

Ppe 1 SLAM :

Appli Frais



Sommaire

I Contexte de la situation professionnelle

II Environnement de développement

III Modélisation fonctionnel

IV IHM

V Architecture Applicative

VI Persistances des données

VII Accès aux données

VIII Déploiement mise en production de l’application

IX tests fonctionnel

X Conclusion, amélioration possible

I Contexte de la situation professionnel

1. Le besoin

Améliorer l’existant d’APPLI-FRAIS en intégrant les comptables pour qu’il puisse agir sur les saisies des visiteurs réduisant le temps d’envoi d’information en papier et l’utilisation de celui-ci

Attention certaine contrainte sont à prendre en compte respecter l’architecture modèle vue contrôleur et respecter le thème déjà en place et bien sur tout ce qui est norme nom des fichiers variable paramètres

Et enfin de fournir une documentation de cette amélioration Framework, classe, bibliothèque et commentaire

1. L’existant

APPLI FRAIS prend en charge par défaut la partie Visiteur médical avec une architecture MVC et une base de données prête

II Environnement de développement

IDE :

Netbeans pour sa prise en main et l’

Langage :

PHP application / serveur

Html pour l’affichage

Méthodologie :

Schéma uml et modele relation de données

Algorithme en pseudo-code

Création de l’environnement Github Versionning et projet

Mise en Application en respectant

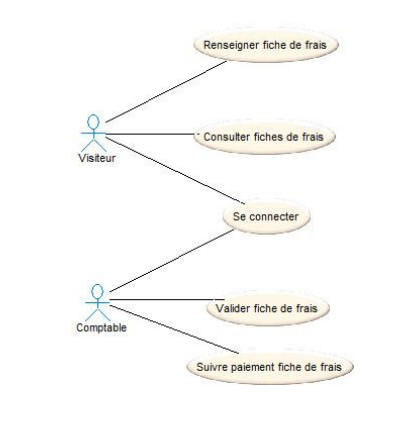
Modèle : SQL/PHP

Vue : HTML

Contrôleurs : PHP

III Modélisation Fonctionnelle

UML



Scénario

|  |
| --- |
| Nom cas d’utilisation : Se connecter |
| Acteur déclencheur : Visiteur médical ou Comptable |
| Pré conditions : Néant |
| Post conditions : L’utilisateur est reconnu visiteur médical ou comptable |
| Scénario nominal :  1- Le système affiche un formulaire de connexion |
| 2- L'utilisateur saisit son login et son mot de passe et valide |
| 3- Le système contrôle les informations de connexion, informe que le profil Visiteur ou Comptable est activé, et maintient affichée l'identité du visiteur médical / comptable connecté. |
| Exceptions : |
| 3-a : le nom et/ou le mot de passe n’est pas valide |
| 3-a.1 Le système en informe l’utilisateur ; retour à l'étape 1 |
| 4- L’utilisateur demande à se déconnecter |
| 5- Le système déconnecte l’utilisateur |
|  |

|  |
| --- |
| Nom cas d’utilisation : Suivre le paiement fiche de frais |
| Acteur déclencheur : Comptable |
| Pré conditions : Utilisateur Comptable authentifié |
| Post conditions : Néant |
| Scénario nominal : |
| 1. L’utilisateur demande à suivre le paiement les fiches de frais. |
| 2. Le système propose de choisir une fiche de frais parmi celles à valider et mises en paiement |
| 3. L’utilisateur sélectionne les informations et valide |
| 4. Le système affiche le détail de la fiche de frais –frais forfaitisés et hors forfait |
| 5. L’utilisateur « Met en paiement » la fiche de frais |
| 6. Le système modifie l’état de la fiche à « Mise en paiement » et met à jour la date de modification. |
| Exceptions : |
|  4-a : Aucune fiche de frais n’existe, le système affiche 'Pas de fiche de frais pour ce visiteur ce mois'. Retour au 2 |
|  5.a : L’utilisateur indique que la fiche a été effectivement payée |
|  6.a : Le système modifie l’état de la fiche à « Remboursée » et enregistre la date de modification |

|  |
| --- |
| Nom cas d’utilisation : Valider fiche de frais |
| Acteur déclencheur : Comptable |
| Pré conditions : 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 |
| Post conditions : Néant |
| Scénario nominal : |
| 1. L’utilisateur demande à valider les fiches de frais |
| 2. Le système propose de choisir le visiteur et le mois concernés |
| 3. L’utilisateur sélectionne les informations et valide |
| 4. Le système affiche le détail de la fiche de frais –frais forfaitisés et hors forfait  5. L’utilisateur actualise les informations des frais forfaitisés. |
| 6. Le système indique que la modification a été prise en compte et affiche les informations actualisées. |
| 7. L’utilisateur demande la suppression des lignes de frais hors forfait non valides |
| 8. Le système modifie le libellé en ajoutant en début le texte « REFUSE : ». |
| 9. L'utilisateur valide la fiche. |
| 10. Le système passe la fiche à l’état «Validée» et met à jour la date de modification de la fiche. |
| Exceptions : |
|  4-a : Aucune fiche de frais n’existe, le système affiche 'Pas de fiche de frais pour ce visiteur ce mois'. Retour au 2 |
|  7.a : L'utilisateur demande le report des frais hors forfait pour lesquels une facture acquittée n’a pas été reçue dans les temps. |
|  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 es t dans l’état « Saisie en cours ». Retour au 9 |
|  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 » |

IV IHM







V Architecture Applicative

Le pattern MVC permet de bien organiser son code source. Il va vous aider à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts, comme l'explique la description qui suit.

* **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.

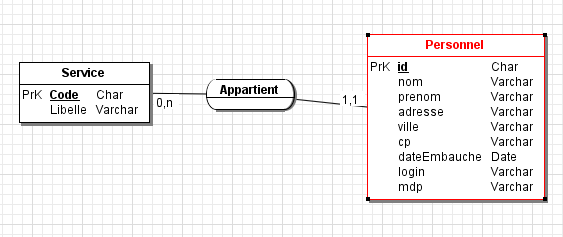
Parfois, les données ne sont pas stockées dans une base de données. C'est plus rare, mais on peut être amené à aller chercher des données dans des fichiers. Dans ce cas, le rôle du modèle est de faire les opérations d'ouverture, de lecture et d'écriture de fichiers (fonctions fopen, fgets, etc.).

* **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).\*

<http://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/organiser-son-code-selon-l-architecture-mvc>

VI Persistances des données

Schéma des données mis à jour MCD :



nous reprendrons l'ancien mcd en dupliquant la table visiteurs donc il conserve les mêmes propriété

script sql

-- --------------------------------------------------------  
-- Hôte : 127.0.0.1  
-- Version du serveur: 5.6.15-log - MySQL Community Server (GPL)  
-- SE du serveur: Win32  
-- HeidiSQL Version: 9.2.0.4947  
-- --------------------------------------------------------

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  
/\*!40101 SET NAMES utf8mb4 \*/;  
/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;  
/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

-- Export de la structure de la base pour entreprise  
CREATE DATABASE IF NOT EXISTS `entreprise` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;  
USE `entreprise`;

-- Export de la structure de table entreprise. comptable  
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` char(20) DEFAULT NULL,  
 `adresse` char(30) DEFAULT NULL,  
 `cp` char(5) DEFAULT NULL,  
 `ville` char(30) DEFAULT NULL,  
 `dateEmbauche` date DEFAULT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- Export de données de la table entreprise.comptable : 1 rows  
DELETE FROM `comptable`;  
/\*!40000 ALTER TABLE `comptable` DISABLE KEYS \*/;  
INSERT INTO `comptable` (`id`, `nom`, `prenom`, `login`, `mdp`, `adresse`, `cp`, `ville`, `dateEmbauche`) VALUES  
 ('Comp', 'Hernandez', 'Nicolas', 'nh', 'test', '', NULL, NULL, NULL);  
/\*!40000 ALTER TABLE `comptable` ENABLE KEYS \*/;  
/\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;  
/\*!40014 SET FOREIGN\_KEY\_CHECKS=IF(@OLD\_FOREIGN\_KEY\_CHECKS IS NULL, 1, @OLD\_FOREIGN\_KEY\_CHECKS) \*/;  
/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

VII Accès aux données

PDO fournit une interface d'abstraction à l'accès de données, ce qui signifie que vous utilisez les mêmes fonctions pour exécuter des requêtes ou récupérer les données quelque soit la base de données utilisée.

http://php.net/manual/fr/intro.pdo.php

Expliquer les trois méthodes de pdogsb les plus importantes

Design Pattern Singleton :

**Une classe, une instance : le pattern Singleton**

Nous avons une classe qui ne doit être instanciée qu'une seule fois.

Il faut donc bien penser à interdire l'accès à la méthode \_\_clone().

http://openclassrooms.com/courses/programmez-en-oriente-objet-en-php/les-design-patterns

Ce qui convient parfaitement à l’accès de la base de données n’avoir qu’une seule instance pour traiter les données

VIII Déploiement, mise en production de l’application

1. développement terminé sur le pré-prod et fonctionnel
2. le basculé sur le serveur production
3. exécuter le script sql
4. importer les comptes comptables grâce à un script

IX tests fonctionnels

|  |  |  |  |
| --- | --- | --- | --- |
| **N°** | **Action** | **Résultat attendu** | **OK / KO / N** |
| 1 | Le système affiche un formulaire de connexion | affichage du formulaire | OK |
| 2 | L'utilisateur saisi son mot de passe et login | possibilité de saisir et envois des données | OK |
| 3 | 3- Le système contrôle les informations de connexion | informe que le profil Visiteur ou Comptable est activé, et maintient affichée l'identité du visiteur médical / comptable connecté. | OK |
| 4 | L’utilisateur demande à suivre le paiement les fiches de frais. | direction sur une page paiement fiche de frais | OK |
| 5 | Le système propose de choisir une fiche de frais parmi celles à valider et mises en paiement | propose de choisir des fiches clôturés par visiteur | OK |
| 6 | Le système affiche le détail de la fiche de frais –frais forfaitisés et hors forfait | affiche en détails la fiche du visiteur | OK |
| 7 | L’utilisateur « Met en paiement » la fiche de frais |  | NN |
| 8 | Le système modifie l’état de la fiche à « Mise en paiement » et met à jour la date de modification. |  | NN |
| 9 | L’utilisateur demande à suivre le paiement les fiches de frais. |  | NN |
| 10 | Le système propose de choisir une fiche de frais parmi celles à valider et mises en paiement |  | NN |
| 11 | L’utilisateur sélectionne les informations et valide |  | NN |
| 12 | Le système affiche le détail de la fiche de frais –frais forfaitisés et hors forfait |  | NN |
| 13 | L’utilisateur « Met en paiement » la fiche de frais |  | NN |
| 14 | Le système modifie l’état de la fiche à « Mise en paiement » et met à jour la date de modification. |  | NN |

X conclusions et améliorations

 Crypter le mot de passe dans la base de données

Mda5 sur la le champ mot de passe

 Distinguer l’indemnité kilométrique en fonction de la puissance du véhicule

Faire un rapport indemnité km et véhicule, lorsque nous aurons les données de gsb concernant leur règles par rapport a la puissance de celles-ci

 Porter l’application sur terminal mobile.

Bootstrap : responsive design

 Automatiser l’alimentation des manipulations comptables sur l’application depuis le PGI de l’entreprise

?

 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).

Généré un pdf et l’afficher en tant que pdf voir extension

 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).

tableau

Afficher tout les étapes du calculs + résultat par ligne

Total frais forfaitisé

 Modifier la base de données (et l’application) pour que l’on connaisse le statut d’un ligne de frais hors forfait (point 8 de l’UC « Valider Frais » qui ajoute le texte « REFUSE » en début de libellé.

Ajouté refusé début libellé

 Adopter smarty pour les vues ou un framework

Augmenté les performances en utilisant des requêtes ajax pour gagné du temps en n'empêchant de recharger toute la page

clôturé automatiquement les fiches de frais du mois en cours le 20 du mois