*/

public static function getPdoGsb()

{

if (GsbManager::\$monPdoGsb == null) {

GsbManager::\$monPdoGsb = new GsbManager();

}

return GsbManager::\$monPdoGsb;

}

```

/**
 * Retourne les informations d'un utilisateur
 *
 * @param String $login Login de l'utilisateur
 * @param String $mdp Mot de passe de l'utilisateur
 *
 * @return l'id, le nom et le prénom sous la forme d'un tableau associatif
 */

public function getInfosUser($login, $mdp, $metier)
{
    if ($metier == 'visiteur')
    {
        $requetePrepare = GsbManager::$monPdo->prepare(
            'SELECT visiteur.id AS id, visiteur.nom AS nom, '
            . 'visiteur.prenom AS prenom '
            . 'FROM visiteur '
            . 'WHERE visiteur.login = :unLogin AND visiteur.mdp = :unMdp'
        );
    } elseif ($metier == 'comptable')
    {
        $requetePrepare = GsbManager::$monPdo->prepare(
            'SELECT comptable.id AS id, comptable.nom AS nom, '
            . 'comptable.prenom AS prenom '
            . 'FROM comptable '
            . 'WHERE comptable.login = :unLogin AND comptable.mdp = :unMdp'
        );
    }
    $requetePrepare->bindParam(':unLogin', $login, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unMdp', $mdp, PDO::PARAM_STR);
    $requetePrepare->execute();
    return $requetePrepare->fetch();
}

/**
 * Retourne sous forme d'un tableau associatif toutes les lignes de frais
 * hors forfait concernées par les deux arguments.
 * La boucle foreach ne peut être utilisée ici car on procède
 * à une modification de la structure itérée - transformation du champ date-
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois Mois sous la forme aaaamm
 *
 * @return tous les champs des lignes de frais hors forfait sous la forme
 * d'un tableau associatif
 */
public function getLesFraisHorsForfait($idVisiteur, $mois)

```



```

{
    $requetePrepare = GsbManager::$monPdo->prepare(
        'SELECT * FROM lignefraishorsforfait '
        . 'WHERE lignefraishorsforfait.idvisiteur = :unIdVisiteur '
        . 'AND lignefraishorsforfait.mois = :unMois'
    );
    $requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
    $requetePrepare->execute();
    $lesLignes = $requetePrepare->fetchAll();
    for ($i = 0; $i < count($lesLignes); $i++) {
        $date = $lesLignes[$i]['date'];
        $lesLignes[$i]['date'] = dateAnglaisVersFrancais($date);
    }
    return $lesLignes;
}

```

```

/**
 * Retourne le nombre de justificatif d'un visiteur pour un mois donné
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois      Mois sous la forme aaaamm
 *
 * @return le nombre entier de justificatifs
 */

```

```

public function getNbjustificatifs($idVisiteur, $mois)
{
    $requetePrepare = GsbManager::$monPdo->prepare(
        'SELECT fichefrais.nbjustificatifs as nb FROM fichefrais '
        . 'WHERE fichefrais.idvisiteur = :unIdVisiteur '
        . 'AND fichefrais.mois = :unMois'
    );
    $requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
    $requetePrepare->execute();
    $laLigne = $requetePrepare->fetch();
    return $laLigne['nb'];
}

```

```

/**
 * Retourne sous forme d'un tableau associatif toutes les lignes de frais
 * au forfait concernées par les deux arguments
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois      Mois sous la forme aaaamm
 *
 * @return l'id, le libelle et la quantité sous la forme d'un tableau
 * associatif
 */

```

```

public function getLesFraisForfait($idVisiteur, $mois)
{

```

```

$requetePrepare = GsbManager::$monPdo->prepare(
    'SELECT fraisforfait.id as idfrais, '
    . 'fraisforfait.libelle as libelle, '
    . 'lignefraisforfait.quantite as quantite '
    . 'FROM lignefraisforfait '
    . 'INNER JOIN fraisforfait '
    . 'ON fraisforfait.id = lignefraisforfait.idfraisforfait '
    . 'WHERE lignefraisforfait.idvisiteur = :unIdVisiteur '
    . 'AND lignefraisforfait.mois = :unMois '
    . 'ORDER BY lignefraisforfait.idfraisforfait'
);
$requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
$requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
$requetePrepare->execute();
return $requetePrepare->fetchAll();
}

```

```

/**
 * Retourne tous les id de la table FraisForfait
 *
 * @return un tableau associatif
 */

```

```

public function getLesIdFrais()
{
    $requetePrepare = GsbManager::$monPdo->prepare(
        'SELECT fraisforfait.id as idfrais '
        . 'FROM fraisforfait ORDER BY fraisforfait.id'
    );
    $requetePrepare->execute();
    return $requetePrepare->fetchAll();
}

```

```

/**
 * Met à jour la table ligneFraisForfait
 * Met à jour la table ligneFraisForfait pour un visiteur et
 * un mois donné en enregistrant les nouveaux montants
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois      Mois sous la forme aaaamm
 * @param Array  $lesFrais  tableau associatif de clé idFrais et
 *                          de valeur la quantité pour ce frais
 *
 * @return null
 */

```

```

public function majFraisForfait($idVisiteur, $mois, $lesFrais)
{
    $lesCles = array_keys($lesFrais);
    foreach ($lesCles as $unIdFrais) {
        $qte = $lesFrais[$unIdFrais];
        $requetePrepare = GsbManager::$monPdo->prepare(
            'UPDATE lignefraisforfait '

```

```

        . 'SET lignefraisforfait.quantite = :uneQte '
        . 'WHERE lignefraisforfait.idvisiteur = :unIdVisiteur '
        . 'AND lignefraisforfait.mois = :unMois '
        . 'AND lignefraisforfait.idfraisforfait = :idFrais'
    );
    $requetePrepare->bindParam(':uneQte', $qte, PDO::PARAM_INT);
    $requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
    $requetePrepare->bindParam(':idFrais', $unIdFrais, PDO::PARAM_STR);
    $requetePrepare->execute();
}
}

/**
 * Met à jour le nombre de justificatifs de la table ficheFrais
 * pour le mois et le visiteur concerné
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois Mois sous la forme aaaamm
 * @param Integer $nbJustificatifs Nombre de justificatifs
 *
 * @return null
 */
public function majNbJustificatifs($idVisiteur, $mois, $nbJustificatifs)
{
    $requetePrepare = GsbManager::$monPdo->prepare(
        'UPDATE fichefrais '
        . 'SET nbjustificatifs = :unNbJustificatifs '
        . 'WHERE fichefrais.idvisiteur = :unIdVisiteur '
        . 'AND fichefrais.mois = :unMois'
    );
    $requetePrepare->bindParam(
        ':unNbJustificatifs',
        $nbJustificatifs,
        PDO::PARAM_INT
    );
    $requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
    $requetePrepare->execute();
}

/**
 * Teste si un visiteur possède une fiche de frais pour le mois passé en argument
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois Mois sous la forme aaaamm
 *
 * @return vrai ou faux
 */
public function estPremierFraisMois($idVisiteur, $mois)
{

```

```

$boolReturn = false;
$requetePrepare = GsbManager::$monPdo->prepare(
    'SELECT fichefrais.mois FROM fichefrais '
    . 'WHERE fichefrais.mois = :unMois '
    . 'AND fichefrais.idvisiteur = :unIdVisiteur'
);
$requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
$requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
$requetePrepare->execute();
if (!$requetePrepare->fetch()) {
    $boolReturn = true;
}
return $boolReturn;
}

```

```

/**
 * Retourne le dernier mois en cours d'un visiteur
 *
 * @param String $idVisiteur ID du visiteur
 *
 * @return le mois sous la forme aaaamm
 */

```

```

public function dernierMoisSaisi($idVisiteur)
{
    $requetePrepare = GsbManager::$monPdo->prepare(
        'SELECT MAX(mois) as dernierMois '
        . 'FROM fichefrais '
        . 'WHERE fichefrais.idvisiteur = :unIdVisiteur'
    );
    $requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
    $requetePrepare->execute();
    $laLigne = $requetePrepare->fetch();
    $dernierMois = $laLigne['dernierMois'];
    return $dernierMois;
}

```

```

/**
 * Crée une nouvelle fiche de frais et les lignes de frais au forfait
 * pour un visiteur et un mois donnés
 *
 * Récupère le dernier mois en cours de traitement, met à 'CL' son champs
 * idEtat, crée une nouvelle fiche de frais avec un idEtat à 'CR' et crée
 * les lignes de frais forfait de quantités nulles
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois      Mois sous la forme aaaamm
 *
 * @return null
 */

```

```

public function creeNouvellesLignesFrais($idVisiteur, $mois)
{

```

```

$dernierMois = $this->dernierMoisSaisi($idVisiteur);
$laDerniereFiche = $this->getLesInfosFicheFrais($idVisiteur, $dernierMois);
if ($laDerniereFiche != null && $laDerniereFiche['idEtat'] == 'CR') {
    $this->majEtatFicheFrais($idVisiteur, $dernierMois, 'CL');
}
$requetePrepare = GsbManager::$monPdo->prepare(
    'INSERT INTO fichefrais (idvisiteur,mois,nbjustificatifs,'
    . 'montantvalide,datemodif,idetat) '
    . "VALUES (:unIdVisiteur,:unMois,0,0,now(),'CR')"
);
$requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
$requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
$requetePrepare->execute();
$lesIdFrais = $this->getLesIdFrais();
foreach ($lesIdFrais as $unIdFrais) {
    $requetePrepare = GsbManager::$monPdo->prepare(
        'INSERT INTO lignefraisforfait (idvisiteur,mois,'
        . 'idfraisforfait,quantite) '
        . "VALUES(:unIdVisiteur, :unMois, :idFrais, 0)"
    );
    $requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
    $requetePrepare->bindParam(
        ':idFrais',
        $unIdFrais['idfrais'],
        PDO::PARAM_STR
    );
    $requetePrepare->execute();
}
}

/**
 * Crée un nouveau frais hors forfait pour un visiteur un mois donné
 * à partir des informations fournies en paramètre
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois      Mois sous la forme aaaamm
 * @param String $libelle   Libellé du frais
 * @param String $date      Date du frais au format français jj//mm/aaaa
 * @param Float  $montant   Montant du frais
 *
 * @return null
 */
public function creeNouveauFraisHorsForfait(
    $idVisiteur,
    $mois,
    $libelle,
    $date,
    $montant
) {
    $dateFr = dateFrancaisVersAnglais($date);

```

```

$requetePrepare = GsbManager::$monPdo->prepare(
    'INSERT INTO lignefraishorsforfait '
    . 'VALUES (null, :unIdVisiteur,:unMois, :unLibelle, :uneDateFr,'
    . ':unMontant) '
);
$requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
$requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
$requetePrepare->bindParam(':unLibelle', $libelle, PDO::PARAM_STR);
$requetePrepare->bindParam(':uneDateFr', $dateFr, PDO::PARAM_STR);
$requetePrepare->bindParam(':unMontant', $montant, PDO::PARAM_INT);
$requetePrepare->execute();
}

```

```

/**
 * Supprime le frais hors forfait dont l'id est passé en argument
 *
 * @param String $idFrais ID du frais
 *
 * @return null
 */

```

```

public function supprimerFraisHorsForfait($idFrais)
{
    $requetePrepare = GsbManager::$monPdo->prepare(
        'DELETE FROM lignefraishorsforfait '
        . 'WHERE lignefraishorsforfait.id = :unIdFrais'
    );
    $requetePrepare->bindParam(':unIdFrais', $idFrais, PDO::PARAM_STR);
    $requetePrepare->execute();
}

```

```

/**
 * Retourne les mois pour lequel un visiteur a une fiche de frais
 *
 * @param String $idVisiteur ID du visiteur
 *
 * @return un tableau associatif de clé un mois -aaaamm- et de valeurs
 *         l'année et le mois correspondant
 */

```

```

public function getLesMoisDisponibles($idVisiteur)
{
    $requete = 'SELECT DISTINCT fichefrais.mois AS mois FROM fichefrais '
    . 'WHERE fichefrais.idvisiteur = :unIdVisiteur ';

    if($_SESSION['metier'] == 'comptable')
    {
        $requete .= "AND fichefrais.idEtat = 'CL' OR fichefrais.idEtat = 'CR'";
    }

    $requete .= ' ORDER BY fichefrais.mois DESC';

    $requetePrepare = GsbManager::$monPdo->prepare($requete);
}

```

```

$requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
$requetePrepare->execute();
$lesMois = array();
while ($laLigne = $requetePrepare->fetch()) {
    $mois = $laLigne['mois'];
    $numAnnee = substr($mois, 0, 4);
    $numMois = substr($mois, 4, 2);
    $lesMois[] = array(
        'mois' => $mois,
        'numAnnee' => $numAnnee,
        'numMois' => $numMois
    );
}
return $lesMois;
}

/**
 * Retourne les informations d'une fiche de frais d'un visiteur pour un
 * mois donné
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois      Mois sous la forme aaaamm
 *
 * @return un tableau avec des champs de jointure entre une fiche de frais
 *         et la ligne d'état
 */
public function getLesInfosFicheFrais($idVisiteur, $mois)
{
    $requetePrepare = GsbManager::$monPdo->prepare(
        'SELECT fichefrais.idetat as idEtat, '
        . 'fichefrais.datemodif as dateModif, '
        . 'fichefrais.nbjustificatifs as nbJustificatifs, '
        . 'fichefrais.montantvalide as montantValide, '
        . 'etat.libelle as libEtat '
        . 'FROM fichefrais '
        . 'INNER JOIN etat ON fichefrais.idetat = etat.id '
        . 'WHERE fichefrais.idvisiteur = :unIdVisiteur '
        . 'AND fichefrais.mois = :unMois'
    );
    $requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
    $requetePrepare->execute();
    $laLigne = $requetePrepare->fetch();
    return $laLigne;
}

/**
 * Modifie l'état et la date de modification d'une fiche de frais.
 * Modifie le champ idEtat et met la date de modif à aujourd'hui.
 *
 * @param String $idVisiteur ID du visiteur

```

```

* @param String $mois      Mois sous la forme aaaamm
* @param String $etat      Nouvel état de la fiche de frais
*
* @return null
*/
public function majEtatFicheFrais($idVisiteur, $mois, $etat)
{
    $requetePrepare = GsbManager::$monPdo->prepare(
        'UPDATE ficheFrais '
        . 'SET idetat = :unEtat, datemodif = now() '
        . 'WHERE fichefrais.idvisiteur = :unIdVisiteur '
        . 'AND fichefrais.mois = :unMois'
    );
    $requetePrepare->bindParam(':unEtat', $etat, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unIdVisiteur', $idVisiteur, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unMois', $mois, PDO::PARAM_STR);
    $requetePrepare->execute();
}

/**
* Modifie l'état et la date de modification d'une fiche de frais.
* Modifie le champ idEtat et met la date de modif à aujourd'hui.
*
* @param String $idVisiteur ID du visiteur
* @param String $mois      Mois sous la forme aaaamm
* @param String $etat      Nouvel état de la fiche de frais
*
* @return null
*/

public function getLesFichesFrais()
{
    $query = 'SELECT * FROM fichefrais';

    if ($_SESSION['metier'] == 'comptable')
    {
        $query .= " WHERE idEtat = 'CL'";
    }

    $requetePrepare = GsbManager::$monPdo->prepare($query);

    $requetePrepare->execute();
    return $requetePrepare->fetchAll();
}

/**
* Retourne tous les visiteurs de la base de données
*
*
*/

```



```

* Modifie les éléments forfaitisés mis à jour par le comptable :
*/
public function modifierElementForfait($idVisiteur, $mois, $elements)
{
    foreach($elements as $key => $element)
    {
        $query = 'UPDATE lignefraisforfait
        SET quantite = :quantite
        WHERE idVisiteur = :idVisiteur
        AND mois = :mois
        AND idFraisForfait = :idFraisForfait';

        $requetePrepare = GsbManager::$monPdo->prepare($query);
        $requetePrepare->execute(['idVisiteur' => $idVisiteur,
            'mois' => $mois,
            'idFraisForfait' => $key,
            'quantite' => $element]);
    }
}
}

```

C. Architecture d'une page

entête :


[Accueil](#)
[✎ Valider une fiche de frais](#)
[🔌 Déconnexion](#)

Page gestion frais :

Gestion des frais - Comptable : Jean Dupont

Navigation

 Valider une fiche de frais

Sélectionner un visiteur :

Villechalane Louis

Valider

Mes fiches de frais

Villechalane Louis

Sélectionner un mois :

Mois :

09/2024

▼

Valider

Annuler

Fiche de frais du mois 08-2024 :

Etat : Saisie clôturée depuis le 07/08/2024

Montant validé : 0.00

Eléments forfaitisés			
Forfait Etape	Frais Kilométrique	Nuitée Hôtel	Repas Restaurant
8	730	13	7

Confirmer les modifications

Descriptif des éléments hors forfait - 10 justificatifs reçus			
Date	Libellé	Montant	
12/08/2024	Voyage SNCF	133.00	<div>Reporter</div> <div>Refuser</div>
22/08/2024	Repas avec praticien	32.00	<div>Reporter</div> <div>Refuser</div>
08/08/2024	Rémunération intervenant/spécialiste	897.00	<div>Reporter</div> <div>Refuser</div>
11/08/2024	Achat de matériel de papèterie	44.00	<div>Reporter</div> <div>Refuser</div>
20/08/2024	Taxi	42.00	<div>Reporter</div> <div>Refuser</div>
26/08/2024	Location salle conférence	399.00	<div>Reporter</div> <div>Refuser</div>

Valider la fiche de frais