# COMPTE RENDU FINAL

## NOM DU PROJET : SERVICE WEB ET SITE

SEMAINE DU 24 AOÛT AU 11 OCTOBRE 2023

PRÉPARÉ PAR :
WAN-TANG-KIVAN LUCAS
HOARAU FLORIAN
VINGADASSALOM BRIAN
ARTHUR NOAH
ABDOU ABDALLAH RADJA
AHOVEY SAINT-GEORGE

# 1. Contexte

Au sein du laboratoire Galaxy Swiss Bourdin (GSB), notre équipe, composée de développeurs et de spécialistes réseau, s'engage dans le développement d'une application web dédiée à la gestion des frais engendrés par les visiteurs médicaux lors de leurs déplacements et actions de terrain. Cette application est cruciale pour la prise en charge comptable de ces dépenses. Notre mission consiste à poursuivre le développement de l'application, tout en garantissant une solution d'hébergement sécurisée. Pour y parvenir, nous allons répartir efficacement les tâches au sein de notre équipe, en capitalisant sur les compétences de chacun, dans le but de fournir une solution opérationnelle et fonctionnelle pour le laboratoire GSB.

# 2. Objectifs à tenir compte (pour les clients)

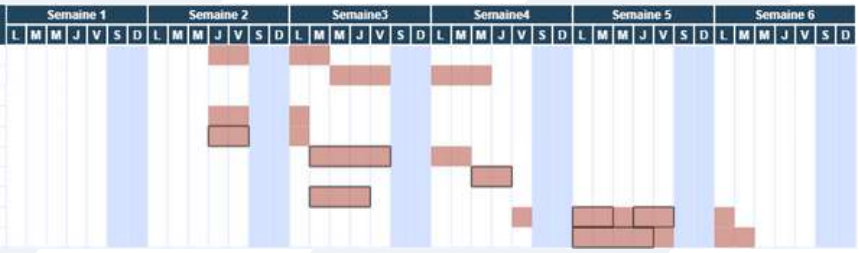
Les objectifs essentiels liés au développement de l'application de gestion des frais sont la convivialité de l'interface pour une utilisation aisée, la fiabilité et la précision des enregistrements, l'accessibilité depuis divers appareils, la sécurité rigoureuse des données, la rapidité et l'efficacité de l'application, un support technique réactif, la conformité aux réglementations, ainsi qu'une solution rentable.

L'ensemble de ces objectifs vise à simplifier la saisie et la gestion des dépenses tout en garantissant la sécurité, la précision et l'efficacité, tout en respectant les contraintes légales, afin de répondre aux besoins et aux attentes des clients de GSB.

# Gestion de projet

Nous avons réalisé un diagramme de Gantt pour avoir une vue d'ensemble du projet ainsi que pour gérer au mieux l'avancement des tâches de chacun

| 1 | Nom des tâches | Option | Durée (jours) | Tâches antérieur |
|---|---|---|---|---|
| 1 | Vérifications du fonctionnement des saisies et affichages | SLAM | 4 | aucune |
| 2 | Codage de la partie Service comptable | SLAM | 6 | 1 |
| 3 | Finition de l'architecture applicative | SLAM | | 1,2 |
| 4 | Mise en place des services sur le serveur web GSB | SISR | 3 | aucune |
| 5 | Mise en place DNS | SISR | 3 | aucune |
| 6 | Mise en place de la base de données sur un autre serveur | SISR | 7 | aucune |
| 7 | Mise en place d'un outil pour la MAJ de la base de donnée | SISR | 2 | 6 |
| 8 | Mise en place de sécurités (https...) sur le serveur | SISR | 3 | 5 |
| 9 | Solution de répartition des charges avec second serveur | SISR | 8 | 4,7 |
| 10 | Sécurisation des échanges entre le serveur web et la BD | SISR | 8 | 4,6 |




Aussi pour qu'on puisse communiquer au mieux sur l'avancement global du projet nous avons décidé de mettre en place un espace de travail collaboratif sur google drive

Estimation du projet :

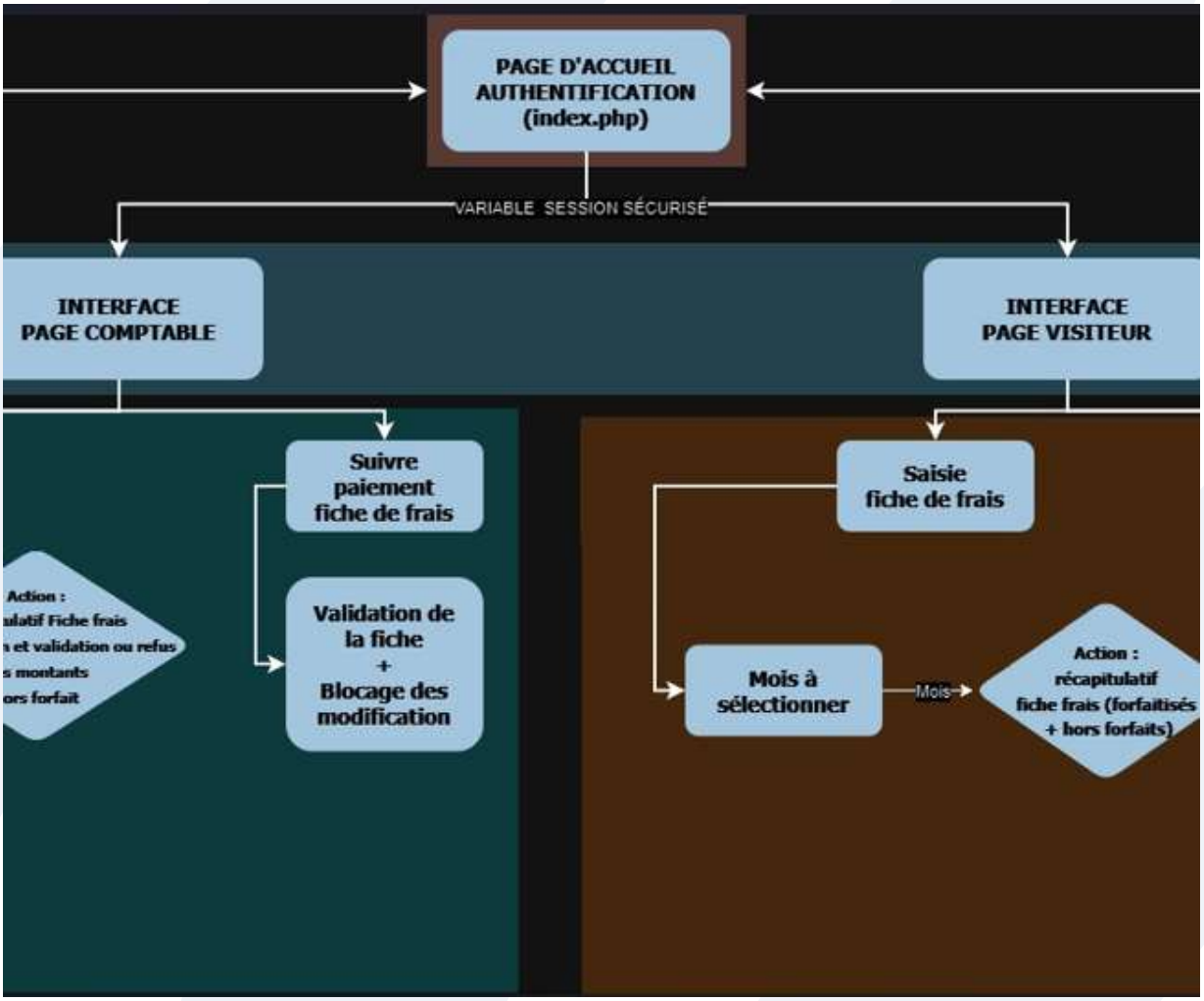
**Calcul des ressources homme/heure** --> On multiplie le nombre d'heures durant lesquelles on a travaillé sur le projet (sur une semaine) par le nombre total d'employés. On multiplie le tout par le nombre total de semaines qu'il a fallu pour livrer ce projet. En sachant que nous sommes 6 dans l'équipe, on alors : **4 x 6 x 6 = 144 homme/heure**

Calcul du coût total du projet --> **(Nombre d'employés) x (Salaire net par semaine) x (Durée du projet en semaines).**
Ce qui nous donne : **6 x 400 x 6 = 14 400€**

# Partie technique SLAM

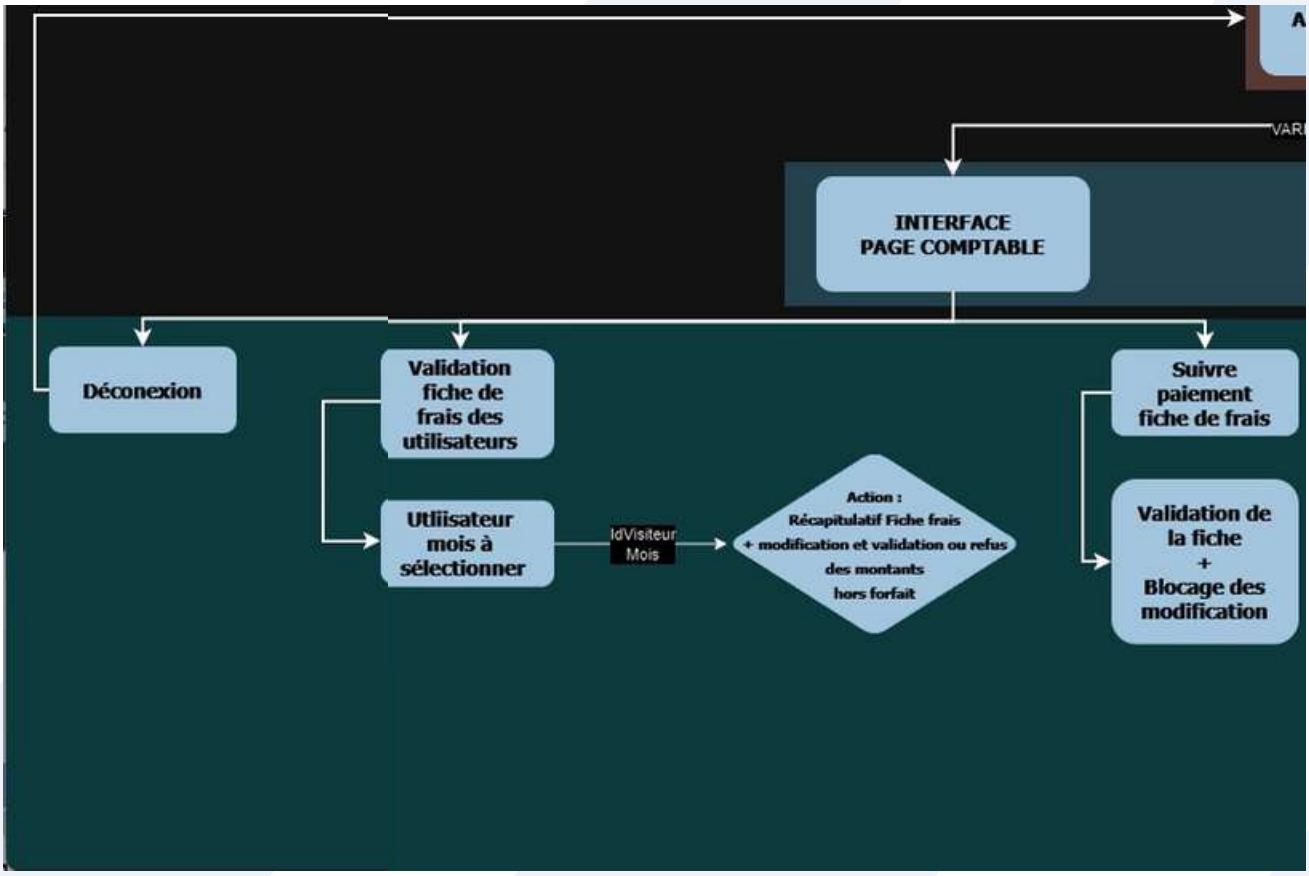
## Schéma de principe de la solution:

1. Page d'Accueil: L'utilisateur doit s'authentifier en entrant son nom d'utilisateur et son mot de passe.




2. Après l'authentification, l'utilisateur sera redirigé vers l'une des deux pages, en fonction de son rôle :

**Page du Comptable:** Le comptable aura accès aux fonctionnalités suivantes :



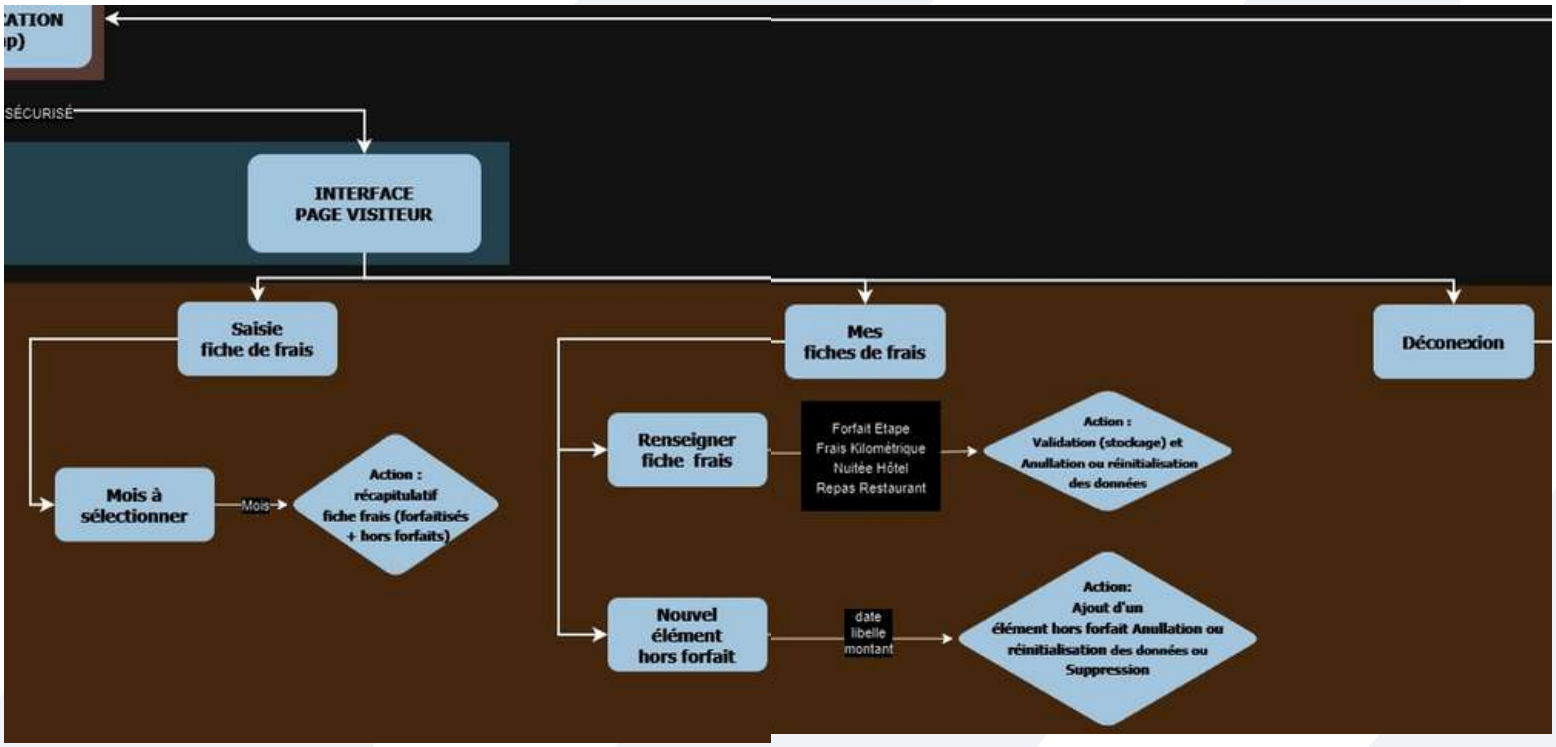

- **Suivi de Paiement des Fiches de Frais:** Suivre le statut des paiements des fiches de frais soumises par les utilisateurs.

- **Validation des Fiches de Frais des Utilisateurs:** Valider ou refuser les fiches de frais soumises par les utilisateurs pour un mois donné. Cela implique :

- Récapitulatif de la Fiche de Frais.
- Possibilité de modifier la fiche de frais soumise par l'utilisateur.
- Validation ou refus des montants hors forfait (utilisant une valeur booléenne 1 pour validation, 0 pour refus). Ces montants seront pris en compte dans le calcul du montant final lorsque la fiche est validée dans "Suivi de Paiement des Fiches de Frais".

- **Déconnexion:** Se déconnecter du système.

**Page du Visiteur:** Le visiteur aura accès aux fonctionnalités suivantes :




- **Saisie de Fiche de Frais:** Créer une nouvelle fiche de frais pour le mois en cours. Cette fiche sera ensuite envoyée au comptable pour traitement.

- **Mes Fiches de Frais:** Consulter les fiches de frais soumises pour différents mois où le visiteur a créé une fiche. Ces données seront stockées dans la base de données.

- **Déconnexion:** Se déconnecter du système.

## Principe général de l'architecture MVC et modifications apportées:

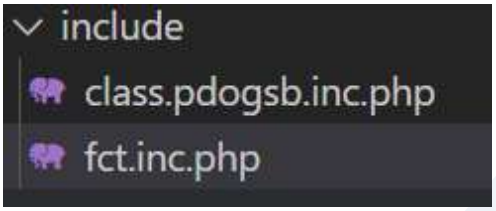
Notre projet de site web repose sur l'architecture MVC, qui est un modèle de conception pour le développement d'applications logicielles. L'architecture MVC est composée de trois principaux éléments qui permettent la séparation claire des fonctionnalités . Chaque composant a un rôle spécifique et ne se préoccupe que de ce rôle, ce qui rend le code plus maintenable, flexible et facilite le travail en équipe. Ces trois principaux éléments sont:

Modèle (Model) : Le modèle est responsable de la gestion des données et de la logique métier de l'application. Il communique avec la base de données, effectue des opérations sur les données et retourne des résultats au contrôleur.
Vue (View) : La vue est chargée de l'interface utilisateur. Elle affiche les données au format approprié pour les utilisateurs et collecte les entrées utilisateur. La vue n'a pas de logique métier propre, elle se contente d'afficher les données.
Contrôleur (Controller) : Le contrôleur est le cerveau de l'application. Il reçoit les demandes de l'utilisateur depuis la vue, traite ces demandes en fonction de la logique métier définie dans le modèle, puis renvoie les résultats appropriés à la vue pour affichage.

Nous avons adopté l'architecture MVC du site comme base, mais nous avons également apporté des modifications pour répondre à nos besoins spécifiques. Ces modifications incluent :

La modification du modèle : Nous avons adapté le modèle pour gérer les données d'authentification, de gestion de frais . Cela inclut la création de nouvel attribut "comptable" (types boolen)

dans la base de données.




c_connexion.php

controleurs > c_connexion.php > ...

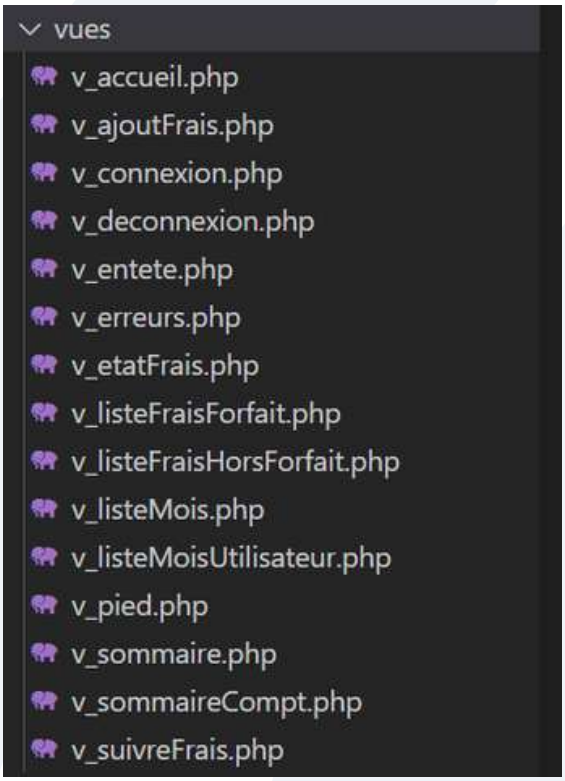
```php
        include("vues/v_connexion.php");
        break;
    }
    case 'valideConnexion':{
        $login = $_REQUEST['login'];
        $mdp = $_REQUEST['mdp'];
        $visiteur = $pdo->getInfosVisiteur($login,$mdp);
        if(!is_array( $visiteur)){
            ajouterErreur("Login ou mot de passe incorrect");
            include("vues/v_erreurs.php");
            include("vues/v_connexion.php");
        }
        else{
            $id = $visiteur['id'];
            $nom =  $visiteur['nom'];
            $prenom = $visiteur['prenom'];
            $fonction = $visiteur['fonction'];
            connecter($id,$nom,$prenom,$fonction);
            if ($fonction == 0){
                // include("vues/v_entete.php");
                include("vues/v_sommaire.php");
            }else{
                // include("vues/v_enteteC.php");
                include("vues/v_sommaireCompt.php");
            }
        }
        break;
    }
    default :{
        include("vues/v_connexion.php");
        break;
    }
```

Grâce à la fonction “connecter”:

```php
function connecter($id,$nom,$prenom,$fonction){
    $_SESSION['idVisiteur']= $id;
    $_SESSION['nom']= $nom;
    $_SESSION['prenom']= $prenom;
    $_SESSION['comptable']= $fonction;
}
```

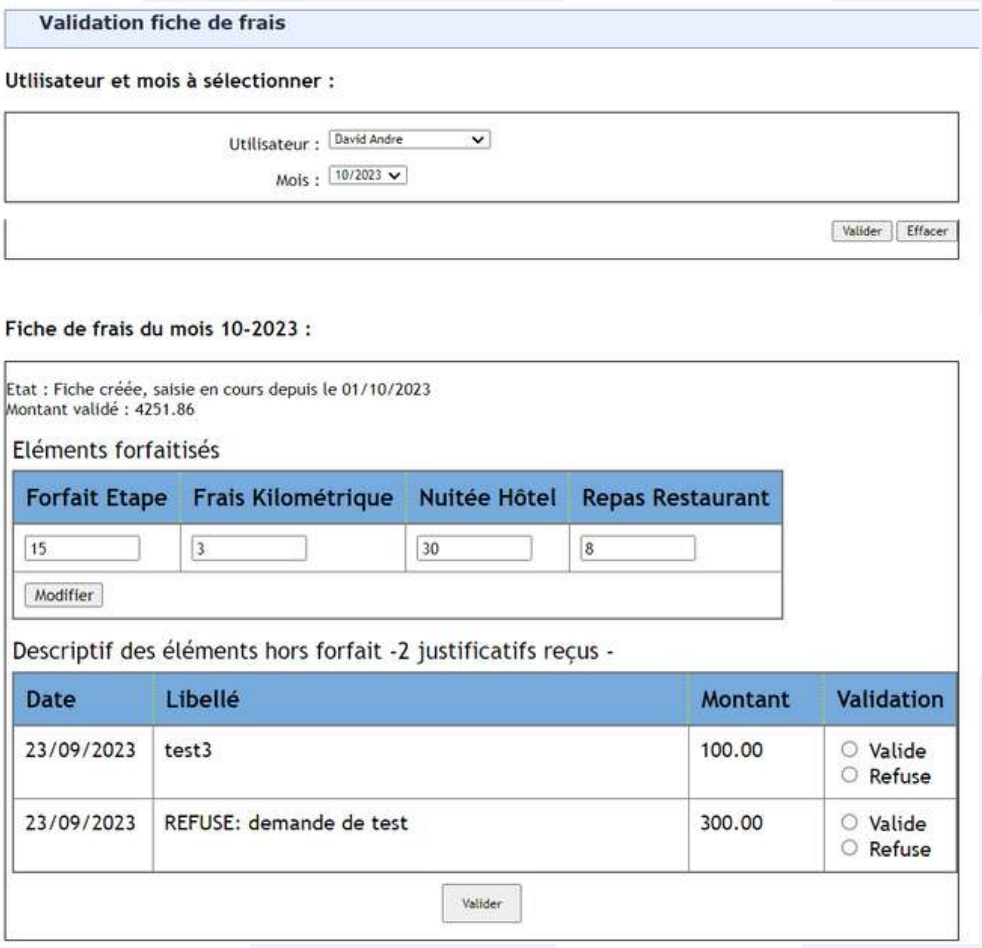
Il vérifiera les conditions ci-dessus et laissera l'utilisateur rentrer si celui-ci a bien mis ses identifiants sinon, dans le cas échéant, il y aura un message d'erreur.
On a donc la possibilité de récupérer ces informations (id, nom, prénom, fonction de l'utilisateur) grâce à la variable $_SESSION de la fonction "connecter". Ces informations pourront alors être réutilisées dans d'autres pages.

Personnalisation de la vue : Nous avons personnalisé la vue pour refléter la conception et l'expérience de l'utilisateur comptable. Cela inclut la création de modèles de page, la définition de styles CSS spécifiques pour la session comptable.

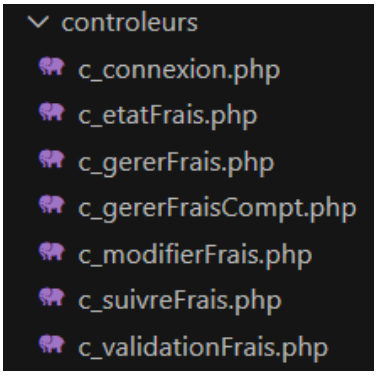



Nous avons ajouté une vue spécifique aux comptables appelée « v_sommaireCompt.php »:


Ainsi qu'une nouvelle vue intitulée « v_suivreFrais.php ».




Personnalisation du contrôleur : Le contrôleur a été ajusté pour gérer les actions de gestion de frais par les comptables. Cela a impliqué la création de contrôleurs personnalisés comme c_gererFraisCompt.php et c_suivreFrais.php .




Pour répondre aux besoins de l'application, trois contrôleurs supplémentaires ont été créés, à savoir « c_suivreFrais.php », « c_modifierFrais.php » et « c_validationFrais.php ».
Le contrôleur « c_suivreFrais » permet aux comptables de visualiser les frais forfaitaires, de les modifier et de valider ou refuser les montants hors forfait.

```php
<?php
/*
 * GSB Project 2022
*/
include("vues/v_sommaireCompt.php");
$action = $_REQUEST['action'];
$idVisiteur = $_SESSION['idVisiteur'];
switch($action){
    case 'selectionnerMois':{

        $lesMois=$pdo->getLesMoisDisponibles($idVisiteur);
        // Afin de sélectionner par défaut le dernier mois dans la zone de liste
        // on demande toutes les clés, et on prend la première,
        // les mois étant triés décroissants
        $lesCles = array_keys( $lesMois );
        // $moisASelectionner = $lesCles[0];
        // include("vues/v_listeMois.php");
        // break;
        // $lesCles = array_keys($lesMois);
        if (!empty($lesCles)) {
            $moisASelectionner = $lesCles[0];
            include("vues/v_listeMoisUtilisateur.php");
        } else {
            // Gérer le cas où $lesCles est vide (par exemple, afficher un message d'erreur).
            echo "Vous n'avez pas encore saisie votre fiche de frais.";
        }
        break;
    }
    case 'voirEtatFrais':{
        if (isset($_REQUEST['lstMois'])){

            $leMois = $_REQUEST['lstMois'];
            $Utilisateur = $_REQUEST['lstUtilisateur'];
        }
        // $Validation = $_POST['validation'];
```

```php
if (isset($leMois) && isset($Utilisateur)){

    // echo $Utilisateur;
    $lesMois=$pdo->getLesMoisDisponiblesComptable();
    $moisASelectionner = $leMois;
    include("vues/v_listeMoisUtilisateur.php");
    $lesFraisHorsForfait = $pdo->getLesFraisHorsForfait($Utilisateur,$leMois);
    // var_dump($lesFraisHorsForfait);
    $lesFraisForfait= $pdo->getLesFraisForfait($Utilisateur,$leMois);
    $lesInfosFicheFrais = $pdo->getLesInfosFicheFrais($Utilisateur,$leMois);
    // Récupérez les quantités saisies par l'utilisateur depuis le tableau $lesFrai
    foreach ($lesFraisForfait as $unFraisForfait) {
        $idFrais = $unFraisForfait['idfrais'];
        $quantite = $unFraisForfait['quantite'];

        // Utilisez la clé $idFrais pour récupérer la valeur correspondante depuis
        if (isset($fraisForfaitaires[$idFrais])) {
            $valeurFrais = $fraisForfaitaires[$idFrais];
            // Calculez le total pour ce type de frais en multipliant la quantité p
            $montantValide += $quantite * $valeurFrais;
        }
    }
    if (isset($Utilisateur)&& isset($leMois) && isset($montantValide)){
        $pdo->mettreAJourMontantValide($Utilisateur, $leMois, $montantValide);

    }

    // ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

    // À ce stade, $montantValide contient le montant valide calculé
    // Vous pouvez l'utiliser comme bon vous semble, par exemple, l'afficher
    $numAnnee =substr( $leMois,0,4);
    $numMois =substr( $leMois,4,2);
}
if (!$lesFraisForfait){
    echo 'Pas de fiche de frais pour ce visiteur ce mois';
}else{

    $libEtat = $lesInfosFicheFrais['libEtat'];
    $montantValide = $lesInfosFicheFrais['montantValide'];
    $nbJustificatifs = $lesInfosFicheFrais['nbJustificatifs'];
    $dateModif =  $lesInfosFicheFrais['dateModif'];
    $dateModif =  dateAnglaisVersFrancais($dateModif);
    // rajout d'un include  formulaire modification ou bien  rendre le tableau dynami
    include("vues/v_suivreFrais.php");
    break;
}
```

Le contrôleur « c_modifierFrais » effectue la modification des valeurs présentes dans le formulaire de la vue "v_suivreFrais". Il communique avec la base de données pour remplacer les valeurs existantes lors de la modification.

```php
<?php
   $utilisateur = $_POST['idUtilisateur'];
   $moisSelectionner = $_POST['moisSelectionner'];
//    echo $utilisateur;
//    echo $moisSelectionner;
   require_once ('../include/class.pdogsb.inc.php');
   $pdoGsb = PdoGsb::getPdoGsb();

   if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (isset($_POST['lesFrais'])) {
        $lesFrais = $_POST['lesFrais'];

        // Afficher les valeurs soumises pour débogage
        // echo "Valeurs soumises : <pre>";
        // print_r($lesFrais);
        // echo "</pre>";

        foreach ($lesFrais as $idFrais => $quantite) {
          $pdoGsb->majFicheFraisForfait($utilisateur, $moisSelectionner, $idFrais, intval($quantite));

        }


    } else {
    }


}

?>
```

Pour mettre à jour les valeurs, on utilisera la fonction "majFicheFraisForfait" :

```php
    public function majFicheFraisForfait($idUtilisateur, $mois, $idFrais, $quantite) {
        $req = "UPDATE lignefraisforfait SET quantite = $quantite
        WHERE idFraisForfait = '$idFrais' AND idVisiteur = '$idUtilisateur' AND mois = '$mois'";
        PdoGsb::$monPdo->exec($req);
    }
```

Le contrôleur "c_validationFrais" est responsable de la validation des fiches de frais non-forfaitaires. S'il est approuvé, le montant hors-forfait sera ajouté au montant valide, sinon le libellé affichera "REFUSÉ:" :

```php
require_once ('../include/class.pdogsb.inc.php');
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Vérifiez si le formulaire a été soumis

    // Récupérez les valeurs des éléments hors forfait
        foreach ($_POST as $key => $value) {
            // La clé est de la forme "validation_idFraisHors"
            // Vous pouvez la diviser pour obtenir l'ID et la valeur de validation
            $parts = explode("_", $key);
            if (count($parts) === 2 && $parts[0] === "validation") {
                $idFraisHors = $parts[1]; // L'ID de l'élément hors forfait
                $validationValue = $value; // La valeur de validation (1 pour valide, 0 pour refusé)
                $pdoGsb = PdoGsb::getPdoGsb();
                // Récupérez le montant et l'idFraisHors correspondants
                $montant = $_POST["montant_" . $idFraisHors];
                $date = $_POST["date_" . $idFraisHors];
                $idVisiteur = $_POST["idVisiteur_" . $idFraisHors];
                $idFraisHorsValue = $_POST["idFraisHors_" . $idFraisHors];
                $montantValide = $_POST["montantFinal_" . $idFraisHors];
                $moisSelectionner = $_POST["moisSelectionner_" . $idFraisHors];
                $montantValide2 = $pdoGsb->getMontantValideFicheFrais($idVisiteur, $moisSelectionner);
                 $date = "23/09/2023";
                 // Divisez la date en jour, mois et année
                 list($jour, $mois, $annee) = explode("/", $date);
                 // Formatez la date en "202309"
                 $dateFormatee = $annee . $mois;
                 // echo $dateFormatee;
                 // echo '<br>';
                 $pdoGsb->validation($idFraisHorsValue, $dateFormatee, $idVisiteur, $validationValue,
                 floatval($montant),floatval($montantValide2),$moisSelectionner );
```

Afin de mettre à jour les fiches validées ou refusées, on utilisera la fonction "validation" :

```php
public function validation($idFraisHors,$leMois,$idVisiteur,$Validation,$montant, $montantValide,$moisSelectionner) {
    if(isset($Validation)){

        if ($Validation == 1) {
            // Faire quelque chose si la valeur est égale à 1


            $req1 = "update fichefrais set idEtat = 'VA'
            where idVisiteur = '$idVisiteur' and mois = '$leMois'";
            PdoGsb::$monPdo->exec($req1);
            $req = "update lignefraishorsforfait set validation = 1
            where id = $idFraisHors and idVisiteur = '$idVisiteur'";
            PdoGsb::$monPdo->exec($req);
            $req3 = "UPDATE fichefrais SET montantValide = montantValide + $montant
            WHERE idVisiteur = '$idVisiteur' AND mois = '$moisSelectionner'";
            PdoGsb::$monPdo->exec($req3);

        //    $montantHorsForfait = $this->getMontantFraisHorsForfait($idFraisHors);

        } elseif ($Validation == 0) {
            // Faire quelque chose si la valeur est égale à 0
            $req4 = "update fichefrais set idEtat = 'VA'
            where idVisiteur = '$idVisiteur' and mois = '$leMois'";
            PdoGsb::$monPdo->exec($req4);
            $req = "update lignefraishorsforfait set validation = 0
            where id = $idFraisHors and idVisiteur = '$idVisiteur'";
            PdoGsb::$monPdo->exec($req);
            $req5 = "UPDATE lignefraishorsforfait SET libelle = CONCAT('REFUSE: ', libelle)
            WHERE validation = 0 and id = $idFraisHors and idVisiteur = '$idVisiteur' AND id = '$idFraisHors'";
            PdoGsb::$monPdo->exec($req5);
        } else {
            // Faire quelque chose si la valeur n'est ni 0 ni 1
            echo 'Veuillez remplir le formulaire.';
        }
    }
}
```

Les avantages de l'utilisation de MVC sont donc la séparation des actions du site, la réutilisabilité du code, la facilité de maintenance et la flexibilité des pages du site.

## Évolution de la base de données :

Dans un premier temps, il a été essentiel de relier la table "lignefraishorsforfait" à celle des "visiteurs". Cette liaison a permis de collecter des informations, telles que l'identifiant du visiteur, le libellé, le mois, la date et le montant, provenant de la table "visiteur". Ces données ont ensuite été transférées dans la table"lignefraishorsforfait". Cette démarche a eu pour objectif de minimiser les redondances et d'éviter la survenue de bugs dans la base de données.

**Table: lignefraishorsforfait**

**Columns:**

| | |
|---|---|
| **id** | int(11) AI PK |
| **idVisiteur** | char(4) |
| mois | char(6) |
| libelle | varchar(100) |
| date | date |
| montant | decimal(10,2) |
| validation | tinyint(1) |

| id | idVisiteur | mois | libelle | date | montant | validation |
|---|---|---|---|---|---|---|
| 4 | a55 | 202309 | RIEN | 2023-08-06 | 300.00 | |
| 13 | a17 | 202310 | test3 | 2023-09-23 | 100.00 | 1 |
| 21 | a17 | 202310 | REFUSE: demande de test | 2023-09-23 | 300.00 | 0 |
| 22 | b13 | 202310 | REFUSE: EXEMPLE | 2023-10-03 | 300.00 | 0 |
| 24 | b13 | 202310 | RIEN | 2023-10-02 | 20.00 | 1 |
| 26 | b16 | 202310 | EXEMPLE | 2023-10-01 | 100.00 | 1 |
| 27 | b16 | 202310 | REFUSE: test | 2023-10-02 | 200.00 | 0 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Un attribut supplémentaire, nommé "validation", a été intégré à la base de données. Cet attribut permet aux comptables et aux visiteurs la possibilité de vérifier si une fiche a été correctement validée ou non. Cette fonctionnalité renforce le contrôle et la transparence dans le processus de validation des fiches.

## Évolution de la base de données :

Par la suite, nous avons ajouté un nouvel attribut appelé "comptable" la table "visiteur.

**Table: visiteur**

**Columns:**

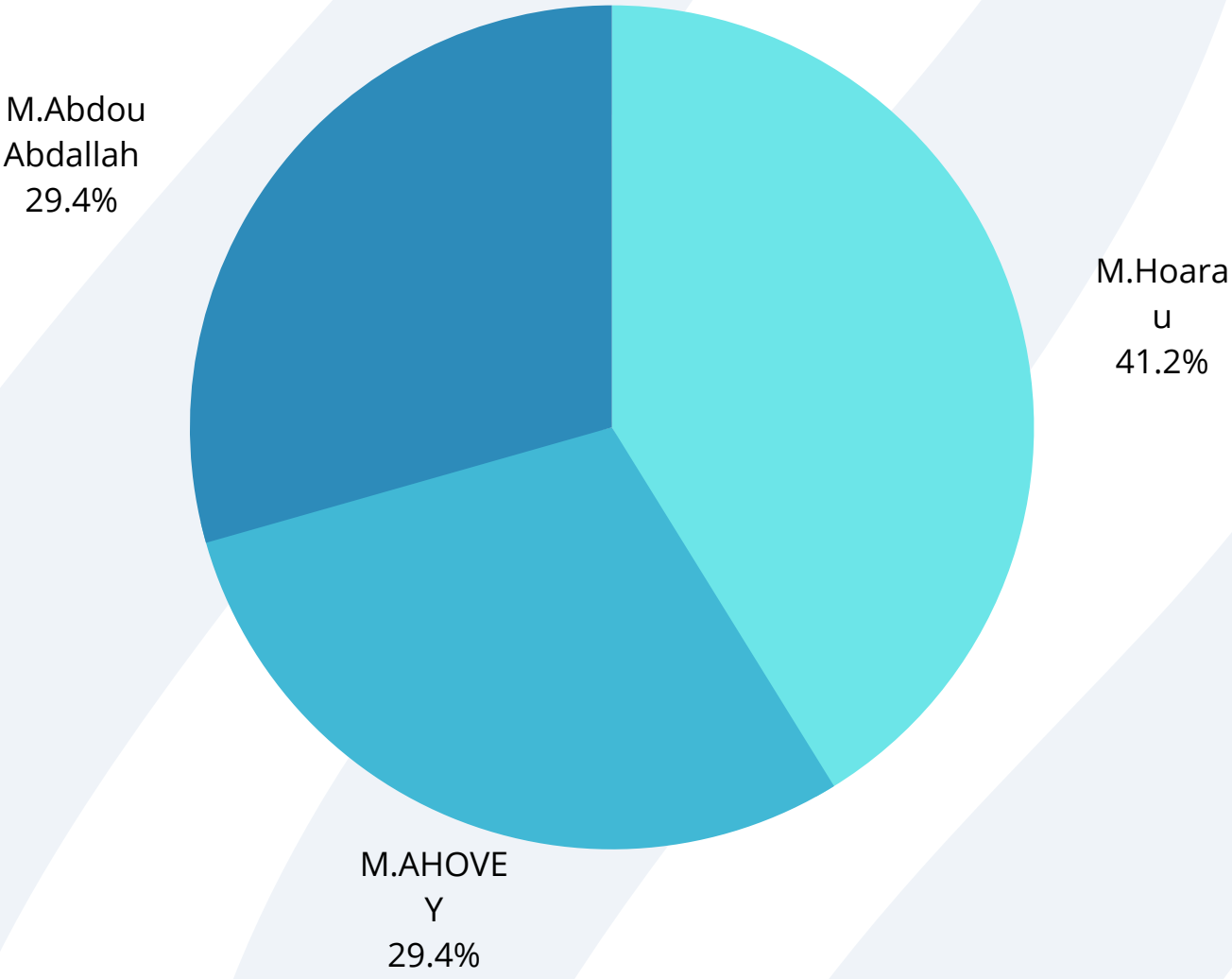
| | |
|---|---|
| id | char(4) PK |
| nom | char(30) |
| prenom | char(30) |
| login | char(20) |
| mdp | char(20) |
| adresse | char(40) |
| cp | char(5) |
| ville | char(30) |
| dateEmbauche | date |
| comptable | tinyint(1) |

| id | nom | prenom | login | mdp | adresse | cp | ville | dateEmbauche | comptable |
|---|---|---|---|---|---|---|---|---|---|
| a131 | Villechalane | Louis | lvillachane | jux7g | 8 rue des Charmes | 46000 | Cahors | 2005-12-21 | 1 |
| a17 | Andre | David | dandre | oppg5 | 1 rue Petit | 46200 | Lalbenque | 1998-11-23 | 0 |
| a55 | Bedos | Christian | cbedos | gmhxd | 1 rue Peranud | 46250 | Montcuq | 1995-01-12 | 0 |
| a93 | Tusseau | Louis | ltusseau | ktp3s | 22 rue des Ternes | 46123 | Gramat | 2000-05-01 | 0 |
| b13 | Bentot | Pascal | pbentot | doyw1 | 11 allée des Cerises | 46512 | Bessines | 1992-07-09 | 0 |
| b16 | Bioret | Luc | lbioret | hrjfs | 1 Avenue gambetta | 46000 | Cahors | 1998-05-11 | 0 |
| b19 | Bunisset | Francis | fbunisset | 4vbnd | 10 rue des Perles | 93100 | Montreuil | 1987-10-21 | 0 |

Pour cela, 2 valeurs ont été ajoutées (1 et 0).
Une valeur de "1" permettant de savoir s'il s'agit d'un comptable, tandis qu'une valeur de "0" permettant de savoir s'il s'agit d'un visiteur.
Permettant ainsi de différencier plus facilement les visiteurs et les comptables.

En ce qui concerne l'opérateur de comparaison, il a été utilisé pour évaluer les valeurs de l'attribut "comptable". Afin de définir de manière précise si un utilisateur est un comptable ou un visiteur, en se basant sur les valeurs "true" et "false".

**RÉPARTITION DU TRAVAIL :**



L'équipe de développement et les spécialistes réseaux du laboratoire Galaxy Swiss Bourdin (GSB) ont travaillé sur un projet crucial : la création d'une application web dédiée à la gestion des frais médicaux des visiteurs médicaux. Le but était de simplifier la saisie et la gestion des dépenses tout en garantissant la sécurité, la précision et l'efficacité et en respectant les contraintes légales, afin de répondre aux besoins et attentes des clients de GSB.

Cependant, l'équipe a rencontré des défis importants, notamment en termes de cohésion et de compréhension de l'architecture, en particulier dans la partie SLAM. Cela a retardé le projet et empêché la réalisation complète des résultats prévus. Les écarts entre les résultats prévus et les résultats réels peuvent être attribués à plusieurs facteurs.

Les difficultés de cohésion et de compréhension de l'architecture dans la partie SLAM ont eu un impact sur la qualité du travail et ont rendu difficile l'atteinte des résultats prévus. De plus, les contraintes de temps ont limité la capacité de l'équipe à achever toutes les fonctionnalités initialement prévues. Enfin, les difficultés de communication et de collaboration au sein de l'équipe ont également contribué aux écarts.

Malgré tout, l'équipe a réussi à créer un site fonctionnel, avec la suppression des bugs du côté client (visiteur) et la création d'un espace privé pour les comptables afin de valider les fiches de frais des visiteurs. Cependant, il reste des améliorations à apporter pour atteindre pleinement les objectifs du projet.

Le projet a fait face à des défis significatifs, mais il a néanmoins atteint certains de ses objectifs du côté du déploiement des services réseaux. La partie la plus compliquée a été la mise en place de la résolution des noms de domaines ainsi que l'accès aux différents sites suite à leur hébergement. Bien que le DNS ait été fonctionnel, il fallait comprendre comment faire pour pouvoir y accéder depuis un navigateur web tout en faisant en sorte que les sites puissent être mis à jour dynamiquement avec la base de données (liaison). Tout cela nous a demandé du temps pour comprendre ce fonctionnement et les écarts peuvent s'expliquer par le délai de livraison qui a été raccourci par rapport à ce qui était prévu.