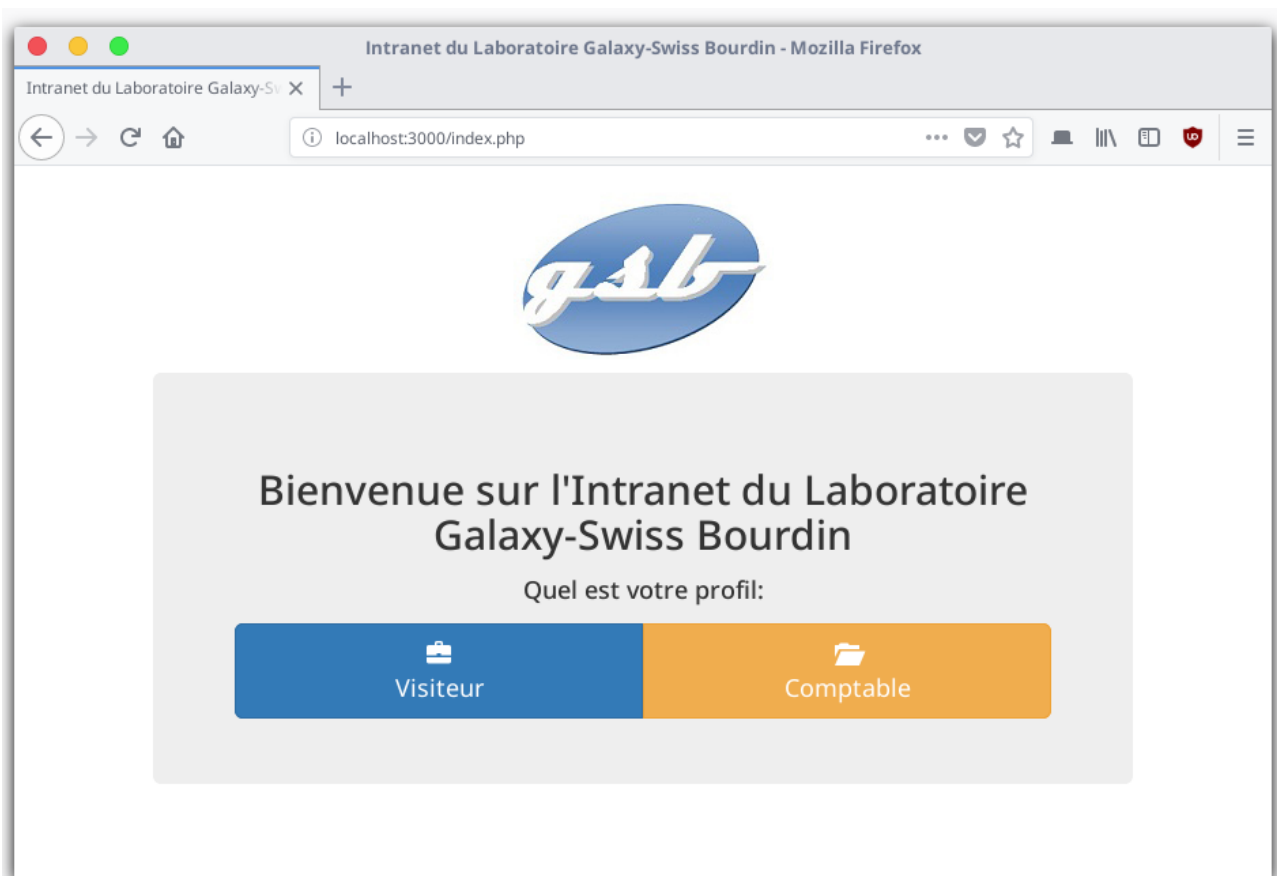


# Compte rendu de développement

## Gestion des frais de déplacements des visiteurs médicaux du laboratoire pharmaceutique GSB

*Développement de la partie comptable de l'application web de gestion des frais de déplacement des visiteurs médicaux*



## Index

Tâche préparatoire : Authentification des comptables.....	3
Tâche 1 : Valisation d'une fiche de frais.....	5
Tâche 2 : Suivi du paiement des fiches de frais.....	6
Tâche 3 : Production de la documentation.....	7
Tâche 4 : Gestion du refus de certains frais hors forfait.....	9
Tâche 5 : Sécurisation des mots de passe stockés.....	10
Tâche 6 : Gestion plus fine de l'indemnisation kilométrique.....	11
Tâche 7 : Génération d'un état de frais au format PDF.....	13
Conclusion.....	14

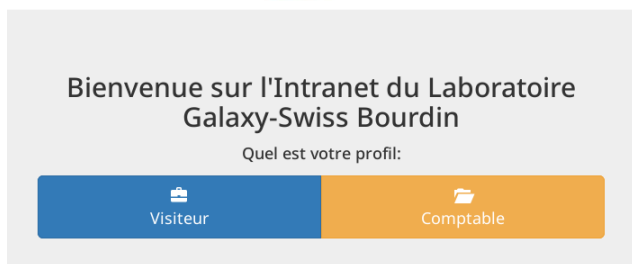
## Tâche préparatoire : Authentification des comptables

Afin de pouvoir réaliser les tâches suivantes, une authentification des utilisateurs de type comptable doit être réalisée. Pour se faire une nouvelle table « comptable » est créée .

```
CREATE TABLE IF NOT EXISTS `comptable`  
(  
  `id`          CHAR(4) NOT NULL,  
  `nom`         CHAR(30) DEFAULT NULL,  
  `prenom`     CHAR(30) DEFAULT NULL,  
  `login`      CHAR(20) DEFAULT NULL,  
  `mdp`        VARCHAR(255) DEFAULT NULL,  
  `adresse`    CHAR(30) DEFAULT NULL,  
  `cp`         CHAR(5) DEFAULT NULL,  
  `ville`      CHAR(30) DEFAULT NULL,  
  `dateembauche` DATE DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)
```



Sur l'accueil de l'application un bouton est rajouté pour que l'utilisateur puisse s'identifier en tant que visiteur ou comptable.



Pour identifier les comptables, le même système que pour les visiteurs est utilisé mais la méthode *getInfosUtilisateur()* de *class.pdogs.inc.php* est modifiée pour prendre un troisième argument *\$type* qui peut être « visiteur » ou « comptable » et qui va permettre de choisir sur quelle table la requête sera effectuée.

Le routage qui se trouve dans *index.php* est alors modifié pour réserver des routes aux visiteurs ou aux comptables.

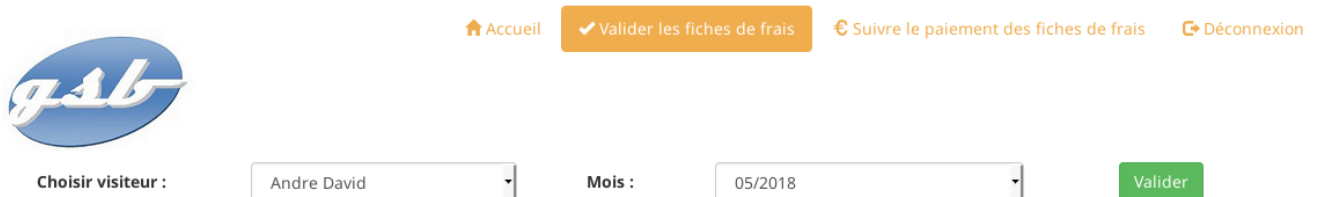
```

// Routes communes à tous les utilisateurs connectés
if ($estConnecte) {
    switch ($uc) {
        case 'connexion':
            include 'controleurs/c_connexion.php';
            break;
        case 'accueil':
            include 'controleurs/c_accueil.php';
            break;
        case 'deconnexion':
            include 'controleurs/c_deconnexion.php';
            break;
    }
}
// Routes réservées aux visiteurs
if ($estVisiteur) {
    switch ($uc) {
        case 'gererFrais':
            include 'controleurs/c_gererFrais.php';
            break;
        [...]
    }
}
// Routes réservées aux comptables
if ($estComptable) {
    switch ($uc) {
        case 'validerfichefrais':
            include 'controleurs/c_validerFicheFrais.php';
            break;
        [...]
    }
}

```

## Tâche 1 : Valisation d'une fiche de frais

L'interface est réalisée en accord avec les images fournies :



The screenshot shows the GSB logo on the left. To the right, there are navigation links: 'Accueil', 'Valider les fiches de frais' (highlighted in orange), 'Suivre le paiement des fiches de frais', and 'Déconnexion'. Below the logo, there is a form with two dropdown menus: 'Choisir visiteur : Andre David' and 'Mois : 05/2018'. A green 'Valider' button is positioned to the right of the second dropdown.

### Valider la fiche des frais

#### Éléments forfaitisés

Forfait Etape

Nuitée Hôtel

Repas Restaurant

Le comptable peut sélectionner un visiteur et un mois pour accéder à la fiche de frais et y effectuer des modifications, mais seulement si son état est « Saisie clôturée », le cas échéant, une erreur est affichée.


Choisir visiteur : Andre David Mois : 04/2018 Valider

Fiche déjà remboursée

Les modifications de la fiche se font via de simple requêtes SQL, et la validation de la fiche se fait via la méthode `majEtatFicheFrais()` de la classe PDO (`class.pdogsb.inc.php`).

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

Selon le même principe que pour la validation des fiches de frais, il est possible de choisir une fiche via la sélection d'un utilisateur et d'un mois.



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

Choisir visiteur : Andre David Mois : 04/2018 Valider

### Valider la fiche des frais

Fiche de frais du mois 04-2018 :

**Etat :** Remboursée depuis le 12/05/2018

Éléments forfaitisés

Forfait Etape	Nuitée Hôtel	Repas Restaurant	Frais Kilométriques
10	0	0	340,34

Descriptif des éléments hors forfait - 0 justificatifs reçus

Date	Libellé	Montant	
01/04/2018	kfc	12.00	<a href="#">Refuser ce frais</a>

Selon l'état de la fiche il est possible de la passer en « en paiement » ou « remboursement émis ».

Changer état :

€ En paiement

Changer état :

↻ Remboursement émis

Il s'agit principalement d'afficher les données inscrites dans la base de données et d'offrir la possibilité de changer l'état de la fiche en utilisant encore une fois la méthode `majEtatFicheFrais()` de la classe PDO (`class.pdogs.inc.php`).

## Tâche 3 : Production de la documentation

La documentation technique est produite grâce à phpDocumentor. Les commentaires étant déjà formatés seules quelques corrections ont été faites dans le code.

Voici un exemple de ce que l'on peut trouver dans la documentation d'une méthode de la classe PdoGsb.

✓ majNbJustificatifs()...

```
majNbJustificatifs(String $idVisiteur, String $mois, Integer $nbJustificatifs) : null
```

</>

*Met à jour le nombre de justificatifs de la table ficheFrais pour le mois et le visiteur concerné*

### Parameters

String	\$idVisiteur	ID du visiteur
String	\$mois	Mois sous la forme aaaamm
Integer	\$nbJustificatifs	Nombre de justificatifs

### Returns

null

La documentation est accessible sur la page du PPE et peut être générée en exécutant phpDocumentor sur le dossier de l'application.

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

Le comptable peut refuser un frais, celui-ci sera alors reporté au mois suivant et « REFUSÉ » sera rajouté devant son libelle.

Descriptif des éléments hors forfait - 0 justificatifs reçus			
Date	Libellé	Montant	
01/04/2018	kfc	12.00	<a href="#">Refuser ce frais</a>

La méthode `refuserFraisHorsForfait()` est créée pour cela :

```
/**
 * Refuse un frais hors forfait en rajoutant "REFUSÉ" devant son libelle
 * et en le reportant au mois suivant.
 *
 * @param String $idVisiteur ID du visiteur
 * @param String $mois      Mois sous la forme aaaamm
 * @param Integer $idFrais  l'ID du frais
 *
 * @return null
 */
public function refuserFraisHorsForfait(
    $idVisiteur,
    $mois,
    $idFrais
) {
    $unMois = $mois+1;
    if (!$this->ficheDeFraisExiste($idVisiteur, $unMois)) {
        $this->creerNouvellesLignesFrais($idVisiteur, $unMois);
    }
    $requetePrepare = PdoGSB::$monPdo->prepare(
        'UPDATE lignefraishorsforfait '
        . 'SET libelle = CONCAT("REFUSÉ ", libelle), '
        . 'mois = :unMois '
        . 'WHERE lignefraishorsforfait.id = :unId '
    );
    $requetePrepare->bindParam(':unId', $idFrais, PDO::PARAM_INT);
    $requetePrepare->bindParam(':unMois', $unMois, PDO::PARAM_STR);
    $requetePrepare->execute();
}
```

Si la fiche de frais du mois suivant n'existe pas, elle est créée grâce à la méthode `creerNouvellesLignesFrais()`.

Cela donne pour le visiteur :

Descriptif des éléments hors forfait			
Date	Libellé	Montant	
01/04/2018	REFUSÉ kfc	12.00	<a href="#">Supprimer ce frais</a>

Il peut alors laisser le frais hors forfait qui sera examiné le mois suivant, ou le supprimer.



## Tâche 5 : Sécurisation des mots de passe stockés

La sécurisation des mots de passe est très important et cela fait même partie des exigences de la CNIL (voir <https://www.cnil.fr/fr/authentication-par-mot-de-passe-les-mesures-de-securite-elementaires> trouvable en cherchant “cnil mots de passes” sur Google).

Jusque la les mots de passes sont stockés en clair et l’authentification s’effectue de la sorte :

```
public function getInfosUtilisateur($type, $login, $mdp)
{
    $requetePrepare = PdoGsb::$monPdo->prepare(
        'SELECT visiteur.id AS id, visiteur.nom AS nom, '
        . 'visiteur.prenom AS prenom '
        . 'FROM visiteur '
        . 'WHERE visiteur.login = :unLogin AND visiteur.mdp = :unMdp'
    );
    $requetePrepare->bindParam(':unLogin', $login, PDO::PARAM_STR);
    $requetePrepare->bindParam(':unMdp', $mdp, PDO::PARAM_STR);
    $requetePrepare->execute();
    return $requetePrepare->fetch();
}
```

La méthode choisie est d’utiliser la fonction password\_hash() de php qui permet de créer un hash du mot de passe en utilisant un algorithme prédéfini. Par défaut depuis PHP 5.5.0, c’est bcrypt qui est utilisé, mais cela peut changer si par exemple une faille est découverte.

La fonction password\_verify() est alors utilisée pour vérifier les mots de passe et la méthode getInfosUtilisateur() est modifiée pour d’abord récupérer dans la requête un utilisateur dont le login correspond, puis vérifier son mot de passe.

```
public function getInfosUtilisateur($type, $login, $mdp)
{
    $requetePrepare = PdoGsb::$monPdo->prepare(
        'SELECT visiteur.id AS id, visiteur.nom AS nom, '
        . 'visiteur.prenom AS prenom, '
        . 'visiteur.mdp AS mdp '
        . 'FROM visiteur '
        . 'WHERE visiteur.login = :unLogin'
    );
    $requetePrepare->bindParam(':unLogin', $login, PDO::PARAM_STR);
    $requetePrepare->execute();
    $utilisateur = $requetePrepare->fetch();
    if (password_verify($mdp, $utilisateur['mdp'])) {
        // Pas besoin de garder le hash du mdp
        unset($utilisateur['mdp']);
        return $utilisateur;
    } else {
        return false;
    }
}
```

## Tâche 6 : Gestion plus fine de l'indemnisation kilométrique

Pour gérer de façon plus fine l'indemnisation kilométrique, les kilomètres parcourus seront maintenant entrés en fonction du véhicule utilisé.

Chaque visiteur peut avoir un ou plusieurs véhicules, et un véhicule peut être attribué à plusieurs visiteurs (véhicule de société par exemple), qu'il devra choisir au moment de rentrer le nombre de kilomètres parcourus.

### Nouveau frais kilométrique

Véhicule

Distance

Ajouter

Effacer

Ces frais seront affichés individuellement sur la fiche et calculés en fonction de la puissance fiscale et du carburant du véhicule:

Frais Kilométrique		
Véhicule	Distance	Montant
AA-123-AA	154	89.32
ZZ-999-AA	987	611.94

Les caractéristiques des différentes puissances fiscales se trouvent dans la base de données :

```
--  
-- Table structure for table `vehicule`  
--  
DROP TABLE IF EXISTS `vehicule`;CREATE TABLE IF NOT EXISTS `vehicule` (  
  `immatriculation` varchar(20) NOT NULL,  
  `idvisiteur` char(4) NOT NULL,  
  `idpuissance` char(4) NOT NULL,  
  PRIMARY KEY (`immatriculation`),  
  KEY `idvisiteur` (`idvisiteur`,`immatriculation`)  
) ENGINE=InnoDB AUTO_INCREMENT=97 DEFAULT CHARSET=latin1;
```

```

--
-- Dumping data for table `vehicule`
--
INSERT INTO `vehicule` (`immatriculation`, `idvisiteur`, `idpuissance`) VALUES
('AA-123-AA', 'a17', '5D'),
('ZZ-999-AA', 'a17', '4e');

--
-- Table structure for table `puissancevehicule`
--DROP TABLE IF EXISTS `puissancevehicule`;CREATE TABLE IF NOT EXISTS `puissance
vehicule` (
  `id` char(3) NOT NULL,
  `puissance` varchar(20) DEFAULT NULL,
  `bareme` decimal(5,2) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `fraisforfait`
--INSERT INTO `puissancevehicule` (`id`, `puissance`, `bareme`) VALUES
('4D', '4CV Diesel', '0.52'),
('5D', '5/6CV Diesel', '0.58'),
('4E', '4CV Essence', '0.62'),
('5E', '5/6CV Essence', '0.67'),
('GPL', 'GPL', '0.49');

```

## Tâche 7 : Génération d'un état de frais au format PDF

Comme conseillé, la classe FPDF est utilisée pour générer l'état de frais. La création d'un gabarit est assez laborieuse puisqu'il faut placer une à une des cellules de texte sur une sorte de canevas.

Heureusement la documentation et les exemples sont très clairs et il est au final assez facile (mais long) de recréer l'exemple fourni dynamiquement.



Télécharger PDF

Pour plus de simplicité, un fichier pdf.php est créé dans la racine de l'application et va générer dynamiquement un pdf et déclencher un téléchargement chez l'utilisateur.

Les méthodes de la classe fpdf permettent de rajouter du contenu à une instance (une page) de la classe, voici par exemple comment sont affichés les frais hors forfait.

```
$largeurPage = "210";
$tiersPage = (($largeurPage-20)/3); // Le 20 corespond aux marges

foreach ($lesFraisHorsForfait as $unFraisHorsForfait) {
    $date = $unFraisHorsForfait['date'];
    $libelle = htmlspecialchars($unFraisHorsForfait['libelle']);
    $montant = $unFraisHorsForfait['montant'];
    // Addition au total
    $totalFiche += number_format($montant, 2, '.', '');
    // Affichage
    $pdf->Cell($tiersPage, 12, $date, '1', 0, 'L');
    $pdf->Cell($tiersPage, 12, $libelle, '1', 0, 'L');
    $pdf->Cell($tiersPage, 12, $montant, '1', 1, 'R');
}
```

# Conclusion

L'application web fonctionne en accord avec ce qui est demandé dans les missions mais elle n'est pas complètement terminée. Bien qu'elle soit fonctionnelle et utilisable tel quel des précisions seraient nécessaires pour affiner certaines fonctions, par exemple la possibilité d'annuler ou non la validation ou le remboursement d'une fiche ou bien la gestion des justificatifs.

Au niveau de l'interface des retours d'utilisateurs seraient d'une très grande aide afin d'optimiser l'expérience utilisateur et de faciliter l'exécution de tâches répétitives.

Il y a aussi tout un aspect de l'application qui n'est pas implémenté : l'administration. Pour l'instant les tâches administratives : création et suppression d'utilisateurs, attributions de véhicules, etc. sont directement effectuées sur la base de données. Et nécessitent donc d'être administrées par une personne qualifiée, car les erreurs sont possibles et aucune protection n'est mise en place.