

DESCRIPTION DES MISSIONS REALISEES

Application de gestion des frais

BTS SIO option SLAM

Bassette Chancereul Gwendoline

2018-2020

Sommaire

I.	Description du besoin.....	3
1.	Définition de l'objet.....	3
2.	Forme de l'objet.....	3
3.	Accessibilité/Sécurité.....	3
II.	Contraintes.....	3
1.	Architecture.....	3
2.	Ergonomie.....	3
III.	Codage.....	3
1.	Environnement.....	4
2.	Modules.....	4
IV.	Description du domaine de gestion.....	4
1.	La gestion des frais de déplacement.....	4
2.	Organisation des remboursements.....	4
3.	Schéma de l'application.....	5
4.	Diagramme de cas d'utilisation.....	5
5.	Schéma physique de la base de données à la fin du projet.....	6
V.	Correction de bug.....	7
VI.	Maintenance évolutive.....	13
1.	Demande d'évolution.....	13
2.	Description des modifications à réaliser.....	13
3.	Diagramme de cas d'utilisation.....	15
4.	Description textuelle de cas d'utilisation.....	16
5.	Schéma de l'application.....	17
6.	Description des modifications réalisées.....	20
A.	Ajout d'une liste déroulante pour sélectionner la fiche de frais.....	20
B.	Affichage de la fiche de frais.....	24
C.	Modification des frais forfaitisés.....	29
D.	Suppression d'un frais hors forfait.....	30
E.	Validation de la fiche de frais.....	31

I. Description du besoin

1. Définition de l'objet

Dans le cadre de leur activité, les visiteurs médicaux de GSB sont amenés à engager différents frais qui leur sont remboursés sur justificatifs.

Le suivi des frais est actuellement géré de plusieurs façons selon la région des visiteurs. On souhaite uniformiser cette gestion.

L'application doit permettre d'enregistrer tout frais engagé, aussi bien pour l'activité directe (déplacement, restauration et hébergement) que pour les activités annexes (événementiel, conférences, autres), et de présenter un suivi daté des opérations menées par le service comptable (réception des pièces, validation de la demande de remboursement, mise en paiement, remboursement effectué).

2. Forme de l'objet

L'application Web destinée aux visiteurs, délégués et responsables de secteur sera en ligne, accessible depuis un ordinateur.

La partie utilisée par les services comptables sera aussi sous forme d'une interface Web.

3. Accessibilité/Sécurité

L'environnement doit être accessible aux seuls acteurs de l'entreprise.

Une authentification préalable sera nécessaire pour l'accès au contenu.

II. Contraintes

1. Architecture

L'application respectera l'architecture MVC maison utilisées par les scripts fournis¹ concernant la gestion de l'enregistrement des frais engagés par les visiteurs.

2. Ergonomie

Les pages fournies ont été définies suite à une consultation. Elles constituent une référence ergonomique. Des améliorations ou variations peuvent être proposées.

III. Codage

Le document "ApplisWeb-NormesDevelpt" présente des règles de bonnes pratiques de développement utilisées par le service informatique de GSB pour encadrer le développement d'applications en PHP et en faciliter la maintenance. L'application GSB-AppliFrais fournie s'efforcera de les mettre en œuvre.

Les éléments à fournir devront respecter le nommage des fichiers, variables et paramètres, ainsi que les codes couleurs et la disposition des éléments déjà fournis.

1. Environnement

Le langage de script côté serveur doit être le même que celui utilisé dans les pages fournies. L'utilisation de bibliothèques, API ou Framework est à l'appréciation du prestataire.

2. Modules

L'application présente deux modules :

- Enregistrement et suivi par les visiteurs (code fourni)
- Enregistrement des opérations par les comptables

3. Environnement technique

L'application a été développée en php en respectant l'architecture MVC.

Avec l'**architecture MVC** l'application est composée de modèles, de vues et de contrôleurs. Les modèles permettent l'accès à la base de données et les vues sont les fichiers html permettant l'affichage à l'utilisateur. Les contrôleurs s'occupent du routing, du contrôle des données fournies, de créer des objets en utilisant les classes, et de demander au modèle de créer les enregistrements dans la base de données.

IV. Description du domaine de gestion

1. La gestion des frais de déplacement

Grand poste de dépense, la gestion des frais de déplacement des visiteurs demande un suivi très précis. L'enveloppe annuelle pour ce seul poste s'élève à près de 25 millions d'euros. Il n'est donc pas question de le laisser s'envoler, tout en ne limitant pas les visiteurs à des hôtels de second ordre ou des repas chiches (il en va aussi de l'image de marque du laboratoire et de la motivation des équipes).

Les prix d'hébergement ou de nourriture étant variables d'un lieu à l'autre, d'une région à l'autre, une évaluation statistique a permis de dégager **un montant forfaitaire** dans la fourchette haute des dépenses pour chaque type de frais standard : repas midi, relais étape (nuit plus repas), nuitée (hôtel seul), kilométrage (remboursement des frais kilométriques, chaque visiteur dispose d'un badge pour le télépéage pour éviter le remboursement de ces petits montants).

À ce jour, les montants forfaitaires sont les suivants :

- 110 euros pour un relais étape
- 80 euros pour une nuitée
- 25 euros pour un repas au restaurant
- 0.62 euro pour tout kilomètre parcouru

Le remboursement de l'ensemble des frais engagés par les visiteurs s'organise mensuellement et donne lieu à une fiche de frais identifiée par le numéro du visiteur et le mois de l'année.

2. Organisation des remboursements

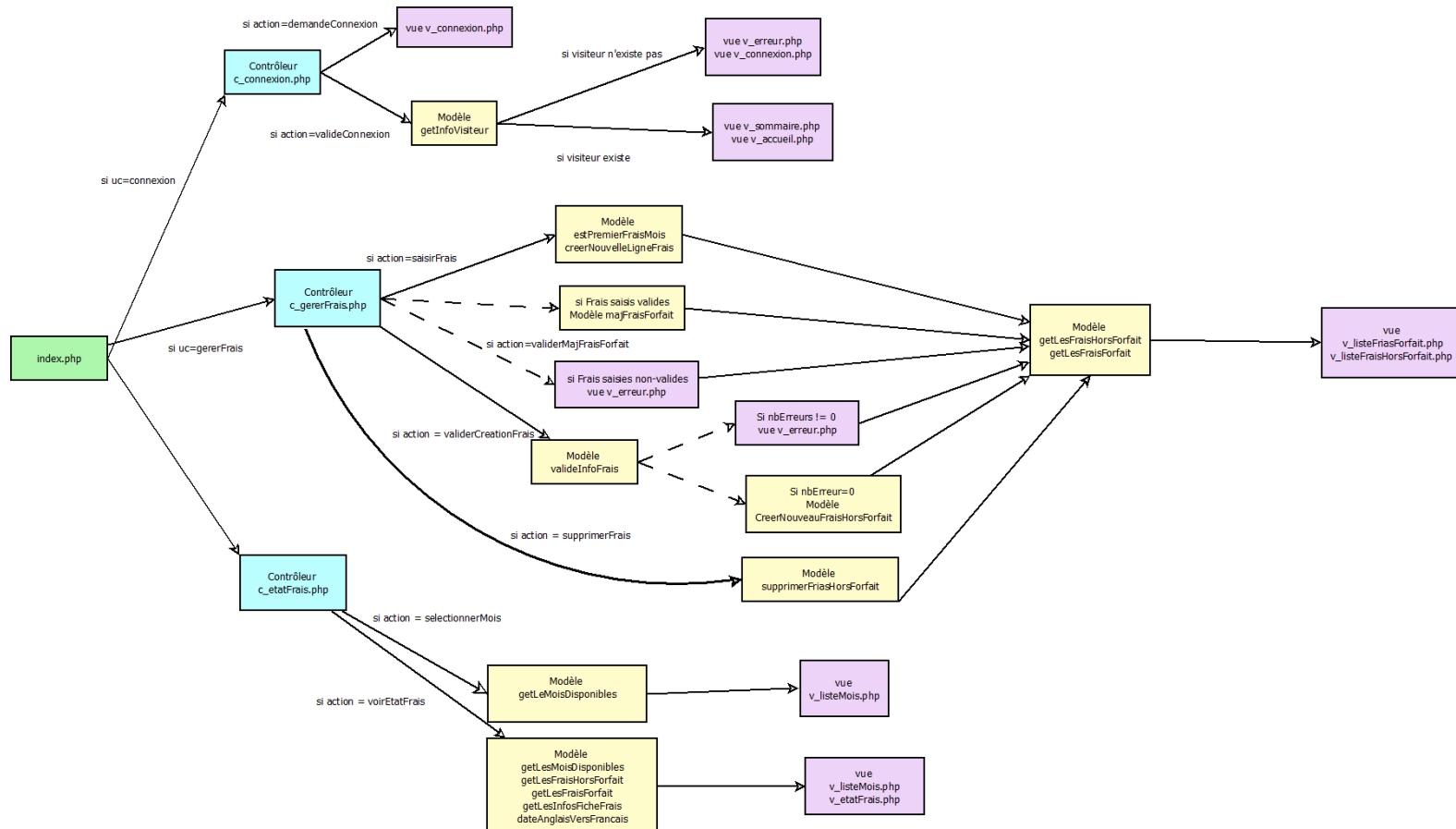
La gestion est la suivante (voir fiche de remboursement fournie) :

- à chaque dépense type (hôtel, repas,...) correspond un **montant forfaitaire** appliqué (on parle de **frais "forfaitisé"**). Le justificatif n'est pas demandé (les rapports de visite serviront de preuve) mais doivent être conservés pendant trois années par les visiteurs. Des contrôles réguliers sont faits par

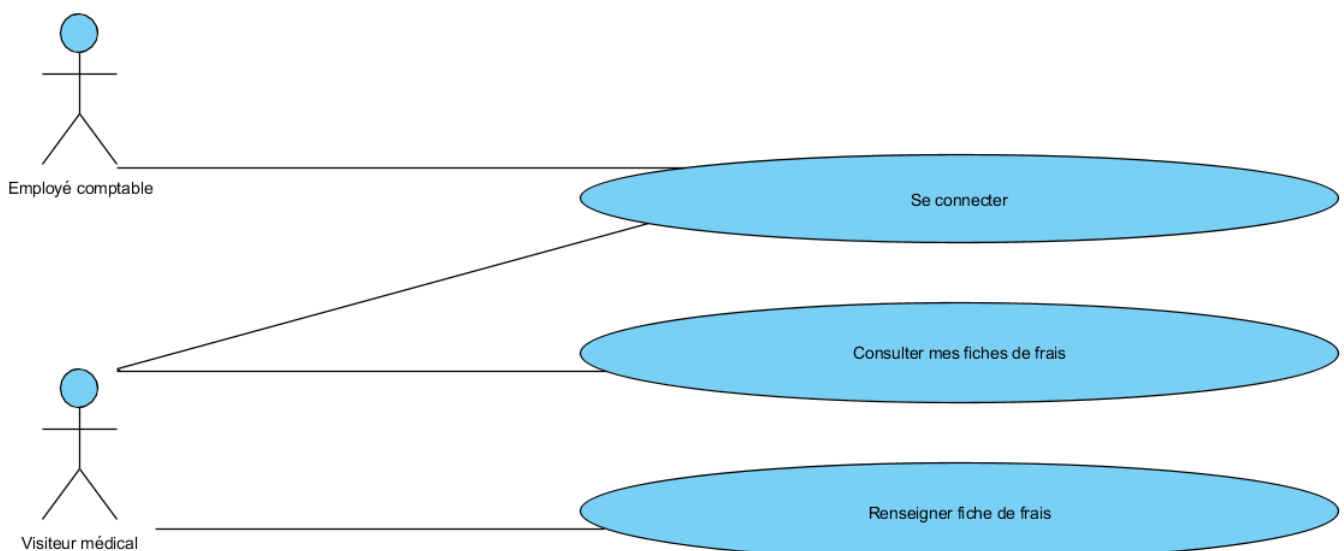
les délégués régionaux qui peuvent donner lieu à des demandes de remboursement de trop perçu par le visiteur

- Pour toute dépense en dehors du forfait (repas en présence d'un spécialiste lors d'une animation, achat de fournitures, réservation de salle pour une conférence, etc.), le visiteur enregistrera la date, le montant et le libellé de la dépense. Il doit fournir au service comptable une facture acquittée. Le système à produire doit lui indiquer le nombre de justificatifs pris en compte dans le remboursement.

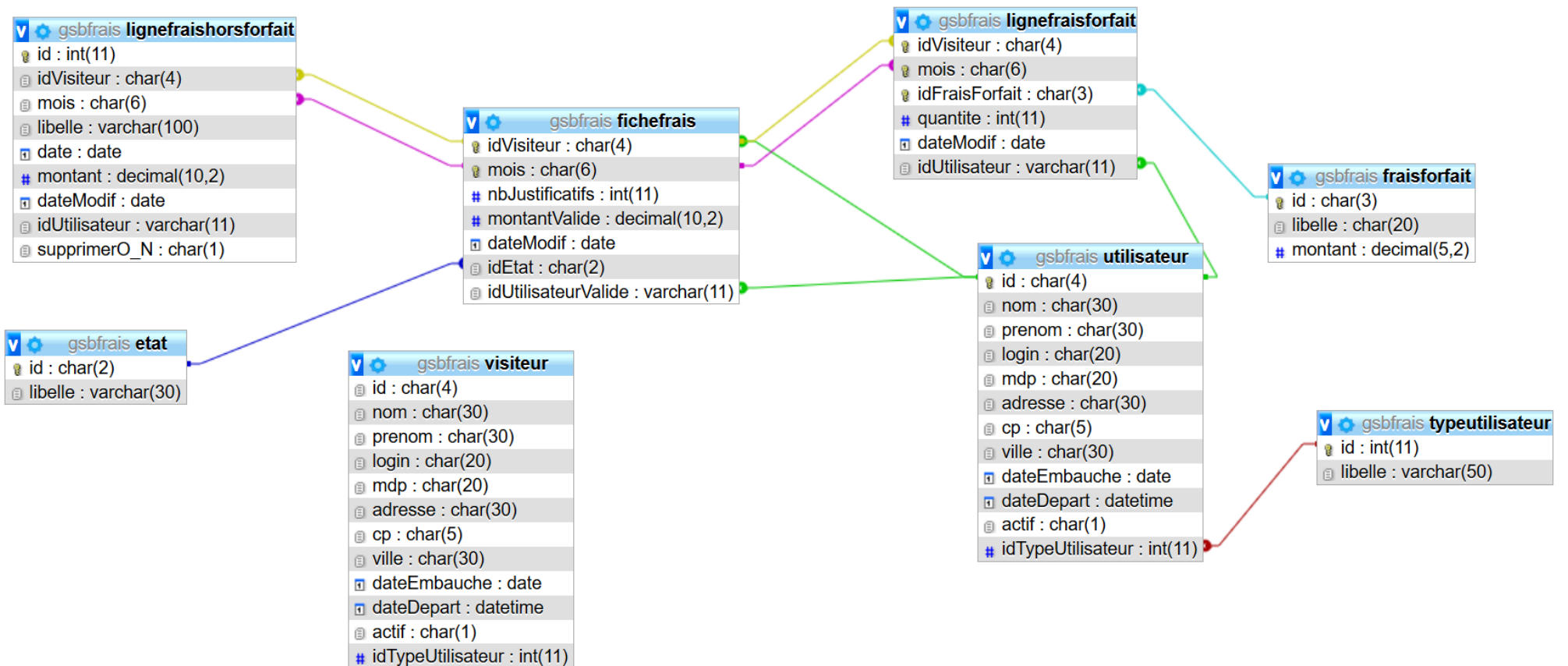
3. Schéma de l'application



4. Diagramme de cas d'utilisation



5. Schéma physique de la base de données à la fin du projet



V. Correction de bug

J'ai reçu le mail suivant d'un utilisateur demandant une correction de bug sur l'application de saisie des frais.

bug application Saisie des frais Boîte de réception x



Agathe Kysai <agathe.kysai@gmail.com>

À moi ▾

Bonjour Gwendoline,

J'ai constaté qu'un visiteur médical avait saisi une date d'engagement de frais antérieure à sa date d'embauche, ceci ne doit pas être possible !

Pourrais-tu, s'il-te-plaît, corriger cette anomalie ?

Cordialement.



Agathe KYSAI


↳ Répondre

➡ Transférer

J'ai d'abord vérifié que le bug existait. Pour cela, j'ai modifié dans la base de données la date d'embauche d'un visiteur médical pour que la date d'embauche soit dans le mois en cours de saisie pour les fiches de frais, j'ai donc saisi comme date d'embauche le 09/11/2019 . Je me suis ensuite connectée avec le compte utilisateur du visiteur médical que j'ai modifié puis j'ai saisi les informations suivantes :

localhost / MySQL / gsbfrais / x bug application Saisie des frais: x +

localhost/GsbMVC/index.php?uc=gererFrais&action=saisirFrais



Suivi du remboursement des frais

[Saisie fiche de frais](#)
[Mes fiches de frais](#)
[Déconnexion](#)

Renseigner ma fiche de frais du mois 11-2019

Eléments forfaitisés

Forfait Etape	<input type="text" value="0"/>
Frais kilométrique	<input type="text" value="0"/>
Nuitée Hôtel	<input type="text" value="0"/>
Repas Restaurant	<input type="text" value="0"/>

Valider Effacer

Descriptif des éléments hors forfait

Date	Libellé	Montant
Nouvel élément hors forfait		
Date (jj/mm/aaaa):	<input type="text" value="02/11/2019"/>	
Libellé	<input type="text" value="Fournitures de bureau"/>	
Montant :	<input type="text" value="45"/>	

Ajouter Effacer

Lorsqu'on valide, aucun message d'erreur ne s'affiche pour informer l'utilisateur que la date saisie est incorrecte. En effet on obtient l'écran suivant :

The screenshot shows a web browser window with the URL `localhost/GsbMVC/index.php?uc=gererFrais&action=saisirFrais`. The page title is "Suivi du remboursement des frais". On the left, there are links: "Saisie fiche de frais", "Mes fiches de frais", and "Déconnexion". The main content area is titled "Renseigner ma fiche de frais du mois 11-2019". It contains two main sections: "Eléments forfaitisés" with four input fields (Forfait Etape, Frais kilométrique, Nuitée Hôtel, Repas Restaurant) and "Descriptif des éléments hors forfait" which includes a table header with "Date", "Libellé", and "Montant". Below the header is a form for "Nouvel élément hors forfait" with input fields for "Date (jj/mm/aaaa)", "Libellé", and "Montant". Buttons for "Valider", "Effacer", "Ajouter", and "Effacer" are present.

Cependant, si on ne renseigne aucun des champs, des messages d'erreurs s'affichent. Il faudrait donc afficher un message d'erreur pour la saisie d'une date d'engagement de frais antérieure à la date d'embauche.

This screenshot shows the same web application but with error messages displayed. A pink banner at the top contains the following error messages:

- Le champ date ne doit pas être vide
- Le champ description ne peut pas être vide
- Le champ montant ne peut pas être vide

 The form structure, including the "Renseigner ma fiche de frais du mois 11-2019" title and the input fields for "Eléments forfaitisés" and "Descriptif des éléments hors forfait", remains the same as in the previous screenshot.

Comme montré ci-dessous, la méthode « valideInfosFrais » qui permet de vérifier les informations saisies, ne contient pas de contrôle qui vérifie que la date saisie n'est pas antérieure à la date d'embauche du visiteur médical.

```
function valideInfosFrais($dateFrais,$libelle,$montant,$dateEmbauche){
    if($dateFrais==""){
        ajouterErreur("Le champ date ne doit pas être vide");
    }
    else{
        if(!estDatevalide($dateFrais)){
            ajouterErreur("Date invalide");
        }
        else{
            if(estDateDepassee($dateFrais)){
                ajouterErreur("date d'enregistrement du frais dépassé, plus de 1 an");
            }
        }
    }
}

if($libelle == ""){
    ajouterErreur("Le champ description ne peut pas être vide");
}
if($montant == ""){
    ajouterErreur("Le champ montant ne peut pas être vide");
}
else
    if( !is_numeric($montant) ){
        ajouterErreur("Le champ montant doit être numérique");
    }
}
```

J'ai donc ajouté des instructions pour effectuer ce contrôle. Pour cela, j'ai tout d'abord réalisé une nouvelle méthode dans le script m_pdogs.php pour obtenir la date d'embauche de l'utilisateur connecté.

```
/** retourne la date d'embauche d'un visiteur médical */
public function getDateEmbaucheVisiteur($idUtilisateur){
    $req = "select dateEmbauche from visiteur where id='$idUtilisateur'";
    $res = PdoGsb::$monPdo->query($req);
    $laLigne = $res->fetch();
    return $laLigne['dateEmbauche'];
}
```

J'appelle ensuite cette méthode dans le contrôleur c_gererFrais.php au moment de valider la création de frais et l'ajoute en paramètre lors de l'appel de la fonction valideInfosFrais.

```
case 'validerCreationFrais':{
    $dateFrais = $_REQUEST['dateFrais'];
    $libelle = $_REQUEST['libelle'];
    $montant = $_REQUEST['montant'];
    $dateEmbauche = $pdo->getDateEmbaucheVisiteur($idVisiteur);
    valideInfosFrais($dateFrais,$libelle,$montant,$dateEmbauche);
    if (nbErreurs() != 0 ){
        include("vues/v_erreurs.php");
    }
    else{
        $pdo->creeNouveauFraisHorsForfait($idVisiteur,$mois,$libelle,$dateFrais,$montant);
    }
    break;
}

case 'supprimerFrais':{
    $idFrais = $_REQUEST['idFrais'];
    $pdo->supprimerFraisHorsForfait($idFrais);
    break;
}
```

J'ai ensuite ajouté une méthode dans le script fct.inc.php, cette méthode renvoie « true » si la date saisie est postérieure à la date d'embauche, false si la date saisie est antérieure à la date d'embauche.

```
/** Vérifie que la date saisie n'est pas antérieure à la date d'embauche
 */
function estDatePosterieurEmbauche($date, $dateEmbauche){
    if($date < $dateEmbauche ){
        return false;
    }
    else{
        return true;
    }
}
```


Enfin, j'ai modifié la méthode valideInfosFrais pour ajouter le message d'erreur si la date saisie est antérieure à la date d'embauche.

```
function valideInfosFrais($dateFrais,$libelle,$montant,$dateEmbauche){
    if($dateFrais==""){
        ajouterErreur("Le champ date ne doit pas être vide");
    }
    else{
        if(!estDatevalide($dateFrais)){
            ajouterErreur("Date invalide");
        }
        else{
            if(estDateDepassee($dateFrais)){
                ajouterErreur("Date d'enregistrement du frais dépassé, plus de 1 an");
            } else{
                if(estDatePosterieurEmbauche($dateFrais, $dateEmbauche)){
                    ajouterErreur("Date d'enregistrement de frais antérieur à la date d'embauche");
                }
            }
        }
    }
}

if($libelle == ""){
    ajouterErreur("Le champ description ne peut pas être vide");
}

if($montant == ""){
    ajouterErreur("Le champ montant ne peut pas être vide");
}
else
    if( !is_numeric($montant) ){
        ajouterErreur("Le champ montant doit être numérique");
    }
}
```

À la suite des modifications réalisées, en saisissant les mêmes informations que précédemment on obtient l'écran suivant :



Suivi du remboursement des frais

a131select dateEmbauche from visiteur where id='a131'

Saisie fiche de frais
Mes fiches de frais
Déconnexion

• date d'enregistrement de frais antérieur à la date d'embauche

Renseigner ma fiche de frais du mois 11-2019

Éléments forfaitisés

Forfait Etape	<input type="text" value="0"/>
Frais kilométrique	<input type="text" value="0"/>
Nuitée Hôtel	<input type="text" value="0"/>
Repas Restaurant	<input type="text" value="0"/>

Valider Effacer

Descriptif des éléments hors forfait

Date	Libellé	Montant						
<div><h4>Nouvel élément hors forfait</h4><table><tr><td>Date (jj/mm/aaaa):</td><td><input type="text"/></td></tr><tr><td>Libellé</td><td><input type="text"/></td></tr><tr><td>Montant :</td><td><input type="text"/></td></tr></table><div>Ajouter Effacer</div></div>			Date (jj/mm/aaaa):	<input type="text"/>	Libellé	<input type="text"/>	Montant :	<input type="text"/>
Date (jj/mm/aaaa):	<input type="text"/>							
Libellé	<input type="text"/>							
Montant :	<input type="text"/>							

Le bug étant corrigé, j'ai envoyé le mail suivant à l'utilisateur :



GWENDOLINE BASSETTE CHANCEREUL

À Agathe ▼

mar. 12 nov. 2019 10:59



Bonjour Madame Kysai,

En effet, un visiteur médical pouvait saisir une date d'engagement de frais antérieure à sa date d'embauche. J'ai donc effectué des modifications pour que ce ne soit plus possible. Désormais, un message d'erreur s'affiche si la date saisie est antérieure à la date d'embauche du visiteur médical.

Voici les informations que j'ai saisies :

La date d'embauche du salarié connecté est le 09/11/2019, la date saisie est donc antérieure à la date d'embauche. Le message d'erreur suivant s'affiche après que l'utilisateur a cliqué sur le bouton "Ajouter" :

Cordialement,
Bassette Chancereul Gwendoline

VI. Maintenance évolutive

1. Demande d'évolution

Entre le 10 et le 20 du mois, le service comptabilité opère une validation des fiches du mois précédent qui sont dans l'état « Saisie clôturée ».

Les comptables contrôlent que les frais forfaitisés sont conformes : nombre de jours enregistrés ne dépassant pas le nombre de jours effectivement travaillés (congrés), distance kilométrique cohérente, éventuellement consultation des fiches de compte-rendu pour s'assurer des déplacements effectifs. En cas d'incohérence ou d'erreur constatée, un contact est pris par téléphone avec le visiteur pour régler le litige. Les valeurs seront corrigées directement par le service comptabilité. On souhaite conserver la date de modification des valeurs s'il y a lieu ainsi que le code de l'employé ayant réalisé la modification.

Pour les **frais hors forfait**, le service comptable s'appuie sur les factures acquittées adressées par les visiteurs au plus tard le 10 du mois suivant la saisie.

Les comptables valident ou non (frais non justifié ou non professionnel par exemple) les éléments de la demande. Une ligne de frais hors forfait **non validée (donc refusée par un comptable)** doit être enregistrée dans un **état « supprimé »**. Le visiteur doit être tenu informé de cette suppression par le comptable qui a traité sa demande. On n'enregistrera pas la raison du refus mais les documents annotés sont conservés par le service comptable. Les agents comptables reportent sur chaque facture reçue le numéro de matricule du visiteur, la date (année/mois) de prise en charge et les classent par ordre chronologique dans une pochette nominative pour chaque visiteur.

Attention : lors de la validation finale de la fiche de frais, le montant total validé (total des frais validés) est enregistré sur la fiche de frais.

2. Description des modifications à réaliser

Modifications à réaliser dans la base de données

- Ajouter deux colonnes dans la table ligne_frais_forfait : dateModif et idUtilisateur
- Ajouter trois colonnes dans la table ligne_frais_forfait : dateModif, idUtilisateur et supprimerO_N
- Ajouter une colonne dans la table fiche_frais : idUtilisateurValide

Modifications à réaliser dans les contrôleurs :

- Créer un nouveau contrôleur c_gérerValidation.php
 - action 1 : Afficher la liste déroulante des fiches de frais clôturées
 - action 2 : Afficher les frais forfaitisés et non-forfaitisés
 - action 3 : Valider la modification des frais forfaitisés
 - action 4 : supprimer les frais hors forfait
 - action 5 : valider la fiches de frais

Modification à réaliser dans le modèle :

- Ajouter une méthode getLesInfosFichesFraisCL qui fait un select dans la base de données et retourne toutes les fiches de frais dans un état CL (clôturées).
- Ajouter une méthode getLesFraisForfaitCL qui fait un select dans la base de données et retourne les frais forfaitisés pour un idVisiteur et un mois passé en paramètre.
- Ajouter une méthode getLesFraisHorsForfaitCL qui fait un select dans la base de données et retourne les frais hors-forfaits pour un idVisiteur et un mois passé en paramètre.
- Ajouter une méthode majFraisForfaitEmpCompt qui met à jour la quantité de chaque frais forfaitisé dans la base de données selon la saisie de l'employé comptable avec la date de modification (date du jour) et l'id de l'employé comptable connecté.

- Ajouter une méthode supprimerFraisHorsForfaitEmpCompt qui fait un update (suppression logique) dans la base de données en mettant supprimerO_N = 'O' ainsi que la date de modification et l'id de l'employé comptable connecté.
- Ajouter une méthode getMontantFraisForfait qui fait un select dans la base de données afin d'obtenir la somme des frais forfaitisés pour un mois et un idVisiteur passé en paramètre.
- Ajouter une méthode getMontantFraisHorsForfait qui fait un select dans la base de données afin d'obtenir la somme des lignes de frais hors-forfait pour un mois et un idVisiteur passé en paramètre.
- Ajouter une méthode validerFicheFrais qui va mettre à jour l'idEtat de la fiche de frais en le mettant dans un état VA (validée), la date de modification sera la date du jour, l'idUtilisateurModif sera l'id de l'utilisateur connecté et le montant validé sera la somme des montant de frais forfaitisés et des frais hors forfait.

Modifications à réaliser dans la vue :

- Modifier la vue sommaire pour ajouter cette fonctionnalité.
- Créer une nouvelle vue v_listeFicheCL.php qui permettra d'afficher la liste déroulante.
- Créer une nouvelle vue v_fraisForfaitCL.php qui permettra l'affichage du formulaire de modification des frais forfaitisés de la fiche de frais sélectionnée précédemment.
- Créer une nouvelle vue v_fraisHorsForfaitCL.php qui permettra l'affichage des frais hors forfait de la fiche de frais sélectionnée précédemment et de les supprimer.
- Créer une nouvelle vue avec le bouton de validation de la fiche de frais.

3. Diagramme de cas d'utilisation

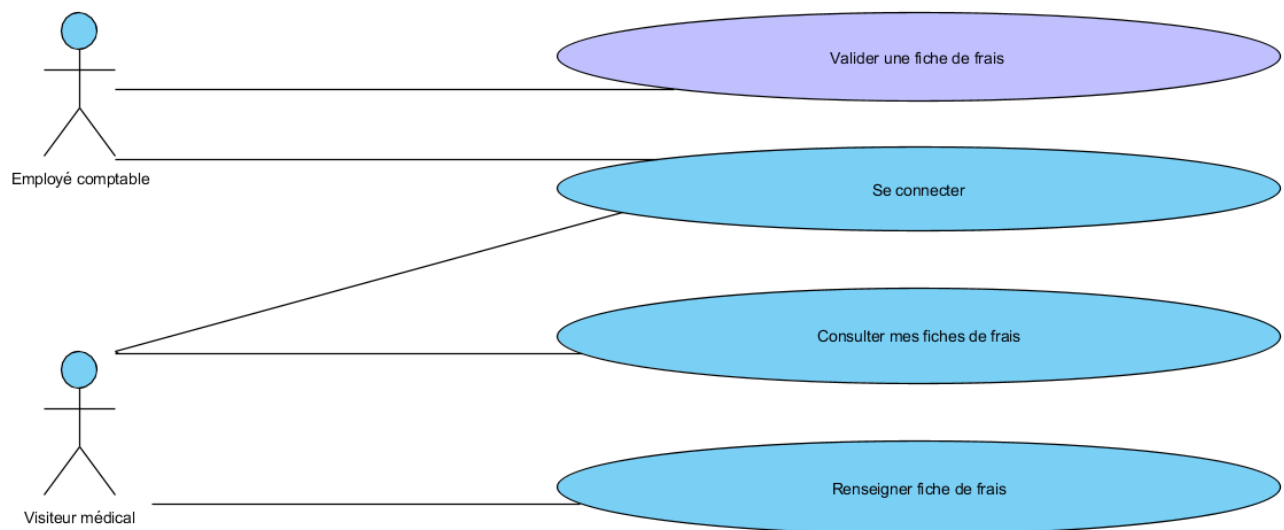


Diagramme de cas d'utilisation de l'application de gestion des frais

4. Description textuelle de cas d'utilisation

PROJET : Application web de gestion des frais	Description cas d'utilisation
Nom cas d'utilisation : Valider une fiche de frais	
Acteur déclencheur : Employé service comptabilité	
Pré conditions : Employé service comptabilité authentifié	
Post conditions : néant	
Scénario nominal : <ol style="list-style-type: none">1. Le système invite à sélectionner une fiche de frais clôturée qui doit être validée.2. L'utilisateur sélectionne une fiche de frais.3. Le système affiche chaque type de frais forfait ainsi que sa quantité.4. L'utilisateur modifie une ou des valeurs des frais forfaitisés et demande la validation.5. Le système enregistre cette ou ces modifications avec la date de modification et le code utilisateur et retourne les valeurs à jour.6. L'utilisateur valide la fiche de frais7. Le système enregistre cette validation et le montant de la fiche de frais validé	
Exceptions : <ol style="list-style-type: none">4.a. Une valeur modifiée n'est pas numérique : le système indique « Valeur numérique attendue ». Retour à 3.	
Alternatif : <ol style="list-style-type: none">8. L'utilisateur sélectionne un frais hors forfait pour suppression.9. Le système enregistre cette suppression après une demande de confirmation.	
Contraintes :	

5. Schéma de l'application

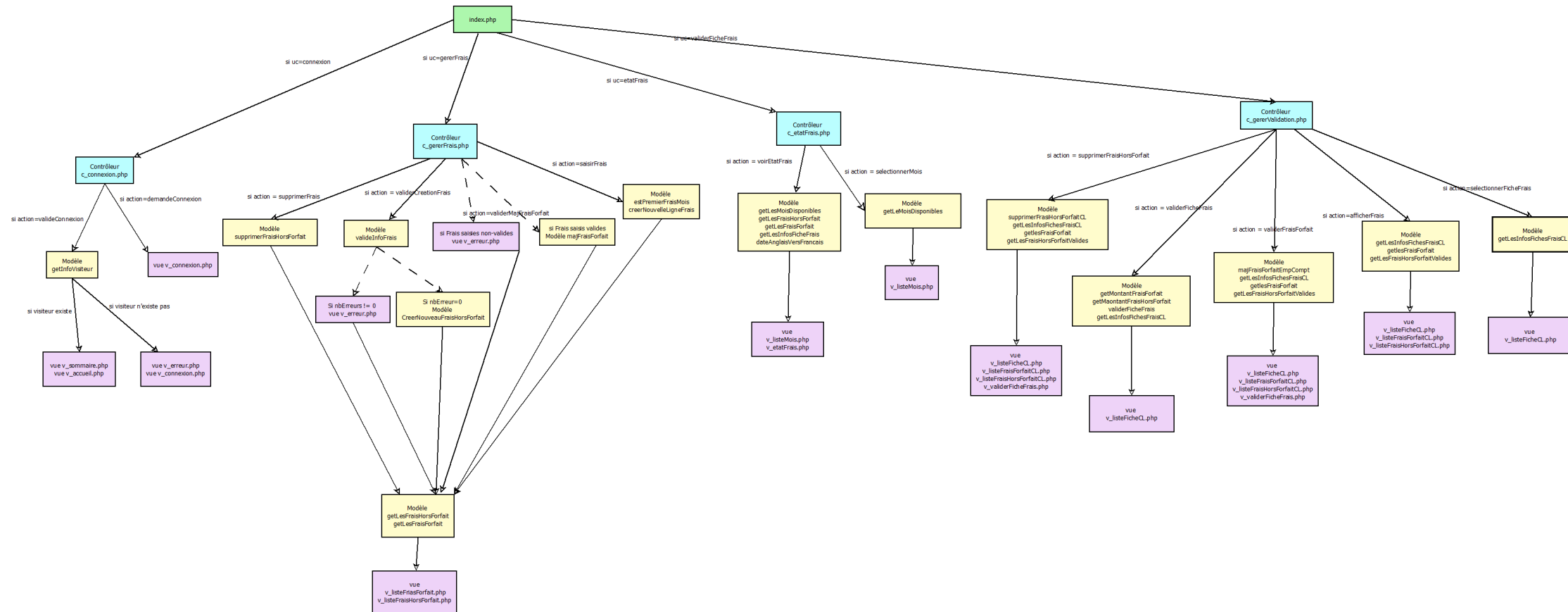


Schéma de l'index.php

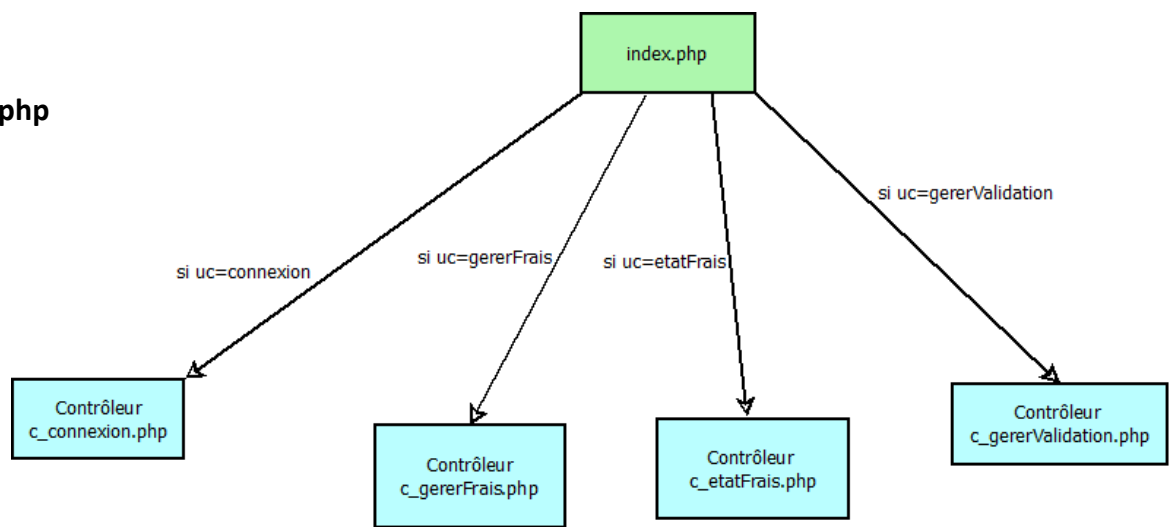


Schéma du contrôleur c_connexion.php

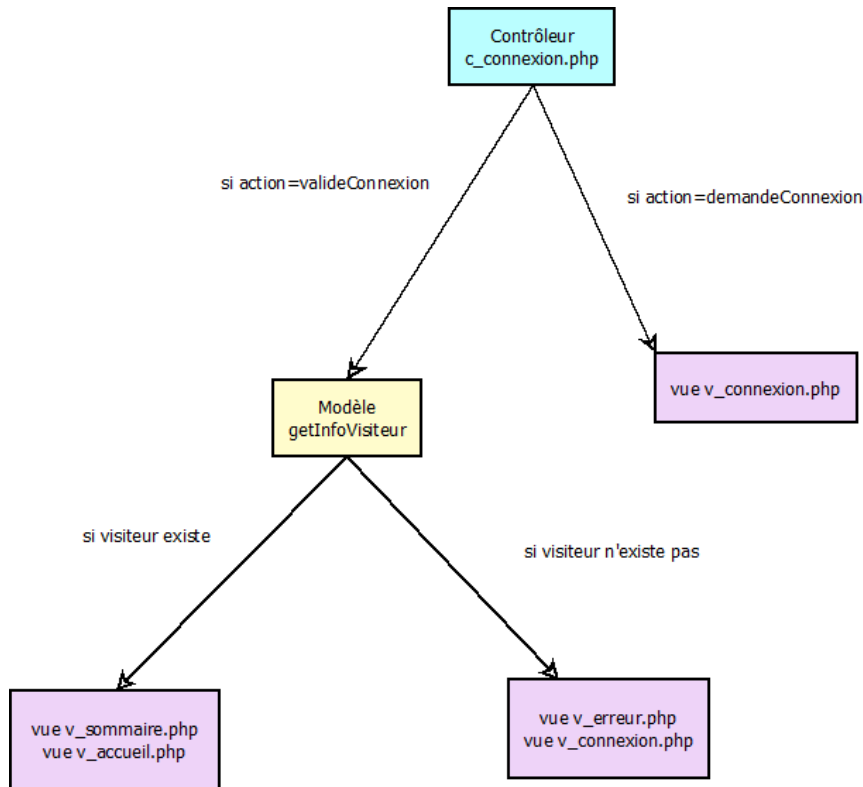


Schéma du contrôleur c_etatFrais.php

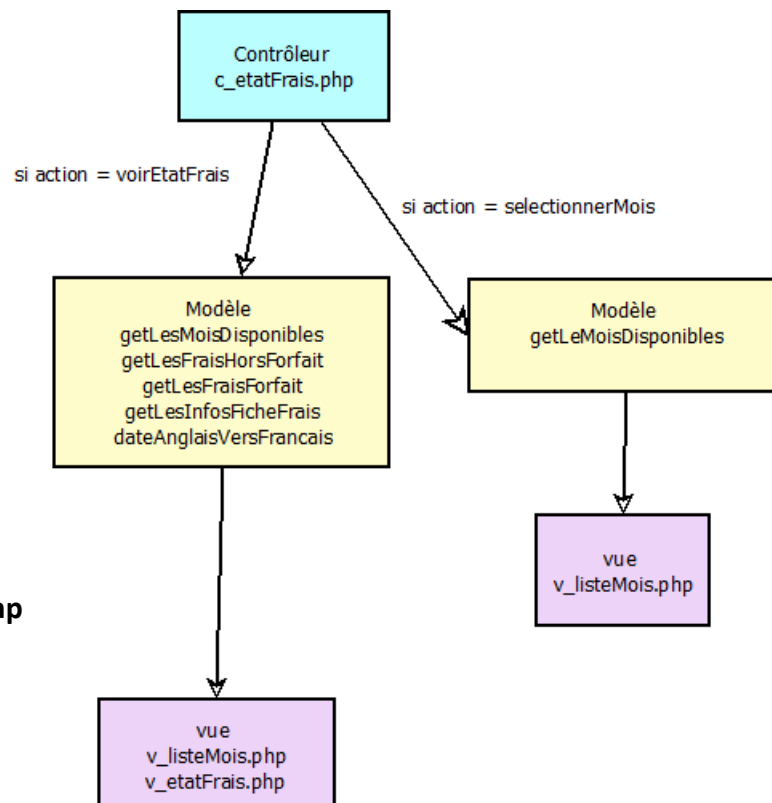


Schéma du contrôleur c_gérerFrais.php

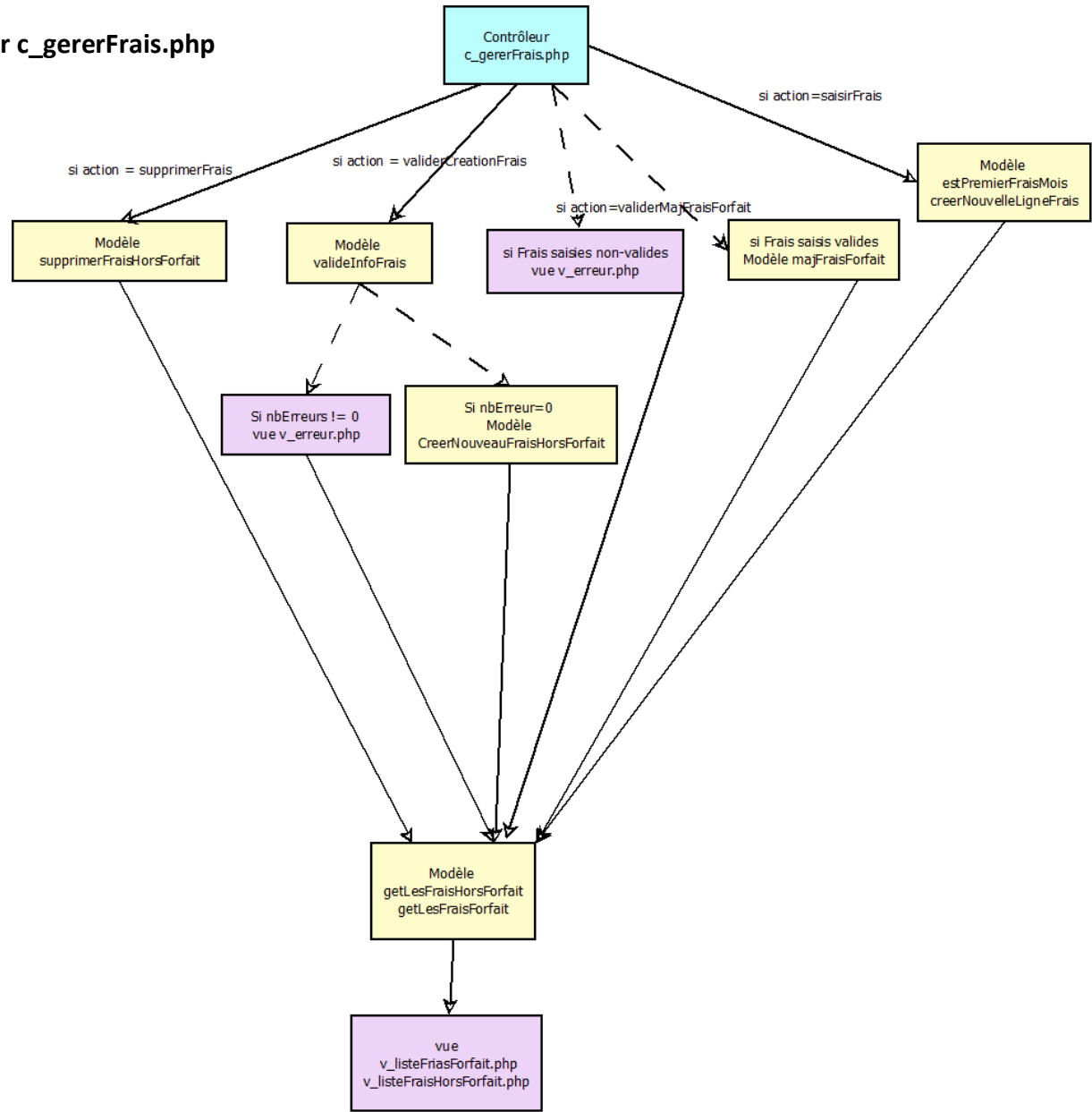
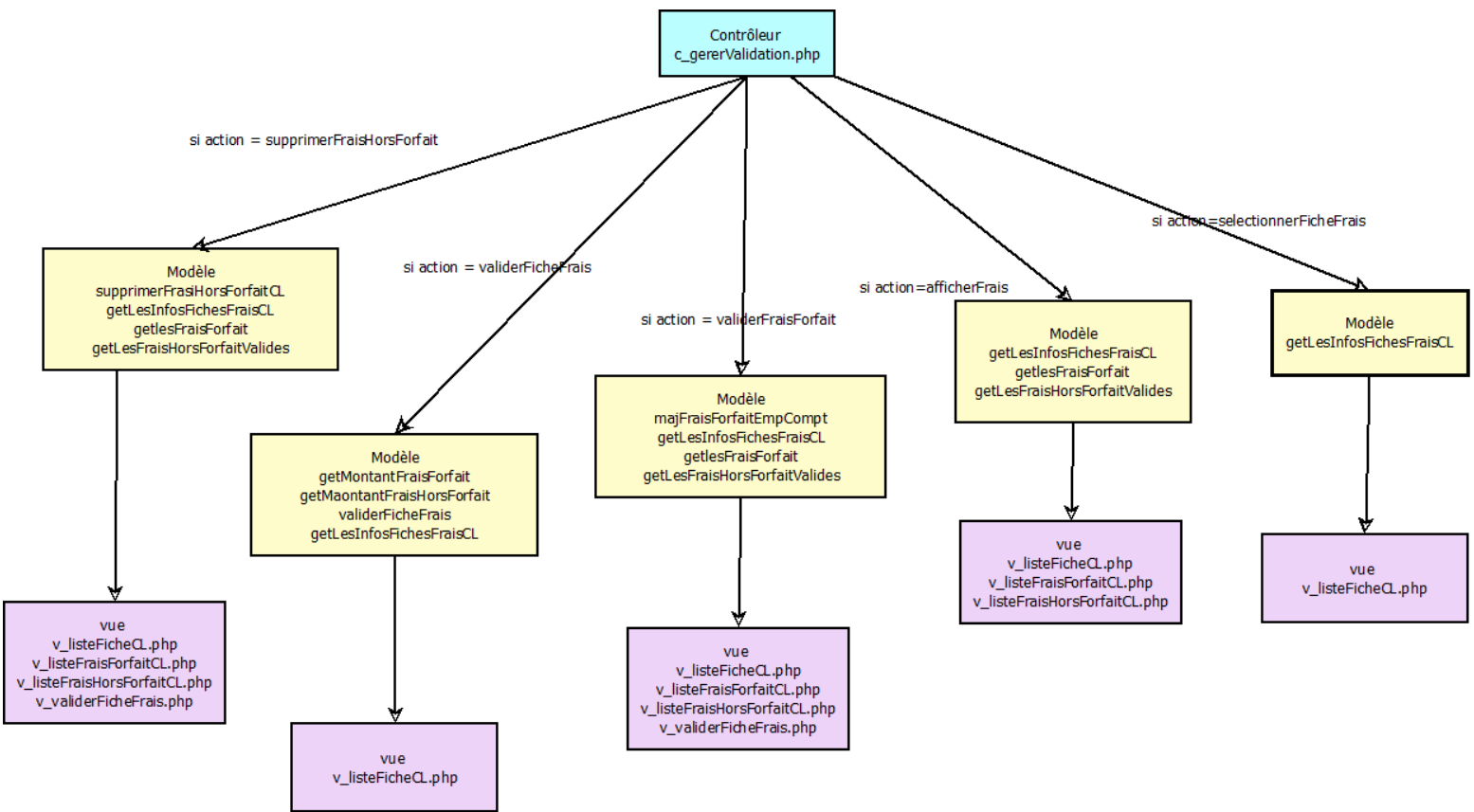


Schéma du nouveau contrôleur c_gérerValidation.php, des vues et des méthodes qu’il appelle :



6. Description des modifications réalisées

Avec l'application existante, seuls les visiteurs médicaux avaient accès à l'application. Le seul type d'utilisateur présent dans la base de données était « visiteur médical ». Pour que les employés comptables puissent aussi se connecter et avoir accès aux fonctionnalités les concernant, j'ai effectué des modifications dans la base de données.

J'ai tout d'abord ajouté une table `typeUtilisateur` avec un `id` et un libellé. J'ai ensuite saisi dans cette table quatre types d'utilisateur : visiteur médical, employé comptable, directeur comptabilité et admin. Une colonne a ensuite été ajoutée dans la table `visiteur` pour gérer le type d'utilisateur, cette colonne a une clé étrangère sur `id` de la table `typeUtilisateur`. J'ai ensuite exécuté la requête SQL suivante :

```
update visiteur set idUtilisateur = 1
```

Avec cette requête l'`idUtilisateur` de toutes les lignes présentes dans la table `visiteur` est 1. Il n'y a pas besoin de faire de `where` car dans la table `visiteur` il n'y a que des visiteurs médicaux.

J'ai ensuite renommé la table `visiteur` en `utilisateur`, puis pour ne pas avoir à modifier toutes les requêtes SQL qui utiliseraient la table `visiteur`, j'ai créé une vue appelée « visiteur » qui fait un `select` de toutes les données de la table `utilisateur` où l'`idTypeUtilisateur` est 1, donc pour les visiteurs médicaux.

Pour pouvoir ajouter la nouvelle fonctionnalité permettant la validation des fiches de frais, j'ai créé un nouveau contrôleur que j'ai ensuite appelé dans le fichier `index.php`. J'ai également créé une nouvelle vue pour le sommaire des employés comptables dans laquelle j'ai ajouté le lien à la fonctionnalité que j'ai développé et un bouton « déconnexion ».

A. Ajout d'une liste déroulante pour sélectionner la fiche de frais

Après que l'employé comptable s'est connecté, il obtient l'écran suivant :



Lorsqu'il clique sur « Valider les fiches de frais » dans le menu, il obtient la page ci-dessous. L'utilisateur est invité à sélectionner une fiche de frais dans la liste déroulante. La liste déroulante affiche le nom et le prénom du visiteur médical ainsi que le mois et l'année auxquels correspond la fiche de frais. Lorsqu'une fiche est sélectionnée dans la liste déroulante, l'employé comptable doit appuyer sur « valider » pour afficher les frais forfaitisés et non-forfaitisés saisies par le visiteur médical.

Pour obtenir cette page, j'ai d'abord réalisé une méthode `getLesInfosFichesFraisCL` dans le modèle `m_pdogsb.php`, qui permet d'obtenir toutes les fiches de frais à valider :

```
public function getLesInfosFichesFraisCL() {
    $req = "select idVisiteur, nom, prenom, mois from fichefrais join utilisateur on idVisiteur = utilisateur.id
    where idEtat='CL'
    order by fichefrais.mois desc ";
    $res = PdoGsb::$monPdo->query($req);
    $lesFichesFraisCL = array();
    $laLigne = $res->fetch();
    $ind = 0;
    while($laLigne != null) {
        $mois = $laLigne['mois'];
        $nomVisiteur = $laLigne['nom'];
        $prenomVisiteur = $laLigne['prenom'];
        $idVisiteur = $laLigne['idVisiteur'];
        $numAnnee = substr($mois, 0, 4);
        $numMois = substr($mois, 4, 2);
        $lesFichesFraisCL[$ind] = array(
            "mois" => "$mois",
            "numAnnee" => "$numAnnee",
            "numMois" => "$numMois",
            "nomVisiteur" => "$nomVisiteur",
            "prenomVisiteur" => "$prenomVisiteur",
            "idVisiteur" => "$idVisiteur"
        );
        $ind = $ind + 1;
        $laLigne = $res->fetch();
    }
    return $lesFichesFraisCL;
}
```

Cette méthode exécute une requête SQL qui permet de récupérer l'idVisiteur, le nom du visiteur, le prénom du visiteur, et le mois correspondant à la fiche de frais dont l'état est clôturé. Les résultats de la requête SQL sont ensuite mis dans un tableau.

Au début du nouveau contrôleur `c_gererValidation.php`, j'ai fait un `include` de la vue sommaire des employés comptables, récupéré l'id de l'utilisateur connecté avec les variables de session et récupéré l'action à réaliser. J'ai ensuite réalisé un `switch` qui comporte plusieurs actions. La première correspond à la sélection d'une fiche de frais dans la liste déroulante. Avec ces instructions on appelle la méthode `getLesInfosFichesFraisCL` pour obtenir toutes les fiches de frais à valider. Ensuite on récupère toutes les clés du tableau et on précise que la fiche à sélectionner sera la première. Enfin, on effectue un `include` de la vue `v_listeFicheCL.php`

```
<?php
include("vues/v_sommaire_empCompt.php");
$idUtilisateur = $_SESSION['idUtilisateur'];
$action = $_REQUEST['action'];
switch($action){
    case 'selectionnerFicheFrais':{
        //On récupère toutes les fiches de frais dans un état clôturé
        $lesFichesFraisCL = $pdo->getLesInfosFichesFraisCL();

        $lesCles = array_keys( $lesFichesFraisCL);

        //La fiche à sélectionner sera la première de la liste
        $ficheASelectionner = $lesCles[0];

        include("vues/v_listeFicheCL.php");
        break;
    }
}
```

Contrôleur `c_gererValidation.php`

La vue `v_listeFicheCL.php` contient le code html permettant l’affichage de la liste déroulante. Pour chaque élément de `$lesFichesFraisCL` on récupère le mois, le numéro de l’année, le numéro du mois, le nom et le prénom du visiteur et son id. Si une fiche a déjà été sélectionnée on la sélectionne dans la liste déroulante, sinon on se positionne sur la première de la liste. Pour chaque élément de la liste déroulante sa valeur est l’id du visiteur et le mois séparé par un slash. La valeur qu’on affiche dans la liste déroulante, donc ce qui est visible pour l’utilisateur, est le nom et le prénom du visiteur ainsi que le mois et l’année de la fiche de frais. Lorsque l’utilisateur clique sur le bouton « valider », on exécute les instruction du case « afficherFrais » du switch présent dans le contrôleur `c_gererValidation.php`. Voici les instructions qu’elle contient :

```
<form action="index.php?uc=gererValidation&action=afficherFrais" method="post">
  <div class="corpsForm">
    <p>
      <label for="lstFicheFrais" accesskey="n">Fiche de frais : </label>
      <select id="lstFicheFrais" name="lstFicheFrais">
        <?php
          var_dump ($lesFichesFraisCL);
          foreach ($lesFichesFraisCL as $uneFicheCL)
          {
            $mois = $uneFicheCL['mois'];
            $numAnnee = $uneFicheCL['numAnnee'];
            $numMois = $uneFicheCL['numMois'];
            $nomVisiteur = $uneFicheCL['nomVisiteur'];
            $prenomVisiteur = $uneFicheCL['prenomVisiteur'];
            $idVisiteur = $uneFicheCL['idVisiteur'];
            if($laFiche == $idVisiteur."/".$mois){
              <?>
              <option selected value="<?php echo $idVisiteur."/".$mois ?>">
                <?php echo $nomVisiteur." ".$prenomVisiteur." ".$numMois."/".$numAnnee?> </option>
              <?php
            }
            else{ ?>
              <option value="<?php echo $idVisiteur."/".$mois ?>">
                <?php echo $nomVisiteur." ".$prenomVisiteur." ".$numMois."/".$numAnnee?> </option>
              <?php
            }
          }
        </?php>
      </select>
    </p>
  </div>
</form>
```

La liste déroulante s’affiche ainsi :

Fiche de frais :	<div> <div>Bedos Christian 10/2019</div> <div> <div>Bedos Christian 10/2019</div> <div>Cacheux Bernard 10/2019</div> <div>Cadic Eric 10/2019</div> </div> </div>	<div>Valider</div> <div>Effacer</div>
------------------	--	---------------------------------------

Lorsque l’utilisateur clique sur « valider », la fiche de frais qu’il a sélectionnée va s’afficher.

B. Affichage de la fiche de frais

La fiche de frais comporte les frais forfaitisés et les frais non-forfaitisés. Pour les frais forfaitisés qui sont forfait étape, frais kilométrique, nuitée hotel et repas restaurant, le visiteur médical saisit une quantité, puis est remboursé le montant forfaitaire du type de frais multiplié par sa quantité. Les frais hors forfait sont saisis par le visiteur médical, il saisit une date, un libellé de frais et un montant, puis valide.

L'employé comptable doit pouvoir modifier la quantité de chaque type de frais forfaitisé et supprimer des frais hors forfait.

Pour cela, il faut que les frais forfaitisés soit affichés sous la forme de textbox pré-remplies avec les informations saisies par le visiteur médicale pour la fiche de frais sélectionné. Pour obtenir ces informations j'ai utilisé la méthode déjà existante `getLesFraisForfait`.

Cette méthode effectue un select dans la base de donnée pour obtenir l'idFrais, le libelle et la quantité pour un idVisiteur et un mois passé en paramètre. Ces paramètres seront l'utilisateur et le mois sélectionné dans la liste déroulante lors du choix de la fiche de frais. Les lignes sont ensuite retournées par la méthode.

Voici la méthode `getLesFraisForfait` :

```
public function getLesFraisForfait($idVisiteur, $mois){
    $req = "select fraisforfait.id as idfrais, fraisforfait.libelle as libelle,
    lignefraisforfait.quantite as quantite from lignefraisforfait inner join fraisforfait
    on fraisforfait.id = lignefraisforfait.idfraisforfait
    where lignefraisforfait.idvisiteur = '$idVisiteur' and lignefraisforfait.mois='$mois'
    order by lignefraisforfait.idfraisforfait";
    $res = PdoGsb::$monPdo->query($req);
    $lesLignes = $res->fetchAll();
    return $lesLignes;
}
```

Les frais hors forfait doivent être affichés sous la forme d'un tableau comportant la date, le libellé, le montant du frais et un bouton « supprimer ce frais ». Pour obtenir les frais hors forfait correspondant à la fiche de frais sélectionné, j'ai réalisé la méthode `getLesFraisHorsForfaitValide`.

Cette méthode effectue un select dans la base de donnée pour obtenir l'id du frais hors forfait, son montant, son libellé et la date du frais pour un idVisiteur et un mois passé en paramètre et le champ `supprimerO_N` contient 'N' ou est null. Cette dernière clause dans le where permettra d'afficher au comptable uniquement les lignes de frais hors forfait qu'il n'a pas supprimées. La méthode retourne ensuite les lignes qui ont été récupérées avec le select. Voici cette méthode :

```
public function getLesFraisHorsForfaitValides($idVisiteur,$mois){
    $req = "select id ,montant, libelle, date from lignefraishorsforfait
    where lignefraishorsforfait.idvisiteur = '$idVisiteur'
    and lignefraishorsforfait.mois = '$mois' and (supprimerO_N='N' or supprimerO_N IS NULL)";
    $res = PdoGsb::$monPdo->query($req);
    $lesLignes = $res->fetchAll();
    return $lesLignes;
}
```

Pour afficher cela, j'ai ajouté un case dans le switch de mon contrôleur `c_gererValidation.php`. Ici, on récupère la valeur de l'élément sélectionné dans la liste déroulante, puis on appelle la méthode `getLesInfosFicheCL` afin de remplir à nouveau la liste déroulante comportant toutes les fiches de frais clôturées. La valeur de l'élément sélectionné qu'on récupère est sous la forme `idVisiteur/mois`, par exemple `a123/112019`, il faut donc séparer l'id du visiteur médical et le mois. Pour cela j'ai fait un `explode` de la valeur récupérée et mis ces informations dans les variables `idVisiteurSelect` et `moisSelect`. J'ai ensuite fait un `include` de la vue permettant l'affichage de la liste déroulante contenant toutes les fiches de frais. Il fallait

ensuite afficher les frais forfaitisés et hors forfait correspondant à la fiche de frais sélectionnée. J'ai donc appelé les méthode getLesFraisForfait et getLesFraisHorsForfaitValides. Puis avec le mois récupéré plus haut, j'ai séparé le mois et l'année afin de les afficher en haut de la page. J'ai ensuite fait les includes des vues permettant l'affichage des frais forfaitisés, non-forfaitisés et du bouton de validation de la fiche de frais. Voici ces instructions :

```
case 'afficherFrais':{  
    //On récupère la valeur de l'élément sélectionné  
    $laFiche = $_REQUEST['lstFicheFrais'];  
  
    //On récupère les fiches de frais dans un état cloturé  
    $lesFichesFraisCL = $pdo->getLesInfosFichesFraisCL();  
    $ficheASelectionner = $laFiche;  
  
    //On sépare la valeur récupérée  
    $ValSelect = explode("/", $laFiche);  
    $idVisiteurSelect = $ValSelect[0];  
    $moisSelect = $ValSelect[1];  
  
    include("vues/v_listeFicheCL.php");  
    //On récupère les frais forfait et hors forfait du visiteur et du mois sélectionné  
    $lesFraisForfait= $pdo->getLesFraisForfait($idVisiteurSelect,$moisSelect);  
    $lesFraisHorsForfait = $pdo->getLesFraisHorsForfaitValides($idVisiteurSelect,$moisSelect);  
  
    //On sépare le mois et l'année pour l'afficher  
    $numAnnee =substr( $moisSelect,0,4);  
    $numMois =substr( $moisSelect,4,2);  
  
    include("vues/v_listeFraisForfaitCL.php");  
    include("vues/v_listeFraisHorsForfaitCL.php");  
    include("vues/v_validerFicheFrais.php");  
    break;  
}
```

Pour afficher les frais forfaitisés j'ai créé la nouvelle vue v_listeFraisForfaitCL.php. Cette vue permet d'afficher pour chaque type de frais son libellé et la quantité saisie par le visiteur médical. J'ai aussi un champ caché pour avoir le mois et l'id du visiteur lorsqu'on clique sur le bouton valider.

```
<div id="contenu">
  <h2>Fiche de frais du mois <?php echo $numMois."-".$numAnnee?></h2>
  <form method="POST" action="index.php?uc=gererValidation&action=validerFraisForfait">
    <div class="corpsForm">
      <fieldset>
        <legend>Eléments forfaitisés
        </legend>

        <input type='hidden' name='idVisiteurMois' value='<?php echo $idVisiteurSelect."/".$moisSelect ?>'>
        <?php
        foreach ($lesFraisForfait as $unFrais)
        {
          $idFrais = $unFrais['idfrais'];
          $libelle = $unFrais['libelle'];
          $quantite = $unFrais['quantite'];
          ?>
          <p>
            <label for="idfrais"><?php echo $libelle ?></label>
            <input type="text" id="idfrais" name="lesFrais[<?php echo $idFrais?>]" size="10"
              maxlength="5" value="<?php echo $quantite?>" >
          </p>

          <?php
          }
        }
      </fieldset>
    </div>
    <div class="piedForm">
      <p>
        <input id="ok" type="submit" value="Valider" size="20" />
        <input id="annuler" type="reset" value="Effacer" size="20" />
      </p>
    </div>
  </form>
```

Pour afficher les frais hors forfait, j'ai ajouté la vue v_listeFraisHorsforfaitCL.php

```
<table class="listeLegere">
  <caption>Descriptif des éléments hors forfait
</caption>
  <tr>
    <th class="date">Date</th>
    <th class="libelle">Libellé</th>
    <th class="montant">Montant</th>
    <th class="action">&nbsp;</th>
  </tr>

  <?php

  foreach( $lesFraisHorsForfait as $unFraisHorsForfait)
  {
    $libelle = $unFraisHorsForfait['libelle'];
    $date = $unFraisHorsForfait['date'];
    $montant=$unFraisHorsForfait['montant'];
    $id = $unFraisHorsForfait['id'];
  >
    <tr>
      <td> <?php echo $date ?></td>
      <td><?php echo $libelle ?></td>
      <td><?php echo $montant ?></td>
      <td><a href="index.php?uc=gererValidation&action=supprimerFraisHorsForfait&idFrais=<?php echo $id ?>
&idVisiteurSelect=<?php echo $idVisiteurSelect?>&moisSelect=<?php echo $moisSelect?>"
onclick="return confirm('Voulez-vous vraiment supprimer ce frais?');">Supprimer ce frais</a></td>
    </tr>
  <?php
  }
```

On crée un tableau avec quatres colonnes : la date, le libellé, le montant et le bouton « supprimer ». Ensuite avec les lignes retournées par la méthode getLesFraisHorsForfaitValide on affiche les lignes de frais dans le tableau.

Pour afficher le bouton « valider » j'ai ajouté une vue v_validerFicheFrais.php qui ne contient que le bouton « valider » et un champ caché avec le mois et l'id du visiteur médical.

```
<form action="index.php?uc=gererValidation&action=ValiderFicheFrais" method="post">
  <p>
    <input type='hidden' name='idVisiteurMois' value='<?php echo $idVisiteurSelect."/".$moisSelect ?>'>
    <input id="valider" type="submit" value="Valider" size="30" />
  </p>
</form>
```

Voici l'écran qu'obtient l'utilisateur après avoir sélectionné une fiche de frais et cliqué sur le bouton « valider » :

[Valider les fiches de frais](#)
[Déconnexion](#)

Les fiches de frais à valider

Fiche de frais à sélectionner :

Fiche de frais : Bedos Christian 10/2019

1

Valider Effacer

Fiche de frais du mois 10-2019

Éléments forfaitisés

Forfait Etape 7

Frais kilométrique 612

Nuitée Hôtel 3

Repas Restaurant 10

2

Valider Effacer

Descriptif des éléments hors forfait

Date	Libellé	Montant	
2019-10-19	Voyage SNCF	90.00	Supprimer ce frais
2019-10-03	location véhicule	423.00	Supprimer ce frais
2019-10-08	repas avec praticien	35.00	Supprimer ce frais
2019-10-25	location équipement vidéo/sonore	164.00	Supprimer ce frais
2019-10-11	traiteur, alimentation, boisson	231.00	Supprimer ce frais
2019-10-07	location équipement vidéo/sonore	109.00	Supprimer ce frais

3

Valider

4

Dans la liste déroulante (1), la fiche sélectionnée est celle qui a été sélectionnée précédemment. Il est possible de changer de fiche en utilisant cette liste déroulante. En dessous, se trouve les frais forfaitisés (2) et non-forfaitisés (3) saisis par le visiteur médical pour le mois sélectionné. Tout en bas de la page se trouve le bouton pour valider la fiche de frais (4).

C. Modification des frais forfaitisés

L'utilisateur peut modifier les quantités pour chaque frais forfaitisé. Pour cela l'utilisateur doit simplement modifier la valeurs dans la zone de saisie correspondant au type de frais forfaitisé puis cliquer sur « valider ».

[Valider les fiches de frais](#)
[Déconnexion](#)

Les fiches de frais à valider

Fiche de frais à sélectionner :

Fiche de frais : Bedos Christian 10/2019

Valider Effacer

Fiche de frais du mois 10-2019

Éléments forfaitisés

Forfait Etape	7
Frais kilométrique	612
Nuitée Hôtel	3
Repas Restaurant	10

Valider Effacer

J'ai donc ajouté un case dans le contrôleur c_gererValidation.php dont voici les instructions :

```
case 'validerFraisForfait':{
    $lesFrais = $_REQUEST['lesFrais'];
    $idSelect = explode("/",$_REQUEST['idVisiteurMois']);
    $idVisiteurSelect = $idSelect[0];
    $moisSelect = $idSelect[1];

    if(lesQteFraisValides($lesFrais)){
        $pdo->majFraisForfaitEmpCompt($idVisiteurSelect,$moisSelect,$lesFrais,$idUtilisateur);
    }
    else{
        ajouterErreur("Les valeurs des frais doivent être numériques");
        include("vues/v_erreurs.php");
    }

    //On récupère les frais forfait et hors forfait du visiteur et du mois sélectionné
    $lesFraisForfait= $pdo->getLesFraisForfait($idVisiteurSelect,$moisSelect);
    $lesFraisHorsForfait = $pdo->getLesFraisHorsForfaitValides($idVisiteurSelect,$moisSelect);
    //On récupère les fiches de frais dans un état cloturé
    $lesFichesFraisCL = $pdo->getLesInfosFichesFraisCL();

    include("vues/v_listeFicheCL.php");
    include("vues/v_listeFraisForfaitCL.php");
    include("vues/v_listeFraisHorsForfaitCL.php");
    include("vues/v_validerFicheFrais.php");

    break;
}
```

On récupère les quantités de chaque frais forfaitisé avec le \$_REQUEST puis l'idVisiteur et le mois. Si les frais saisis sont valides, c'est-à-dire que les chiffres sont cohérents avec le nombre de jours du mois, que les valeurs saisies sont numériques... alors on appelle la méthode de mise à jour des frais forfaitisés dans la base de données. Sinon un message d'erreur est affiché à l'utilisateur. Ensuite on appelle les méthodes permettant d'afficher la liste déroulante, les frais forfaitisés, les frais hors forfait et le bouton « valider », puis on fait des include de toutes les vues à afficher.

Voici la méthode permettant la mise à jour des frais forfaitisés :

```
public function majFraisForfaitEmpCompt($idVisiteur, $mois, $lesFrais, $idEmpCompt){
    $lesCles = array_keys($lesFrais);
    $today = date("y-m-d");
    foreach($lesCles as $unIdFrais){
        $qte = $lesFrais[$unIdFrais];
        $req = "update ligneFraisforfait set ligneFraisforfait.quantite = $qte, idUtilisateur = '$idEmpCompt', dateModif = '$today'
        where ligneFraisforfait.idvisiteur = '$idVisiteur' and ligneFraisforfait.mois = '$mois'
        and ligneFraisforfait.idfraisforfait = '$unIdFrais'";
        PdoGsb::$monPdo->exec($req);
    }
}
```

On récupère la clé de chaque élément du tableau et la date du jour au format attendu dans la base de données. Puis pour chaque ligne de frais forfaitisé, un update est effectué dans la base de données. L'update met à jour la ligne de frais avec la nouvelle quantité, mais aussi avec l'id de l'utilisateur connecté et la date de modification qui est la date du jour.

D. Suppression d'un frais hors forfait

L'employé comptable peut supprimer un frais en cliquant sur le bouton supprimer à la fin de la ligne de frais.

Descriptif des éléments hors forfait

Date	Libellé	Montant	
2019-10-24	repas avec praticien	32.00	Supprimer ce frais
2019-10-21	taxi	58.00	Supprimer ce frais

J'ai ajouté une méthode dans le modèle afin de permettre la suppression de la ligne. L'employé commerciale réalise une suppression logique, c'est-à-dire que la ligne n'est pas supprimée de la base de données mais que la colonne supprimerO_N aura comme valeur 'O'. Ainsi on garde une trace du frais hors forfait saisi, mais on ne l'affiche plus et on ne le compte pas dans le montant total remboursé au visiteur.

On réalise donc un update du frais hors forfait dont l'id est passé en paramètre, la colonne supprimerO_N contiendra 'O', l'idUtilisateur sera l'id de l'utilisateur connecté et la dateModif sera la date du jour au format attendu par la base de données. Voici la méthode réalisant cela :

```
public function suppressionFraisHorsForfaitEmpCompt($idFrais, $idEmpCompt){
    $req = "update ligneFraisHorsForfait set supprimerO_N='O', idUtilisateur = '$idEmpCompt', dateModif = '$today'
    where ligneFraisHorsForfait.id = $idFrais ";
    PdoGsb::$monPdo->exec($req);
}
```

Pour la suppression, j'ai ajouté un case dans le switch du contrôleur c_gererValidation.php. On récupère l'id du frais à supprimer, le mois et l'id du visiteur à laquelle correspond la fiche de frais. On appelle ensuite la méthode qui permet de faire une suppression logique du frais hors forfait. Ensuite on appelle à nouveau

toutes les méthodes permettant l’affichage de la liste déroulante, des frais forfaitisés, des frais hors forfait et du bouton « valider », puis on fait des include de toutes les vues à afficher.

```
case 'supprimerFraisHorsForfait':{
    //On récupère les données
    $idFraisHF = $_REQUEST['idFrais'];
    $idVisiteurSelect = $_REQUEST['idVisiteurSelect'];
    $moisSelect = $_REQUEST['moisSelect'];
    //On supprime (update et non delete) la ligne de frais hors forfait
    $pdo->supprimerFraisHorsForfaitEmpCompt($idFraisHF, $idUtilisateur);
    //On récupère les frais forfait et hors forfait du visiteur et du mois sélectionné
    $lesFraisForfait= $pdo->getLesFraisForfait($idVisiteurSelect,$moisSelect);
    $lesFraisHorsForfait = $pdo->getLesFraisHorsForfaitValides($idVisiteurSelect,$moisSelect);
    //On récupère les fiches de frais dans un état cloturé
    $lesFichesFraisCL = $pdo->getLesInfosFichesFraisCL();

    include("vues/v_listeFicheCL.php");
    include("vues/v_listeFraisForfaitCL.php");
    include("vues/v_listeFraisHorsForfaitCL.php");
    include("vues/v_validerFicheFrais.php");
    break;
}
```

Lorsque l’utilisateur supprime une ligne, celle-ci n’apparaît plus dans le tableau comportant les frais hors forfait de la fiche de frais.

E. Validation de la fiche de frais

Lorsqu’une fiche de frais est validée, le montant validé doit être enregistré dans la base de donnée. Pour cela, il faut calculer le montant total des frais forfaitisés et des frais hors forfait puis faire la somme des deux. J’ai donc réalisé la méthode getMontantFraisForfait. Cette méthode effectue un select dans la base de données et multiplie la quantité par le montant du frais forfaitisé puis fait la somme de toutes les lignes de frais forfaitisé pour l’utilisateur et le mois passé en paramètre avant de retourner cette somme. Voici la méthode réalisant cela :

```
public function getMontantFraisForfait($idVisiteur, $mois){
    $req = "select sum(quantite * montant) from fraisforfait join lignefraisforfait on idFraisForfait= id
    where idVisiteur='$idVisiteur' and mois='$mois'";
    $rs = PdoGsb::$monPdo->query($req);
    $laLigne = $rs->fetch();
    return $laLigne['sum(quantite * montant)'];
}
```

J’ai ensuite ajouté la méthode getMontantFraisHorsForfait. Elle fait un select dans la base de données qui effectue la somme de toutes les lignes de frais hors forfait pour un mois et un utilisateur passé en paramètre et pour les lignes dont la colonne supprimerO_N contient ‘N’ ou null.

```
public function getMontantFraisHorsForfait($idVisiteur, $mois){
    $req = "select sum(montant) from lignefraishorsforfait where idVisiteur='$idVisiteur'
    and mois='$mois' and (supprimerO_N='N' or supprimerO_N IS NULL)";
```

Enfin, j’ai ajouté la méthode validerFicheFrais qui va faire un update dans la base de données. La fiche de frais aura pour état VA (validée), le montant validé sera la somme du montant total des frais forfaitisés et des frais hors forfait, l’idUtilisateur sera l’id de l’utilisateur connecté et la date de modification sera la date du jour. Voici la méthode réalisant cela :

```

public function validerFicheFrais ($idVisiteur, $mois,$idEmpCompt, $montantTotalValide){
    $today = date("y-m-d");
    $req = "update fichefrais set idEtat = 'VA', montantValide = $montantTotalValide,
    idUtilisateurValide = '$idEmpCompt', dateModif='$today'
    where idVisiteur='$idVisiteur' and mois='$mois'";
    PdoGsb::$monPdo->exec($req);
}

```

Pour effectuer la validation, j'ai ajouté un case dans le contrôleur c_gererValidation. On récupère tout d'abord l'id du visiteur médical et le mois de la fiche de frais sélectionnée. Ensuite on appelle les méthodes getMontantFraisHorsForfait et getMontantFraisForfait, avant de faire la somme des résultats retournés par la méthode. Puis on appelle la méthode validerFicheFrais en passant en paramètre l'id du visiteur médical, le mois, l'id de l'employé comptable connecté ainsi que le résultat de la somme des frais effectuée précédemment. Enfin, on appelle la méthode permettant d'obtenir toutes les fiches de frais à valider et la vue permettant de l'afficher. Voici le case correspondant à cela :

```

case 'validerFicheFrais':{
    //On récupère le visiteur et mois sélectionné
    $idSelect = explode("/",$_REQUEST['idVisiteurMois']);
    $idVisiteurSelect = $idSelect[0];
    $moisSelect = $idSelect[1];

    //On récupère dans la base de données les montants de frais forfait et hors forfait.
    $montantFraisForfait = $pdo->getMontantFraisForfait($idVisiteurSelect,$moisSelect);
    $montantFraisHorsForfait = $pdo->getMontantFraisHorsForfait($idVisiteurSelect,$moisSelect);

    //On calcule le montant total
    $montantTotalValide = $montantFraisForfait + $montantFraisHorsForfait;

    //On valide la fiche
    $pdo->validerFicheFrais($idVisiteurSelect,$moisSelect, $idUtilisateur, $montantTotalValide);

    //On récupère les fiches de frais dans un état cloturé
    $lesFichesFraisCL = $pdo->getLesInfosFichesFraisCL();

    include("vues/v_listeFicheCL.php");
    break;
}

```

Lorsque l'employé médical clique sur la bouton « valider » en bas de la page, la fiche est validée.

2019-10-07	location équipement vidéo/sonore	109.00	Supprimer ce frais
<div>Valider</div>			

Après que l'utilisateur a cliqué sur « valider », il doit à nouveau sélectionner une fiche de frais à valider. Voici l'écran qu'il obtient :



Suivi du remboursement des frais

[Valider les fiches de frais](#)
[Déconnexion](#)

Les fiches de frais à valider

Fiche de frais à sélectionner :

Fiche de frais : Cadic Eric 10/2019 ▾

Valider Effacer