

# GSB - Applifrais

Compte-rendu de la démarche de développement de l'application



- GSB - Applifrais
  - Ressources fournies
  - Cahier des charges
  - Modification de la base de données
  - Design de l'interface utilisateur
  - Architecture logicielle de la nouvelle application
  - Présentation des modules
    - Page de connexion
    - Menu
    - Page de saisie de frais du mois en cours (Visiteur)
    - Page de visualisation de ses frais (Visiteur)
    - Page de saisie du visiteur et du mois des frais (Comptable)
    - Page de visualisation des frais d'un visiteur et d'un mois donnés (Comptable)
    - Module de validation des frais d'un visiteur et d'un mois donnés (Comptable)
    - Page de modification des frais d'un visiteur et d'un mois donnés (Comptable)
  - Intégration continue avec [GitHub Actions](#)

## Ressources fournies

Divers ressources ont été fournies afin de réaliser l'application :

- Documents relatifs à la présentation du contexte GSB
  - Présentation de l'organisation
  - Fiche descriptive de l'ancienne application
- Script de l'ancienne base de données

- Code source de l'ancienne application en PHP pur
- Code source de l'ancienne application en architecture MVC + PDO

## Cahier des charges

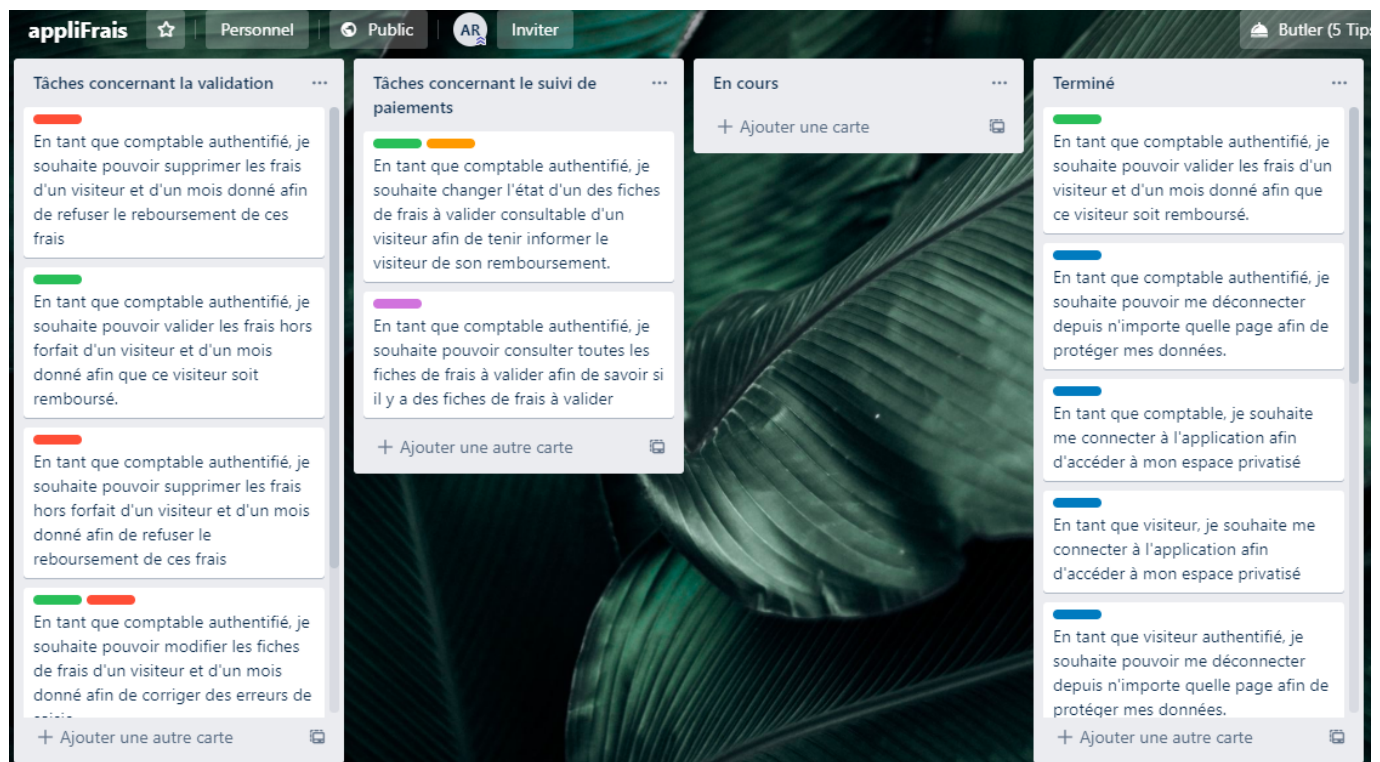
À l'aide du contexte donné, j'ai étudié l'ancienne application Applifrais ainsi que l'expression du nouveau besoin.

La nouvelle application doit donc inclure un système de droits entre les différents utilisateurs ainsi qu'un nouveau type d'utilisateur : le **comptable**.

Cet utilisateur possède des fonctionnalités propre à lui et des droits plus important concernant la base de données (Suppression, Modification et Ajout de données). Ces fonctionnalités propres au comptable sont **la gestion des paiements** et **la gestion de fiche de frais**.

La gestion des paiements consiste à gérer l'état de validation des paiements d'une fiche de frais donnée. La gestion de fiche frais concerne la visualisation, la modification et la suppression d'une fiche de frais donnée.

Pour la gestion de projet, j'ai choisi la méthode agile dite Kanban avec l'application **Trello** :



## Modification de la base de données

Afin de différencier les deux types d'utilisateur, il nous faut modifier la structure de la base de données en renommant la table **visiteur** par **utilisateur** et en ajoutant un champ permettant de différencier les visiteurs des comptables :

```

1 // Modification de la table visiteur
2 alter table visiteur rename to utilisateur;
3 // Création d'un champ pour faire la distinction entre visiteur et comptable
4 alter table utilisateur add typeUtilisateur char(1);
5 // Initialisation
6 update utilisateur set type typeUtilisateur = 'V';
7 // Ajout d'un compte comptable
8 insert into utilisateur(nom, prenom, login, mdp, adresse, cp, ville, dateEmbauche)
9 values ('Richard', 'Arthur', 'arthurrhd', '12345', '40 rue victor bertel', '76000', 'Rouen', '2019-10-20', 'C')

```

Aperçu de la modification avec `phpMyAdmin` :

id	nom	prenom	login	mdp	adresse	cp	ville	dateEmbauche	typeUtilisateur
a131	Villechalane	Louis	lvillachane	jux7g	8 rue des Charmes	46000	Cahors	2005-12-21	V
a17	Andre	David	dandre	opp95	1 rue Petit	46200	Lalbenque	1998-11-23	V
a55	Bedos	Christian	cbedos	gmhxd	1 rue Peranud	46250	Montcuq	1995-01-12	V
a7	Richard	Arthur	arthurrhd	12345	40 rue victor bertel	76000	Rouen	2019-10-20	C
a93	Tusseau	Louis	ltusseau	ktp3s	22 rue des Ternes	46123	Gramat	2000-05-01	V
k12	Bertel	David	dbertel	daud1	11 allée des Grises	46512	Bergerac	1992-07-09	V

## Design de l'interface utilisateur

Pour l'interface utilisateur, j'ai opté pour des couleurs froides, sobres et professionnelles.



```

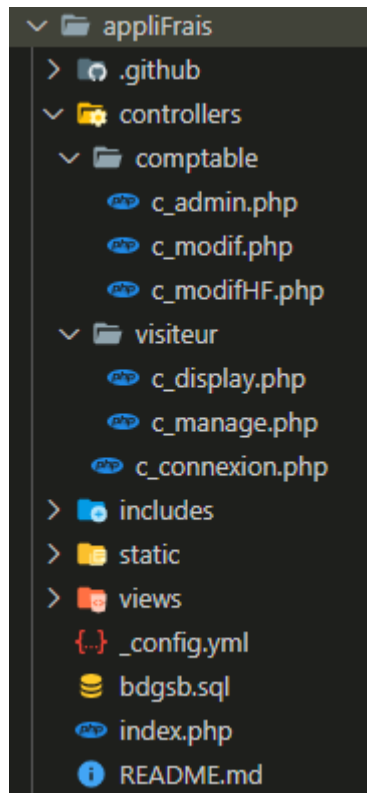
html .container {
  background: rgb(59, 59, 148);
  color: white;
}

```

La stylisation restante est assurée par le framework front-end `Bootstrap 4`, qui permet de créer des composants déjà prototypés en nommant.

## Architecture logicielle de la nouvelle application

Pour ajouter la partie comptable à l'application, j'ai repris le code source de base de l'application en MVC + PDO. J'ai par la suite modifier l'architecture de l'application pour faciliter la lecture du code source. L'architecture maintenant appropriée, j'ai développé des contrôleurs pour la gestion de session, la gestion des paiements et la gestion de fiche de frais.



Voici à quoi ressemble le contrôleur faisant office de routeur pour les autres contrôleurs sous une architecture MVC :

```
# index.php
<?php
require_once("includes/fct.inc.php");
require_once("includes/class.pdogsbs.inc.php");
include("views/base/v_entete.php") ;
session_start();
$pdo = PdoGsb::getPdoGsb();
$estConnecte = estConnecte();
if(!isset($_GET['uc']) || !$estConnecte){
    $_REQUEST['uc'] = 'connexion';
}
$uc = $_REQUEST['uc'];
switch($uc){
    case 'connexion':
        include("controllers/c_connexion.php");break;
    case 'manage' :
        include("controllers/visiteur/c_manage.php");break;
    case 'display' :
        include("controllers/visiteur/c_display.php");break;
    case 'admin':
        include("controllers/comptable/c_admin.php");break;
    case 'modif':
```

```
include("controllers/comptable/c_modif.php");break;
}
include("../views/base/v_pied.php") ;
```

## Présentation des modules

### Page de connexion



### Menu

Le menu différencie les visiteurs des comptables, et cela permet de rediriger les utilisateurs vers les modules auxquels ils ont le droit d'accéder.





## Page de saisie de frais du mois en cours (Visiteur)

The screenshot displays the 'Fiche de frais datant du 05-2020' page. The header is dark grey with a green button labeled 'VISITEUR Christian Bedos' on the left, followed by the links 'Saisie fiche de frais', 'Mes fiches de frais', and 'Déconnexion'. The main title 'Fiche de frais datant du 05-2020' is centered in a large white font. Below the title, there are two main sections: 'Eléments forfaitisés' and 'Nouvel élément hors forfait'. The 'Eléments forfaitisés' section has the instruction 'Veuillez remplir ces champs' and four input fields: 'Forfait Etape' (value 4), 'Frais Kilométrique' (value 0), 'Nuitée Hôtel' (value 0), and 'Repas Restaurant' (value 0). At the bottom of this section are 'Valider' and 'Effacer' buttons. The 'Nouvel élément hors forfait' section also has the instruction 'Veuillez remplir ces champs' and three input fields: 'Date (jj/mm/aaaa):', 'Libellé', and 'Montant :'. At the bottom of this section are 'Ajouter' and 'Effacer' buttons. At the bottom of the page, there is a section titled 'Descriptif des éléments hors forfait' which contains a table with three columns: 'Date', 'Libellé', and 'Montant'.

Date	Libellé	Montant
------	---------	---------

## Page de visualisation de ses frais (Visiteur)

Suivi du remboursement des frais

**VISITEUR**  
Christian Bedos

Saisie fiche de frais   Mes fiches de frais   Déconnexion

## Mes fiches de frais

Mois à sélectionner :

04/2020

Valider   Effacer

### Fiche de frais du mois 04-2020

Christian Bedos

Etat : Saisie clôturée depuis le 05/05/2020  
Montant validé : 0.00

#### Quantités des éléments forfaitisés

Forfait Etape	Frais Kilométrique	Nuitée Hôtel	Repas Restaurant
4	0	0	0

#### Descriptif des éléments hors forfait -0 justificatifs reçus -

Date	Libellé	Montant
------	---------	---------

Page de saisie du visiteur et du mois des frais (Comptable)

Suivi du rembo

**COMPTABLE**  
Arthur Richard

Valider des fiches de frais   Déconnexion

### Validation

Visiteur   Christian Bedos   ⇅   Valider

Mois   201910   ⇅   C'est parti !

## Page de visualisation des frais d'un visiteur et d'un mois donnés (Comptable)

COMPTABLE  
Arthur Richard

Valider des fiches de frais

Déconnexion

Remboursée

Validée et mise en paiement

Saisie cloturée

Modifier

# Fiche de frais du mois 10-2019

## Christian Bedos

Etat : Validée et mise en paiement depuis le 06/11/2019  
Montant validé : 3061.58

### Quantités des éléments forfaitisés

Forfait Etape	Frais Kilométrique	Nuitée Hôtel	Repas Restaurant
5	250	12	50

### Descriptif des éléments hors forfait -0 justificatifs reçus -

Date	Libellé	Montant
09/10/2019	Parc astérix	52.00
22/10/2019	location salle conférence	251.00
13/10/2019	achat de matériel de papèterie	17.00
18/10/2019	taxi	22.00

## Module de validation des frais d'un visiteur et d'un mois donnés (Comptable)

Remboursée

Validée et mise en paiement

Saisie cloturée

## Page de modification des frais d'un visiteur et d'un mois donnés (Comptable)



**COMPTABLE**  
**Arthur Richard**

Valider des fiches de frais   Déconnexion

Quitter la modification

## Fiche de frais datant du 10-2019

Nom du visiteur : **Christian Bedos**

### Éléments forfaitisés

Veuillez remplir ces champs

Forfait Etape   5

Frais Kilométrique   250

Nuitée Hôtel   12

Repas Restaurant   50

Valider   Effacer

### Nouvel élément hors forfait

Veuillez remplir ces champs

Date (jj/mm/aaaa):

Libellé

Montant :

Ajouter   Effacer

### Descriptif des éléments hors forfait

Date	Libellé	Montant	Refuser le frais	Restaurer le frais	Reporter à la semaine prochaine
09/10/2019	Parc astérix	52.00	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
22/10/2019	location salle conférence	251.00	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13/10/2019	achat de matériel de papèterie	17.00	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18/10/2019	taxi	22.00	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Actualiser   Annuler

## Intégration continue avec **GitHub Actions**

Voici le script d'automatisation des tests PHP

```
# This is a basic workflow to help you get started with Actions

name: PHP CI

# Controls when the action will run. Triggers the workflow on push or pull request
# events but only for the master branch
on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]

# A workflow run is made up of one or more jobs that can run sequentially or in
parallel
```

```

jobs:
  # This workflow contains a single job called "build"
  build:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access
      it
      - uses: actions/checkout@v2

      - name: PHP Runner
        uses: franzliedke/gh-action-php@0.2.0

```

Voici le script d'automatisation des tests SQL

```

# This is a basic workflow to help you get started with Actions

name: SQL CI

# Controls when the action will run. Triggers the workflow on push or pull request
# events but only for the master branch
on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]


# A workflow run is made up of one or more jobs that can run sequentially or in
# parallel
jobs:
  # This workflow contains a single job called "build"
  build:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access
      it
      - uses: actions/checkout@v2

      # Runs a single command using the runners shell
      - name: SQLCheck Action
        uses: yokawasa/action-sqlcheck@1.2.1
        with:
          token: 3
          # Set of SQL anti-patterns to check: 1,2, or 3
          postfixes: sql

```

Ajout des badges sur le dépôts GitHub afin de tester l'intégrité de l'application lors d'un commit :

 README.md

# appliFrais

 SQL CI **passing**  PHP CI **passing**