

# Tâche 1 : Validation d'une fiche de frais

PROJET : Application web de gestion des frais	Description cas d'utilisation
<b>Nom cas d'utilisation :</b> Valider fiche de frais	
<b>Acteur déclencheur :</b> Comptable	
<b>Pré conditions :</b> Utilisateur Comptable authentifié Toutes les fiches du mois qui vient de s'achever sont clôturées par un script Le comptable dispose de tous les éléments pour évaluer la conformité des frais forfaitisés	
<b>Post conditions :</b> Néant	
<b>Scénario nominal :</b>  <ol style="list-style-type: none"><li>1. L'utilisateur demande à valider les fiches de frais</li><li>2. Le système propose de choisir le visiteur et le mois concernés</li><li>3. L'utilisateur sélectionne les informations et valide</li><li>4. Le système affiche le détail de la fiche de frais –frais forfaitisés et hors forfait</li><li>5. L'utilisateur actualise les informations des frais forfaitisés.</li><li>6. Le système indique que la modification a été prise en compte et affiche les informations actualisées.</li><li>7. L'utilisateur demande la suppression des lignes de frais hors forfait non valides</li><li>8. Le système modifie le libellé en ajoutant en début le texte « REFUSE : ».</li><li>9. L'utilisateur valide la fiche.</li><li>10. Le système passe la fiche à l'état « Validée » et met à jour la date de modification de la fiche.</li></ol>	
<b>Exceptions :</b> <ul style="list-style-type: none"><li>• 4-a : Aucune fiche de frais n'existe, le système affiche 'Pas de fiche de frais pour ce visiteur ce mois'. Retour au 2</li><li>• 7.a : L'utilisateur demande le report des frais hors forfait pour lesquels une facture acquittée n'a pas été reçue dans les temps.</li><li>• 8.a : Le système ajoute la ligne hors forfait dans la fiche du mois suivant et la supprime de la fiche courante. Si la fiche du mois suivant n'existe pas, le système génère une nouvelle fiche pour le visiteur en cours de traitement et pour le mois suivant. Cette nouvelle fiche a des valeurs à 0 pour les frais forfaitisés et est dans l'état « Saisie en cours ». Retour au 9</li><li>• 8.b : le texte ainsi complété dépasse la taille maximale du champ « libelle » : le texte est tronqué par la fin au nombre de caractères du champ « libelle »</li></ul>	
<b>Contraintes :</b>	
<b>Questions ouvertes :</b>	

Tout d'abord, nous devons modifier la méthode estConnecte() ;

```
require_once 'includes/fct.inc.php';
require_once 'includes/class.pdogs.inc.php';
session_start();
$pdo = PdoGsb::getPdoGsb();
$estConnecte = estConnecte();
require 'vues/v_entete.php';
$uc = filter_input(INPUT_GET, 'uc', FILTER_SANITIZE_STRING);
if ($uc && !$estConnecte) {
    $uc = 'connexion';
} elseif (empty($uc)) {
    $uc = 'accueil';
}
switch ($uc) {
case 'connexion':
    include 'controleurs/c_connexion.php';
    break;
case 'accueil':
    include 'controleurs/c_accueil.php';
    break;
case 'gererFrais':
    include 'controleurs/c_gererFrais.php';
    break;
case 'etatFrais':
    include 'controleurs/c_etatFrais.php';
    break;
case 'deconnexion':
    include 'controleurs/c_deconnexion.php';
    break;
}
require 'vues/v_pied.php';
```

Nous avons modifié la variable \$visiteur par \$utilisateur, afin de permettre de d'avoir deux type d'utilisateurs, les visiteurs et les comptables

```
switch ($action) {
case 'demandeConnexion':
    include 'vues/v_connexion.php';
    break;
case 'valideConnexion':
    $login = filter_input(INPUT_POST, 'login', FILTER_SANITIZE_STRING);
    $mdp = filter_input(INPUT_POST, 'mdp', FILTER_SANITIZE_STRING);
    $statut = getStatutUtilisateur($login); //on récupère ici le statut, visiteur ou comptable de l'utilisateur
    $utilisateur = $pdo->getInfosUtilisateur($login,$mdp,$statut); //on a remplacé ici la méthode getVisiteur par getUtilisateur pour mieux fit aux comptables
    if (!is_array($utilisateur)) { //qui ont un login normalisé avec un C à la fin de leur login
        ajouterErreur('Login ou mot de passe incorrect');
        include 'vues/v_erreurs.php';
        include 'vues/v_connexion.php';
    } else {
        $id = $utilisateur['id'];
        $nom = $utilisateur['nom'];
        $prenom = $utilisateur['prenom'];
        connecter($id, $nom, $prenom,$statut);
        header('Location: index.php');
    }
    break;
default:
    include 'vues/v_connexion.php';
    break;
}
```

Voici la méthode qui nous permet de différencier un visiteur d'un comptable

```
* Retourne le statut de l'utilisateur
*
* @param String $login Login de l'utilisateur
*
* @return String le statut de l'utilisateur
*/
function getStatutUtilisateur($login)
{
    if ($login[strlen($login)-1] == 'C') {
        return 'comptable';
    } else {
        return 'visiteur';
    }
}
```

Selon le statut que nous avons récupéré nous appelons la vue correspondante à l'utilisateur

```
<div class="col-md-8">
    <?php
    if($_SESSION['statut'] == 'comptable')
    {
        include 'v_menu_comptable.php';
    }
    else
    {
        include 'v_menu_visiteur.php';
    }
    ?>
</div>
```

Il fallait ensuite développer le contrôle validerFrais pour répondre au 3.

```
case 'validerFrais':
    $pdo->cloturerFicheFrais($mois);
    $lesVisiteurs =
    $_SESSION['lesVisiteurs'] = $pdo->getVisiteursATraiter('VA','CL');//Stocke dans une variable de session la liste des visiteurs ayant des fiches de frais à traiter
    require 'vues/v_choisirFichesOuvrir.php';
    require 'vues/v_choisirFichesChoixEtat.php';// on propose de choisir l'état de la fiche de frais qu'on veut traiter si le comptable se rend compte s'être trompé après validation
    require 'vues/v_choisirFichesListeVisiteurs.php';// on affiche la liste récupérée précédemment
    require 'vues/v_choisirFichesValider.php';
    break;
```

Après avoir sélectionné un visiteur à traiter, on vérifie s'il a des fiches à Valider

```
$lesVisiteurs =  
$_SESSION['lesVisiteurs'] = $pdo->getVisiteursATraiter('VA','CL');// on rappelle ici la fonction getVisiteursATraiter dans le cas où le visiteur précède  
$etat =  
$_SESSION['etat'] = filter_input(INPUT_POST, 'rbEtat', FILTER_SANITIZE_STRING);  
$idVisiteur =  
$_SESSION['leVisiteur'] = filter_input(INPUT_POST, 'lstVisiteurs', FILTER_SANITIZE_STRING);  
$lesMois =  
$_SESSION['lesMois'] = $pdo->getMoisATraiter($idVisiteur,$etat);  
/* Nous stockons lesVisiteurs, l'état, leVisiteur, lesMois dans des variables de session afin de ne plus appeler les fonctions qui les ont récupérés  
*/  
/* n'ayant plus besoin de les définir plus tard dans le programme*/  
if (isset($lesMois[0])) { // on vérifie ici que la ligne retournée par la fonction getMoisATraiter n'est pas vide  
    require 'vues/v_choisirFichesOuvrir.php';  
    require 'vues/v_choisirFichesChoixEtat.php';  
    require 'vues/v_choisirFichesListeVisiteurs.php';  
    require 'vues/v_choisirFichesListeMois.php'; // on affiche la liste de mois disponibles du visiteur  
    require 'vues/v_choisirFichesValider.php';  
} else {  
    if ($etat == 'VA') { // si elle est vide alors on affiche ces messages d'erreur 4.a  
        ajouterErreur('Pas de fiche de frais validée pour ce visiteur');  
    } else {  
        ajouterErreur('Pas de fiche de frais à valider pour ce visiteur');  
    }  
    require 'vues/v_erreurs.php';  
    require 'vues/v_choisirFichesOuvrir.php';  
    require 'vues/v_choisirFichesChoixEtat.php';  
    require 'vues/v_choisirFichesListeVisiteurs.php';  
    require 'vues/v_choisirFichesListeMois.php';  
    require 'vues/v_choisirFichesValider.php';  
}  
break;
```

Ensuite on cherche à voir les fiches de frais du mois sélectionné

```
case 'voirFicheFrais':  
    $leMois = // ayant récupéré le mois de la fiche de frais on le stock dans une variable de session  
    $_SESSION['leMois'] = filter_input(INPUT_POST, 'lstMois', FILTER_SANITIZE_STRING);  
    $lesFraisForfait = $pdo->getLesFraisForfait($idVisiteur, $leMois);  
    $lesFraisHorsForfait = $pdo->getLesFraisHorsForfait($idVisiteur, $leMois);  
    $nbJustificatifs = $pdo->getNbJustificatifs($idVisiteur,$leMois);//Nous récupérons toutes les données en rapport avec la fiche de frais à traiter et nous l'affichons  
    require 'vues/v_choisirFichesOuvrir.php';  
    require 'vues/v_choisirFichesChoixEtat.php';  
    require 'vues/v_choisirFichesListeVisiteurs.php';  
    require 'vues/v_choisirFichesListeMois.php';  
    require 'vues/v_choisirFichesValider.php';  
    require 'vues/v_correctionElementsForfaitises.php';  
    require 'vues/v_correctionElementsHorsForfaits.php';  
    break;
```

Une fois que les frais hors forfait sont corrigés, il faut bien cibler quel cas est celui invoqué

```
case 'corrigerFraisHorsForfait':
    $id = filter_input(INPUT_POST, 'id', FILTER_SANITIZE_STRING); //Tout d'abord nous récupérons la valeur du bouton pressé
    $reponse = SupprimerLigne($id); // nous récupérons les variables qui nous permettront de savoir si on doit supprimer la/les ligne/s sélectionnée/s
    $id = $reponse['id']; //la/les reporter au mois suivant ou la corriger
    if ($id == $idVisiteur) { // si l'id récupéré est celui du visiteur alors ça veut dire que les boutons pressés sont ceux en rapport avec les checkbox
        foreach ($_POST['chk'] as $ligne) {
            $lesLignes[$ligne]['montant'] = filter_input(INPUT_POST, $ligne.'montant', FILTER_SANITIZE_STRING);
            $lesLignes[$ligne]['date'] = filter_input(INPUT_POST, $ligne.'date', FILTER_SANITIZE_STRING);
            $lesLignes[$ligne]['libelle'] = filter_input(INPUT_POST, $ligne.'libelle', FILTER_SANITIZE_STRING);
            $lesLignes[$ligne]['id'] = $ligne;
            //On remplit ici un tableau associatif définissant les lignes de frais hors forfait sélectionnées
        }
        if ($reponse['isTrue']) { // Si le bouton pressé est celui de suppression
            $succes = 'Les lignes du : ' . $id . ' ont été supprimées.';
            foreach ($lesLignes as $ligne) { //pour toutes les lignes sélectionnées
                $montant = $ligne['montant'];
                $date = dateFrancaisVersAnglais($ligne['date']);
                $libelle = estLibelleValide('REFUSE : ' . $ligne['libelle']); //8 - 8.b on va appliquer REFUSE : devant le libellé et vérifier qu'il ne dépasse pas la taille
                $idFrais = $ligne['id']; //maximale autorisée
                $pdo->majFraisHorsForfait($date, $libelle, $montant, $idFrais); // et mettre à jour chaque lignes dans la base de donnée
                $succes = $succes . $ligne['date'] . ' ' . $idFrais . ' ' . $montant . ' ' . $libelle . ' ' . $idFrais . ' ' . $succes;
            }
            $succes = $succes . 'ont été supprimées.';
            ajouterSucces($succes); // on affiche que la suppression s'est bien déroulée
            include 'vues/v_succes.php';
        } else { // ici c'est le cas où l'on veut reporter la/les ligne/s sélectionnée/s 7.a
            $moisSuivant = setMoisSuivant($leMois); // on définit ici le mois suivant de la fiche de frais
            $ficheMoisSuivant = $pdo->getFicheFrais($idVisiteur, $moisSuivant); // on récupère cette fiche de frais
            if (isset($ficheMoisSuivant[0])) { // et si elle n'existe pas
                $pdo->creerNouvellesLignesFrais($idVisiteur, $moisSuivant); // on la crée
                $succes = 'Nouvelle fiche de frais créée.'; // et on stock un message de succes
                ajouterSucces($succes);
            }
            foreach ($lesLignes as $ligne) { //Quoi qu'il arrive toutes les lignes sélectionnées vont être déplacées
                $idFrais = $ligne['id'];
                $pdo->majFraisHorsForfait($moisSuivant, $idFrais); // en changeant simplement le mois de la fiche de hors forfait
            }
            $succes = 'Les lignes de frais hors forfait ont été reportées.'; //on ajoute un message de succes
            ajouterSucces($succes);
            include 'vues/v_succes.php'; // et on affiche les messages de succès stockés
        }
    }
}
```

```
} else {
    //ce cas est celui de la modification d'une ligne de frais de hors forfait
    $montant = filter_input(INPUT_POST, $id.'montant', FILTER_SANITIZE_STRING);
    $date = filter_input(INPUT_POST, $id.'date', FILTER_SANITIZE_STRING); //on récupère les nouvelles informations de la ligne
    $libelle = filter_input(INPUT_POST, $id.'libelle', FILTER_SANITIZE_STRING);
    valideInfosFrais($date, $libelle, $montant); //on vérifie leurs validité
    $date = dateFrancaisVersAnglais($date);
    if (nbErreurs() != 0) {
        include 'vues/v_erreurs.php'; // si elles ne sont pas valides on va alors afficher un message d'erreur
    } else {
        $pdo->majFraisHorsForfait($date, $libelle, $montant, $id); //sinon on va mettre à jour la base de donnée
        $succes = 'Les lignes de frais hors forfait ont été mis à jour.';
        ajouterSucces($succes); //et afficher un message de succès
        include 'vues/v_succes.php';
    }
}
break;
```

Lorsque les Frais forfaitisés sont modifiés on appelle une méthode pour mettre à jour la BDD

```
case 'validerMajFraisForfait':
    $lesFrais = filter_input(INPUT_POST, 'lesFrais', FILTER_DEFAULT, FILTER_FORCE_ARRAY);
    if (lesQteFraisValides($lesFrais)) {
        if ($statut != 'comptable') {
            $pdo->majFraisForfait($idVisiteur, $mois, $lesFrais);
        } else {
            $pdo->majFraisForfait($idVisiteur, $leMois, $lesFrais); // nous avons ajouté au code source de base une modification permettant de gérer la mise
            ajouterSucces('Les frais forfaitisés ont bien été mis à jour.');// à jour des frais forfaitisés par le comptable
            include 'vues/v_succes.php';
        }
    } else {
        ajouterErreur('Les valeurs des frais doivent être numériques');
        include 'vues/v_erreurs.php';
    }
}
break;
```

Une fois que tout est mis à jour et modifié, on peut valider la fiche de frais qui va passer à l'état VA

```
case 'validerFicheFrais' :  
    $lesFraisForfait = $pdo->getLesFraisForfait($idVisiteur, $leMois);//pour cela nous devons récupérer les frais forfaitisés  
    $lesFraisHorsForfait = $pdo->getLesFraisHorsForfait($idVisiteur, $leMois);// et les hors forfait  
    $montantFrais = $pdo->getLesMontantFrais();//récupérer les coefficients de paiement des frais forfaitisés  
    //si un frais est refusé c'est dans la méthode ci-dessous que l'exception est notée  
    $montant = getMontantValide($idVisiteur,$leMois,$lesFraisForfait,$lesFraisHorsForfait,$montantFrais);//et calculer ainsi le montant total des frais à rembourser  
    $pdo->majFicheFrais($idVisiteur,$leMois,$montant,'VA');//puis on met à jour la fiche dans la base de donnée et on passe la fiche à l'état Validé en attente de Remboursement  
    ajouterSucces('La fiche de frais a été validée.');//on affiche ainsi un message de succès  
    include 'vues/v_succes.php';  
    require 'vues/v_choisirFichesOuvrir.php';  
    require 'vues/v_choisirFichesChoixEtat.php';  
    require 'vues/v_choisirFichesListeVisiteurs.php';  
    require 'vues/v_choisirFichesValider.php';  
    break;
```



## Tâche 2 : Suivi du paiement des fiches de frais

PROJET : Application web de gestion des frais	Description cas d'utilisation
---	-------------------------------

<b>Nom cas d'utilisation :</b> Suivre le paiement fiche de frais
<b>Acteur déclencheur :</b> Comptable
<b>Pré conditions :</b> Utilisateur Comptable authentifié
<b>Post conditions :</b> Néant
<b>Scénario nominal :</b>  <ol style="list-style-type: none"><li>1. L'utilisateur demande à suivre le paiement les fiches de frais.</li><li>2. Le système propose de choisir une fiche de frais parmi celles à valider et mises en paiement</li><li>3. L'utilisateur sélectionne les informations et valide</li><li>4. Le système affiche le détail de la fiche de frais –frais forfaitisés et hors forfait</li><li>5. L'utilisateur « Met en paiement » la fiche de frais</li><li>6. Le système modifie l'état de la fiche à « Mise en paiement » et met à jour la date de modification.</li></ol>
<b>Exceptions :</b> <ul style="list-style-type: none"><li>• 4-a : Aucune fiche de frais n'existe, le système affiche 'Pas de fiche de frais pour ce visiteur ce mois'. Retour au 2</li><li>• 5.a : L'utilisateur indique que la fiche a été effectivement payée</li><li>• 6.a : Le système modifie l'état de la fiche à « Remboursée » et enregistre la date de modification</li></ul>
<b>Contraintes :</b>
<b>Questions ouvertes :</b>

Tout d'abord il fallait utiliser la même méthode c'est-à-dire sélectionner un visiteur et le mois de la fiche de frais à valider

```
case 'suiviPalementFrais':
    $lesVisiteurs = // on stocke ici les visiteurs dans une variable de session
    $_SESSION['lesVisiteurs'] = $pdo->getVisiteursATraiter('VA','VA');// nous définissons les filtres des états des fiches des visiteurs à récupérer uniquement à 'Validé'
    if (isset($lesVisiteurs[0])) {
        require 'vues/v_choisirFichesOuvrir.php'; //on récupère les visiteurs selon l'état de leur fiche
        require 'vues/v_choisirFichesListeVisiteurs.php';
        require 'vues/v_choisirFichesValider.php';
    } else {
        ajouterSucces('Plus aucune fiche de visiteur à traiter');
        require 'vues/v_choisirFichesOuvrir.php'; //si la ligne ne retourne aucune valeur
        require 'vues/v_succes.php'; //alors tous les visiteurs ont été remboursés
        require 'vues/v_choisirFichesValider.php';
    }
    break;
```

Nous choisissons donc la fiche de frais à rembourser selon le mois

```
case 'choixFiche':
    $lesVisiteurs =
    $_SESSION['lesVisiteurs'] = $pdo->getVisiteursATraiter('VA','VA');
    $idVisiteur =
    $_SESSION['leVisiteur'] = filter_input(INPUT_POST, 'lstVisiteurs', FILTER_SANITIZE_STRING);
    $leMois =
    $_SESSION['leMois'] = $pdo->getMoisATraiter($idVisiteur, 'VA');//on récupère les mois
    /* Nous stockons lesVisiteurs, leVisiteur, lesMois dans des variables de session afin de ne plus appeler les fonctions qui les ont récupérés
    * n'ayant plus besoin de les définir plus tard dans le programme*/
    if (isset($leMois[0])) { // si la ligne est remplie
        require 'vues/v_choisirFichesOuvrir.php';
        require 'vues/v_choisirFichesListeVisiteurs.php'; //on affiche les fiches de frais à valider disponibles
        require 'vues/v_choisirFichesListeMois.php'; //sous forme de mois à traiter
        require 'vues/v_choisirFichesValider.php';
    } else { // si la ligne est vide
        ajouterErreur('Pas de fiche de frais validée pour ce visiteur');//cela veut dire que ce visiteur n'a pas de fiche de frais à remplir
        require 'vues/v_erreurs.php';
        require 'vues/v_choisirFichesOuvrir.php';
        require 'vues/v_choisirFichesListeVisiteurs.php';
        require 'vues/v_choisirFichesValider.php';
    }
    break;
```

```
case 'voirFicheFrais':
    $leMois =
    $_SESSION['leMois'] = filter_input(INPUT_POST, 'lstMois', FILTER_SANITIZE_STRING);
    $lesFraisForfait = $pdo->getLesFraisForfait($idVisiteur, $leMois);
    $lesFraisHorsForfait = $pdo->getLesFraisHorsForfait($idVisiteur, $leMois);
    $lesInfosFicheFrais = $pdo->getLesInfosFicheFrais($idVisiteur, $leMois);
    $numAnnee = substr($leMois, 0, 4);
    $numMois = substr($leMois, 4, 2);
    $libEtat = $lesInfosFicheFrais['libEtat'];
    $montantAValider = $lesInfosFicheFrais['montantValide'];
    $nbJustificatifs = $lesInfosFicheFrais['nbJustificatifs'];
    $dateModif = dateAnglaisVersFrancais($lesInfosFicheFrais['dateModif']);
    //on récupère ci-dessus toutes les variables nécessaires au remplissage de la fiche de frais
    require 'vues/v_choisirFichesOuvrir.php';
    require 'vues/v_choisirFichesListeVisiteurs.php'; //on affiche ainsi la fiche de frais
    require 'vues/v_choisirFichesListeMois.php'; //avec pour seule variable modifiable
    require 'vues/v_choisirFichesValider.php'; //le taux de remboursement du visiteur
    require 'vues/v_etatFrais_comptable.php';
    break;
    /*Dans ce cas nous allons changer l'état de la fiche de frais à 'Remboursée'*/
```

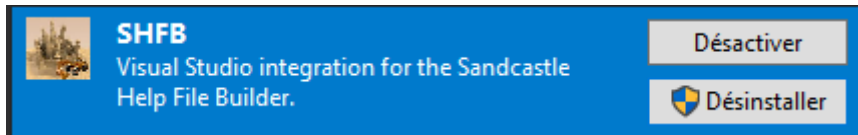
Une fois que toutes les informations sont justes, on procède au remboursement de la fiche de frais en prenant en compte d'éventuelles modifications du montant total à rembourser

```
case 'rembourserFiche':
    $montant = filter_input(INPUT_POST, 'montant', FILTER_SANITIZE_STRING);//on récupère ici le montant, modifié ou non par le comptable, de la fiche de frais
    if ($montant == '') { //nous vérifions que la section n'est pas vide
        ajouterErreur('Le champ montant ne peut pas être vide');
    } elseif (!is_numeric($montant)) { // et que la valeur est bien numérique
        ajouterErreur('Le champ montant doit être numérique');
    }
    if (nbErreurs() > 0)
    {
        require 'vues/v_erreurs.php'; // si ce n'est pas le cas on affiche une erreur
    } else {
        $pdo->majFicheFrais($idVisiteur, $leMois, $montant, 'RB'); //si tout s'est bien déroulé
        ajouterSucces('La fiche a été remboursée.');//nous mettons à jour la base de donnée
        require 'vues/v_succes.php'; //et affichons un message de succès du remboursement
    }
    $lesVisiteurs =
    $_SESSION['lesVisiteurs'] = $pdo->getVisiteursATraiter('VA','VA');
    require 'vues/v_choisirFichesOuvrir.php'; //On actualise la liste de visiteurs dans le cas où le visiteur actuel ait été entièrement traité
    require 'vues/v_choisirFichesListeVisiteurs.php'; //On affiche ainsi la liste de visiteurs
    require 'vues/v_choisirFichesValider.php'; //afin de sélectionner une autre fiche à traiter
    break;
```

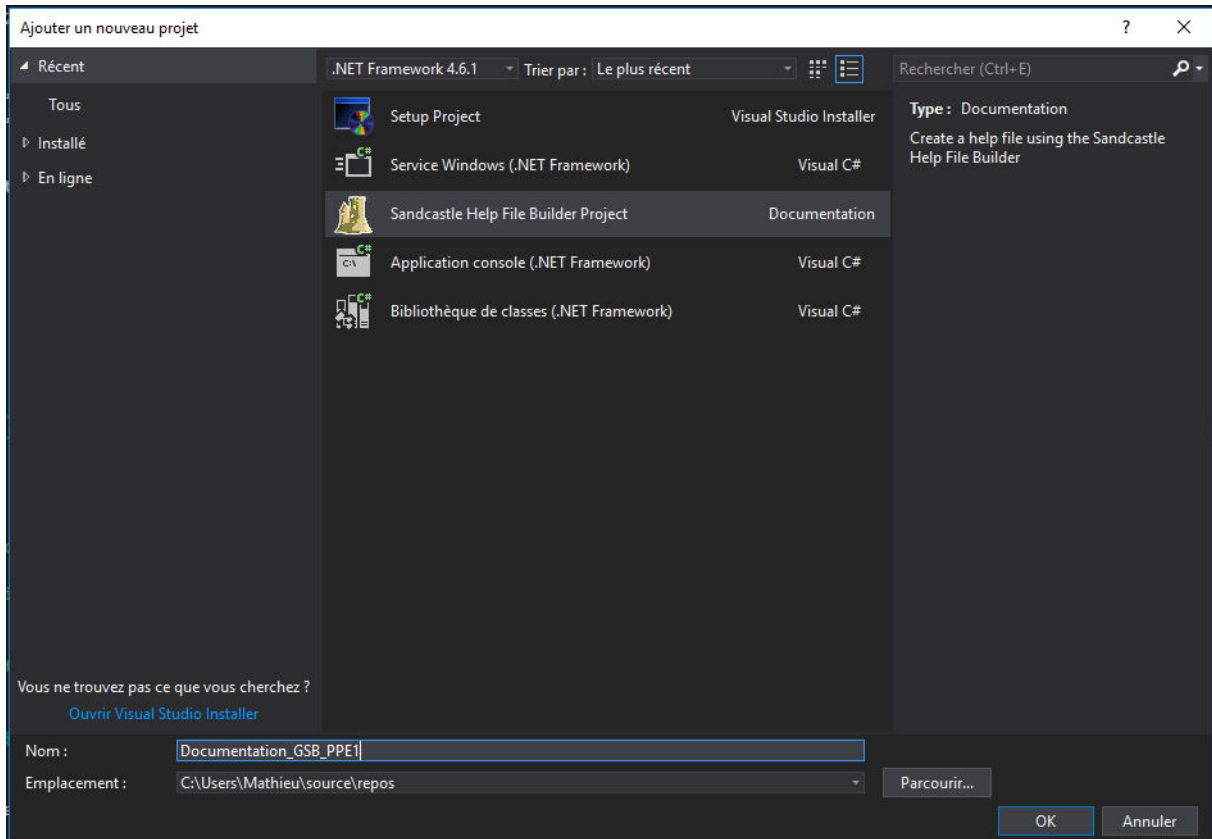


## Tâche 3 : Production de la documentation

Pour produire la documentation sur Visual studio, nous avons choisi la solution SHFB qui nous permet de générer la documentation des méthodes utilisées dans l'application PHP



Puis nous générons la solution Sandcastle afin d'avoir accès à cette documentation



## Tâche 4 : Gestion du refus de certains frais hors forfait

Pour répondre à cette tâche il fallait simplement appliquer un libelle « REFUSE : » devant les lignes que nous avons considéré comme non remboursable et ainsi garder une certaine traçabilité

```
if ($reponse['isTrue']) { // Si le bouton pressé est celui de suppression
    $succes = 'Les lignes du : ';//alors
    foreach ($leslignes as $ligne){ //pour toutes les lignes sélectionnées
        $montant = $ligne['montant'];
        $date = dateFrancaisVersAnglais($ligne['date']);
        $libelle = estLibelleValide('REFUSE :'.$ligne['libelle']); //8 - 8.b on va appliquer REFUSE : devant le libellé et vérifier qu'il ne dépasse pas la taille
        $idFrais = $ligne['id']; //maximale autorisée
        $pdo->majFraisHorsForfait($date,$libelle,$montant,$idFrais); // et mettre à jour chaque lignes dans la base de donnée
        $succes = $succes . $ligne['date'] . ' ';
    }
    $succes = $succes . 'ont été supprimées.';
    ajouterSucces($succes); // on affiche que la suppression s'est bien déroulée
    include 'vues/v_succes.php';
}
```