04/09/2014

Déborah Lurienne

groupe école pratique - Marseille

Documentation technique

Proj Application web de gestion des frais



SOMMAIRE

[1 Arborescences des modules 2](#_Toc398901862)

[1.1 Module 1 - Enregistrement et suivi par les visiteurs : 2](#_Toc398901863)

[1.2 Module 2 – Enregistrement des opérations par les comptables 2](#_Toc398901864)

[2 Documentation technique 3](#_Toc398901865)

[2.1 Contexte de la situation professionnelle 3](#_Toc398901866)

[2.2 Qu’est- ce que le MVC ? 3](#_Toc398901867)

[2.3 Environnement de développement 6](#_Toc398901868)

[2.4 Architecture de la base de données : 7](#_Toc398901869)

[2.4.1 Création de la base 8](#_Toc398901870)

[2.4.2 Qu’est-ce que c’est la PDO ? 9](#_Toc398901871)

[2.5 Détail des pages 10](#_Toc398901872)

[2.5.1 Espace d’authentification 10](#_Toc398901873)

[2.5.2 Validation des frais 13](#_Toc398901874)

[2.5.3 Consultation des frais 14](#_Toc398901875)

[2.5.4 Déclaration des frais 15](#_Toc398901876)

[2.5.5 Déconnexion 16](#_Toc398901877)

[2.6 Bibliothèques externes utilisées 16](#_Toc398901878)

[3 Proposition d’amélioration 17](#_Toc398901879)

[4 Conclusion 17](#_Toc398901880)

# 1 Arborescences des modules

## Module 1 - Enregistrement et suivi par les visiteurs :

Ce module se représente de cette manière :

* Déclaration des frais
* Consultation des frais
* Déconnexion

## Module 2 – Enregistrement des opérations par les comptables

Ce module se représente de cette manière :

* Validation des frais
* Consultation des frais
* Déconnexion

*Déconnexion*

*Consultation des frais*

*Validation des frais*

*Consultation des frais*

*Déconnexion*

*Déclaration des frais*

*Accueil*

*Comptable*

*Visiteur*

# Documentation technique

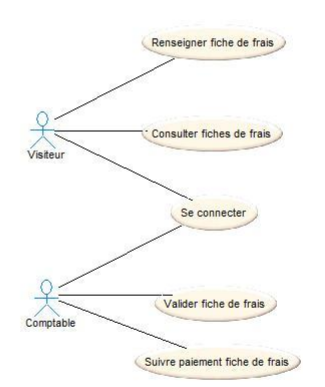
## Contexte de la situation professionnelle

Actuellement l’application de GSB ne permet qu’au visiteur de s’y connecter.

Une demande a été faite : celle de rajoutée un nouveau module consacré pour les comptable, ce qu’ils leur permettraient de s’authentifier et d’avoir leur propre espace.

L’application de GSB devra donc constituer à présent de deux modules :

* **Le module des visiteurs**
* **Le module des comptables**

****

Qui sont fondée sur une architecture dite « **MVC** »

## Qu’est- ce que le MVC ?

Tout d’abord le MVC est l’acronyme de « **M**odel **V**iew **C**ontroller ». En français « **M**odel **V**ue **C**ontrôleur », l’un des plus célèbres design patterns.

C’est une manière de travailler pour rendre l’application facilement maintenable, offre un code souple et structurer ce qui permet d’accélérer considérablement les développements futurs.

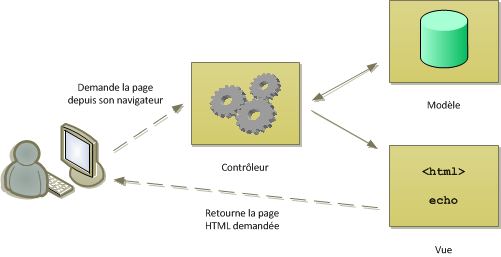
Mais en quoi cela consiste véritablement ?

Le but du MVC est justement de séparer la logique du code en trois parties :

* **Modèle** : cette partie gère les *données* de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc les requêtes SQL.
* **Vue** : cette partie se concentre sur l'*affichage*. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple la liste des messages des forums.
* **Contrôleur** : cette partie gère la logique du code qui prend des *décisions*. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).

La figure suivante schématise le rôle de chacun de ces éléments



Il faut tout d’abord retenir que le contrôleur est le chef d’orchestre, c’est lui qui reçoit la requête du visiteur et qui contacte d’autre fichiers mais pas seulement, il y a des calculs ou des vérifications d'autorisations à faire, il doit contacter le Modèle s’il a besoin de donnée en base, des images à miniaturiser, c'est lui qui s'en chargera et une fois tous ces calcules finis, il donne tout à la vue. De dette manière le code est bien rangé et chaque partie est facile à réutiliser.

La requête du client arrive au contrôleur et celui-ci lui retourne la vue.

Pour reprendre le code de l’application:

Dans l’index « index.php » (Le premier fichier exécutait)

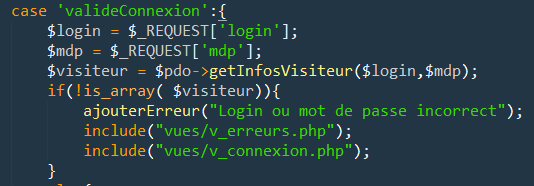


*On inclut le contrôleur*

*On appel une fonction du model (appelé plus haut dans l’index) pour récupérer des informations en base de données*

*On inclut le model*

Dans le contrôleur «constroleurs/c\_connexion.php »

**

*On inclut les vues*

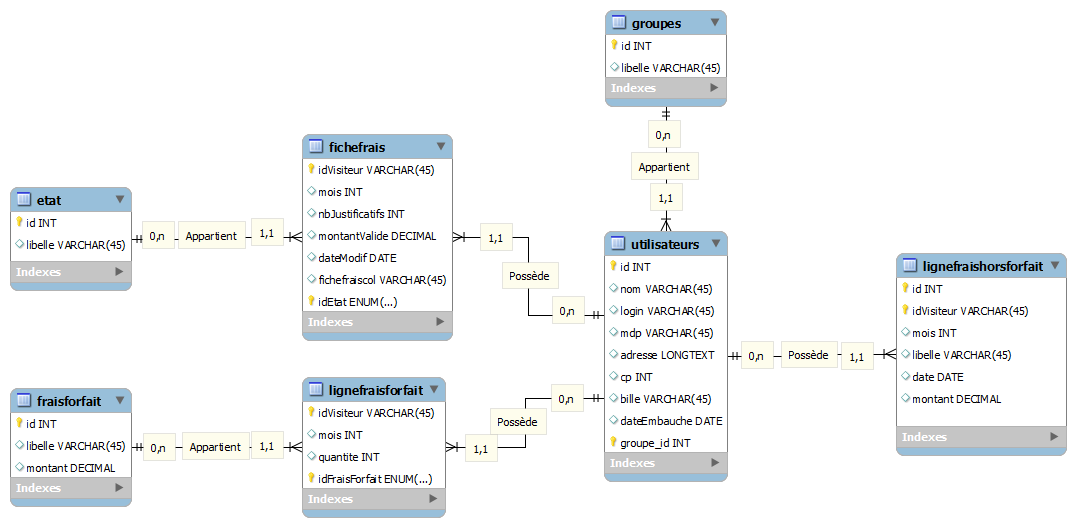
## Environnement de développement

***L’IDE*** *: Sublime Text*

***Langages*** *: HTML/CSS, PHP, JQuery, SQL*

***Nature*** *: Application web*

## Architecture de la base de données :

Le schéma MCD (Modèle conceptuel de données) :

1

3

2

Pour pouvoir créer deux modules séparait, il a fallu revoir un peu la base de données. Quelques modifications ont donc été apportées.

1. Une table groupes a été rajoutée dans la base de données : De cette manière il sera facile de rajouter de nouveau groupe par la suite, tout se passera ici.
2. La table visiteur s’est faite renommée en « utilisateurs » pour lui donner un nom plus général.
3. Un champ par clé étrangère a été rajouté dans la table utilisateur pour relier cette à la table groupes.

### Création de la base

CREATE TABLE IF NOT EXISTS `groupes` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`libelle` varchar(255) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=4 ;

CREATE TABLE IF NOT EXISTS `utilisateurs` (

`id` char(4) NOT NULL,

`nom` char(30) DEFAULT NULL,

`prenom` char(30) DEFAULT NULL,

`login` char(20) DEFAULT NULL,

`mdp` char(20) DEFAULT NULL,

`adresse` char(30) DEFAULT NULL,

`cp` char(5) DEFAULT NULL,

`ville` char(30) DEFAULT NULL,

`dateEmbauche` date DEFAULT NULL,

`groupe\_id` int(11) NOT NULL DEFAULT '1',

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

ALTER TABLE `fichefrais`

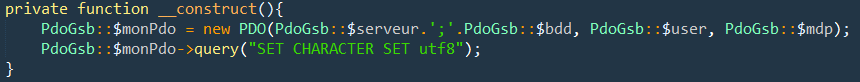
ADD CONSTRAINT `fichefrais\_ibfk\_1` FOREIGN KEY (`idEtat`) REFERENCES `etat` (`id`),

ADD CONSTRAINT `fichefrais\_ibfk\_2` FOREIGN KEY (`idVisiteur`) REFERENCES `utilisateurs` (`id`);

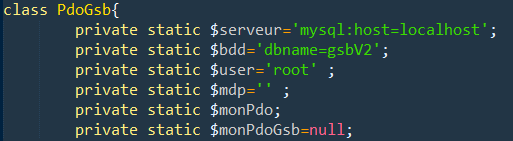
### Qu’est-ce que c’est la PDO ?

PDO (**P**HP **D**ata **O**bjects) vise à permettre la création d'un code comportant des accès aux BDD en faisant abstraction du moteur de SGBD utilisé.

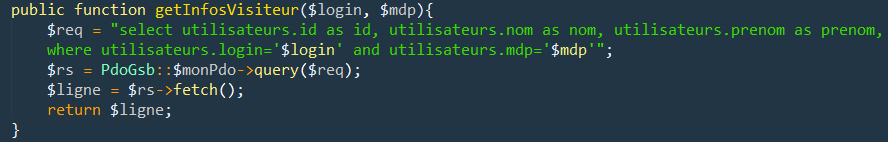
La connexion se crée dans le constructeur de la classe « class.pdogsb.inc.php » comme ceci dans notre application :



Et elle donne les valeurs des attributs :

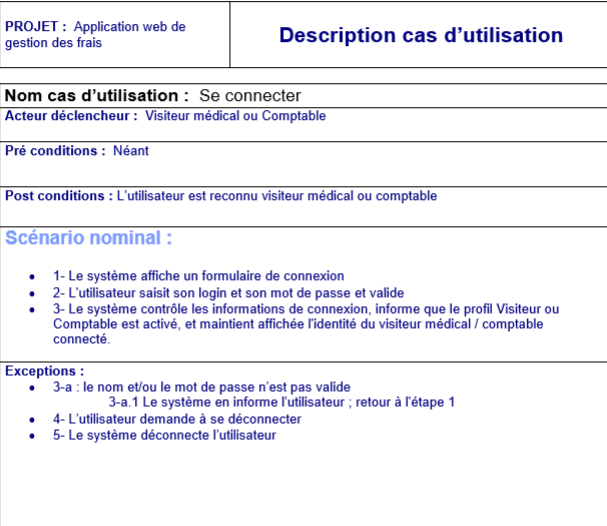


Et grâce à ça on peut créer des requêtes :



## Détail des pages

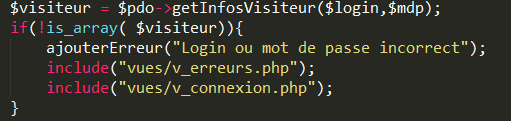
### Espace d’authentification



Cet espace permet l’accès à l’application par le biais d’une authentification.



Avant l’authentification, le contrôleur « c\_connexion » vérifie si cet utilisateur existe bien :

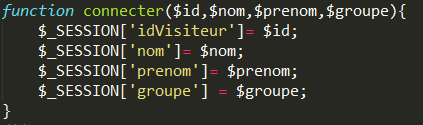


Dans le cas échéant il renvoie une erreur :

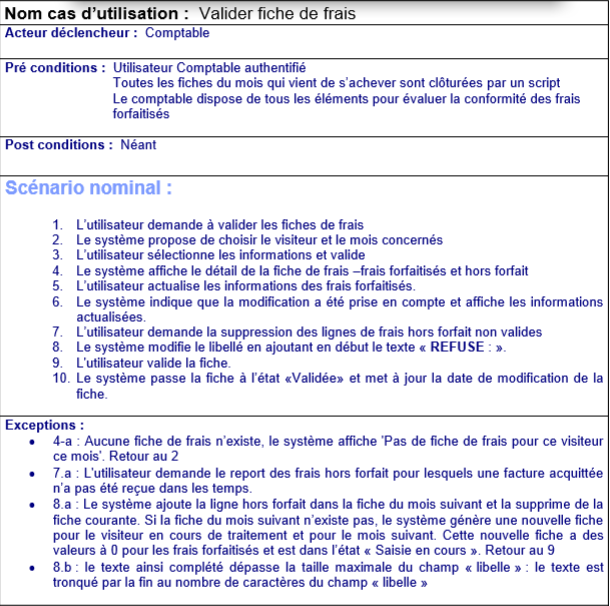


Une fois l’authentification réussite l’application sera d’elle-même si il s’agit d’un compte visiteur ou bien celui d’un comptable.

A partir de là, des variables de session se créaient :



### Validation des frais

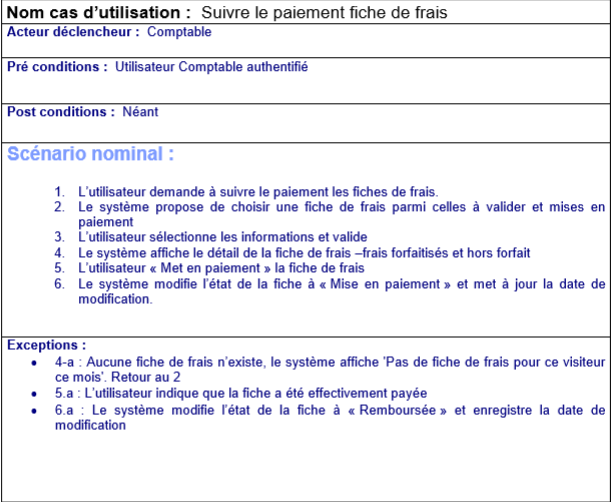


Espace réservé uniquement aux comptables, il permet de valider ou pas les frais ayant été rentrés par le visiteur.

Ces fiches sont générées en Ajax ainsi que toutes modifications portées dessus. (Tous se fait en temps réel).



### Consultation des frais



Cette fonction est disponible depuis l’espace du comptable mais aussi à celui des visiteurs. Elle permet à ces utilisateurs de visualiser les frais selon chaque mois de l’année. La seule différence entre les deux espaces réside dans le fait que l’espace comptable bénéficie d’une fonction en plus, celle de rechercher des fiches de frais de tous les visiteurs présent dans la base de donnée de l’entreprise et de pouvoir modifier son état, tant dis que le visiteur lui, ne peut consulter juste que sa propre fiche de frais.

### Déclaration des frais



Espace uniquement réservé aux visiteurs, il permet à tout moment de renseigner les derniers frais ayant été fait, que ce soit compris dans le forfait, ou également en hors-forfait.

Tout est en Ajax la aussi.



### Déconnexion

Accessible aux visiteurs et aux comptables, cet espace leurs permet simplement de se déconnecter de l’application ainsi toutes les variables de session seront détruites et l’utilisateur sera redirigé sur la page de connexion.

Destruction des variables :



## Bibliothèques externes utilisées

Pour la réalisation de cette application les bibliothèques externes suivantes seront utilisées :

* JQuery (Pour rajouter de l’Ajax est ainsi donner un meilleur confort d’utilisation).
* Bootstrap (Pour embellir graphiquement l’application).

# Proposition d’amélioration

Pour améliorer d’avantage cette application, des améliorations sont possibles:

Améliorations techniques :

* Migrer l’application sur un Framework
* Crypter le mot de passe dans la base de données
* Au niveau des UCs "Saisir fiche frais" et "Consulter fiche frais", prévoir l'affichage des éléments intermédiaires de calcul permettant d'apprécier le montant correspondant à chaque ligne de frais forfaitisé, ainsi que les totaux des éléments forfaitisés et hors forfait, de sorte à être plus proche du document relatif à la fiche de demande de remboursement (élément du cahier des charges non respecté pour l'instant).
* Modifier la base de données (et l’application) pour que l’on connaisse le statut d’une ligne de frais hors forfait (point 8 de l’UC « Valider Frais » qui ajoute le texte « REFUSE » en début de libellé.

Améliorations fonctionnelles :

* Création d’un nouveau module « Administrateur » qui permettrait d’avoir une vue d’ensemble sur les frais de l’entreprise.
* Les justificatifs fournis par les visiteurs peuvent être téléchargées directement sur l’application.
* Installation d’une boîte de messagerie qui permettrait de communiquer plus rapidement entre les visiteurs et les comptables.
* Distinguer l’indemnité kilométrique en fonction de la puissance du véhicule
* Porter l’application sur terminal mobile.
* Au niveau de l'UC "Consulter fiche frais", rendre la fiche de frais facilement imprimable, par exemple proposer la génération d'un fichier PDF (mise en œuvre proposée dans le contexte JPE de terminale GSI).

# Conclusion

La réalisation de ce projet m’a permis d’améliorer mes connaissances en HTML/CSS, php, SQL et Jquery ainsi qu’à la maîtrise de l’architecture MVC et de la POO (**P**rogrammation **O**rientée **O**bjet).