

# PROJET APPLI-FRAIS

## Application d'enregistrement des frais engagés et de suivi des remboursements

---

### I. Présentation de l'entreprise

### II. Présentation de l'application

### III. Détails de l'application

#### A. Authentification :

#### B. Affichage des fiches de frais :

#### C. Saisie/modification des fiches de frais :

#### D. Modification de l'état d'une fiche de frais(comptable) :

### IV. Fonctionnement du code

#### A. Controllers :

#### B. Entity

#### C. Formulaires

#### D. Repository

### V. Tests fonctionnels

#### A. Authentification d'un utilisateur à l'application

#### B. Affichage des fiches frais

#### C. Saisie des fiches de frais

#### D. Modification des fiches de frais

#### E. Modification de l'état d'une fiche

---

## I. Présentation de l'entreprise

GSB est une entreprise pharmaceutique qui a été créée par la fusion entre le géant américain Galaxy et le conglomérat européen Swiss Bourdin. Elle se spécialise dans les médicaments pour les maladies virales et conventionnelles, avec son siège administratif à Paris et son siège social à Philadelphie. Après la fusion, l'entreprise vise à améliorer ses opérations en réalisant des économies d'échelle grâce à des restructurations et des licenciements, tout en préservant les atouts des deux laboratoires. Chez GSB, les représentants médicaux doivent conseiller et informer les professionnels de la santé sur les produits de l'entreprise sans réaliser des

ventes directes. Les délégués médicaux sont organisés selon le modèle existant chez Galaxy, avec une hiérarchie par région et par secteur géographique.

L'entreprise vise à gérer efficacement les dépenses liées aux déplacements et aux actions sur le terrain tout en préservant son image de marque. Elle veut unifier la gestion des dépenses et des remboursements pour tous les visiteurs.

## II. Présentation de l'application

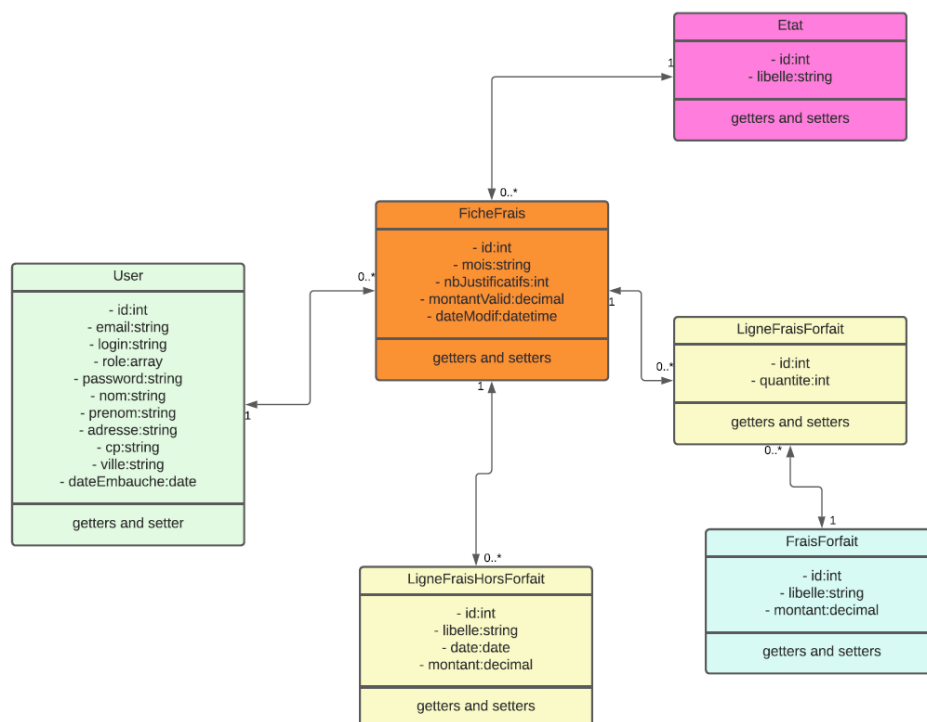
Une application web Symfony pour gérer les frais de déplacement des visiteurs médicaux a donc été mise en place

L'application permet d'enregistrer tous les frais engagés, y compris les dépenses directes (déplacements, restauration, hébergement) et les activités annexes (événementiel, conférences, etc.). Elle offre un suivi daté des opérations menées par le service comptable, depuis la réception des pièces jusqu'au remboursement effectué.

Cette application web est accessible et son accès est restreint aux seuls acteurs de l'entreprise, nécessitant une authentification préalable. Elle respecte l'architecture Model-View-Controller et utilise le PHP comme langage côté serveur.

Les dépenses de déplacement sont gérées en détail, avec un suivi précis des coûts et un processus informatisé pour la soumission, le traitement et l'approbation des demandes de remboursement. Il est possible pour les visiteurs d'entrer les frais forfaitaires et hors forfait, avec la possibilité de les modifier à tout moment au cours du mois. À la fin du mois, les fiches sont automatiquement clôturées et soumises à une campagne de validation par le service comptable entre le 10 et le 20 du mois suivant. Les visiteurs font la saisie et les remboursements sont effectués au 20 du mois suivant, avec un suivi précis de l'état de chaque fiche.

Le diagramme de classe de l'application :



### III. Détails de l'application

L'application a donc plusieurs fonctionnalités :

#### A. Authentification :

Le contrôleur *SecurityController* gère l'authentification des utilisateurs. La méthode `index` gère la page de connexion. Elle utilise `AuthenticationUtils` pour récupérer d'éventuelles erreurs de connexion et le dernier nom d'utilisateur saisi. Ces informations sont passées au template `security/index.html.twig` pour l'affichage. La méthode `logout` gère la déconnexion.

#### B. Affichage des fiches de frais :

Le contrôleur *MesFichesFraisController* gère l'affichage des fiches de frais pour l'utilisateur connecté. La méthode `selectedMonth` vérifie d'abord si l'utilisateur est authentifié, puis récupère les fiches de frais de l'utilisateur à partir de la base de données via l'EntityManager. Ensuite, elle crée un formulaire de sélection du mois en utilisant `MonthSelectorFormType`, et gère la soumission du formulaire pour obtenir la fiche de frais sélectionnée. Enfin, elle rend la vue `mes_fiches_frais/index.html.twig`, en passant le formulaire et la fiche de frais sélectionnée pour l'affichage.

### **C. Saisie/modification des fiches de frais :**

Le contrôleur *SaisirFicheFraisController* gère la création et la saisie des fiches de frais pour l'utilisateur connecté. Le contrôleur vérifie d'abord si l'utilisateur est authentifié lorsque celui-ci accède à la route `/saisirFicheFrais`. Il obtient ensuite la fiche de frais du mois en cours de l'utilisateur à partir de la base de données. Si la fiche de frais pour le mois actuel n'est pas disponible, elle sera créée avec des valeurs par défaut. Après cela, le contrôleur s'occupe de la création des formulaires pour les frais forfaitisés et les frais hors forfaits.

Les quantités saisies sont enregistrées dans la base de données si le formulaire de frais forfaitisés est soumis et valide. De plus, les informations sont également enregistrées si le formulaire de saisie des frais hors forfait est soumis et valide. Finalement, le contrôleur rend la vue `saisir_fiche_frais/index.html.twig` et affiche la fiche de frais, le formulaire de saisie des frais forfaitisés ainsi que le formulaire de saisie des frais hors forfait.

### **D. Modification de l'état d'une fiche de frais(comptable) :**

Le contrôleur *ComptablesController* gère la récupération des fiches frais de chaque utilisateur(sélectionné par le comptable grâce à un formulaire de type `ChoiceType` qui récupère chaque utilisateur dans la base de données) et les affiche dans un tableau, présent dans la vue `comptables/index.html.twig`, qui affiche le mois, l'état et un bouton qui

permet de consulter la fiche de choix. Ce bouton mène à la vue gérée par le contrôleur *FicheFraisController*. Ce contrôleur gère l'affichage d'une fiche de frais individuelle et permet au comptable de modifier l'état de cette fiche. Pour permettre que ces actions soient possibles seulement pour le comptable connecté, un rôle comptable a été créé dans *config/packages/security.yaml* puis l'accès à la page a été accordé seulement lorsqu'un rôle comptable est connecté dans le template de base.

## IV. Fonctionnement du code

Grâce à l'utilisation de l'architecture MVC dans l'application, le code est plus ergonomique et se constitue de la manière suivante :

### A. Controllers :

#### - EtatController :

Ce controller gère les opérations CRUD pour l'entité "Etat". Quand l'utilisateur accède à la route /etat, le contrôleur affiche une liste de tous les états enregistrés dans la base de données. Lorsqu'il accède à la route /etat/new, le contrôleur affiche un formulaire pour créer un nouvel état. Lorsque le formulaire est soumis et valide, l'état nouvellement enregistré est stocké dans la base de données.

L'utilisation de la route /etat/{id} permet à l'utilisateur de voir les détails d'un état spécifique. Le contrôleur obtient l'état associé à l'identifiant fourni dans l'URL et affiche la vue associée, etat/show.html.twig.

L'utilisateur peut modifier un état existant en utilisant la route /etat/{id}/edit. Le contrôleur affiche un formulaire déjà rempli avec les données de l'état à modifier. Lorsque le formulaire est soumis et validé, les changements sont enregistrés dans la base de données, et la vue affichée est etat/edit.html.twig.

Finalement, avec la méthode POST, l'utilisateur peut supprimer un état existant en empruntant la route /etat/{id}. Pour des raisons de sécurité, le contrôleur commence par vérifier le jeton CSRF,

puis supprime l'état de la base de données. Après avoir effectué l'opération dans chaque cas, l'utilisateur est redirigé vers la liste des états.

- FraisForfaitController :

Ce contrôleur gère les opérations CRUD pour l'entité "FraisForfait". Lorsque l'utilisateur accède à la route /fraisforfait, le contrôleur affiche une liste de tous les frais forfaitaires enregistrés dans la base de données.

L'utilisation de la route /fraisforfait/new permet à l'utilisateur de créer un nouveau frais forfaitisé. Le contrôleur affiche un formulaire vide pour saisir les détails du frais. Si le formulaire est soumis et valide, le nouveau frais forfaitisé est enregistré dans la base de données, la vue affichée est frais\_forfait/new.html.twig.

L'utilisateur peut visualiser les détails d'un frais forfaitisé spécifique en accédant à la route /fraisforfait/{id}. Le contrôleur récupère le frais forfaitaire correspondant à l'identifiant fourni dans l'URL et l'affiche dans la vue frais\_forfait/show.html.twig. La route /fraisforfait/{id}/edit permet à l'utilisateur de modifier un frais forfaitaire existant. Le contrôleur affiche un formulaire pré-rempli avec les données du frais forfaitaire à modifier. Si le formulaire est soumis et valide, les modifications sont enregistrées dans la base de données, la vue affichée est frais\_forfait/edit.html.twig. Enfin, la route /fraisforfait/{id} avec la méthode POST permet à l'utilisateur de supprimer un frais forfaitaire existant. Le contrôleur vérifie d'abord le jeton CSRF pour des raisons de sécurité, puis supprime le frais forfaitaire de la base de données.

Après avoir effectué l'opération dans chaque cas, l'utilisateur est redirigé vers la liste des frais forfaitisés.

- LigneFraisHorsForfaitController :

Ce contrôleur gère les opérations CRUD pour l'entité LigneFraisHorsForfait. Lorsque l'utilisateur accède à la route /ligne/frais/hors/forfait, le contrôleur affiche une liste de toutes les lignes de frais hors forfait enregistrées dans la base de données et renvoie la vue ligne\_frais\_hors\_forfait/index.html.twig.

La route `/ligne/frais/hors/forfait/new` permet à l'utilisateur de créer une nouvelle ligne de frais hors forfait. Le contrôleur affiche un formulaire vide pour saisir les détails de la ligne de frais hors forfait. Si le formulaire est soumis et valide, la nouvelle ligne de frais hors forfait est enregistrée dans la base de données et renvoie la vue `ligne_frais_hors_forfait/new.html.twig`.

La route `/ligne/frais/hors/forfait/{id}` permet à l'utilisateur de visualiser les détails d'une ligne de frais hors forfait spécifique. Le contrôleur récupère la ligne de frais hors forfait correspondant à l'identifiant fourni dans l'URL et l'affiche dans la vue `ligne_frais_hors_forfait/show.html.twig`.

La route `/ligne/frais/hors/forfait/{id}/edit` permet à l'utilisateur de modifier une ligne de frais hors forfait existante. Le contrôleur affiche un formulaire pré-rempli avec les données de la ligne de frais hors forfait à modifier. Si le formulaire est soumis et valide, les modifications sont enregistrées dans la base de données et renvoie la vue `ligne_frais_hors_forfait/edit.html.twig`.

Enfin, la route `/ligne/frais/hors/forfait/{id}` avec la méthode POST permet à l'utilisateur de supprimer une ligne de frais hors forfait existante. Le contrôleur vérifie d'abord le jeton CSRF pour des raisons de sécurité, puis supprime la ligne de frais hors forfait de la base de données.

## B. Entity

L'application possède 6 six classes différentes dans la base de données. Certaines de ses classes sont reliées par différentes relations.

- Il existe une relation One-to-Many entre l'entité Etat et l'entité FicheFrais, cette relation est maintenue par la propriété `etat` dans l'entité FicheFrais.
- Il existe une relation Many-to-One entre l'entité FicheFrais et l'entité Etat, cette relation est maintenue par la propriété `ficheFrais` dans l'entité Etat.

Il y a également une relation One-to-Many entre l'entité FicheFrais et l'entité LigneFraisForfait, cette relation est maintenue par la propriété

fichefrais dans l'entité LigneFraisForfait. Cette entité doit être supprimée de la base de données si elle n'est plus référencée par l'entité parente(FicheFrais).

Enfin, il y a une relation One-to-Many entre l'entité FicheFrais et l'entité LigneFraisHorsForfait, cette relation est maintenue par la propriété fichefrais de l'entité LigneFraisHorsForfait. Cette entité doit être supprimée de la base de données si elle n'est plus référencée par l'entité parente(FicheFrais).

- Il existe une relation One-to-Many entre l'entité FraisForfait et l'entité LigneFraisForfait, cette relation est maintenue par la propriété fraisforfait dans l'entité LigneFraisForfait.
- Il existe une relation Many-to-One entre l'entité LigneFraisForfait et les entités FicheFrais et FraisForfait. Chaque instance de l'entité actuelle appartient à une seule instance de FicheFrais et à une seule instance de FraisForfait. Ces relations sont maintenues par les propriétés ligneFraisForfaits dans les entités FicheFrais et FraisForfait respectivement.
- Il existe une relation Many-to-One entre l'entité LigneFraisHorsForfait et l'entité FicheFrais, cette relation est maintenue par la propriété ligneFraisHorsForfaits dans l'entité FicheFrais.
- Il existe une relation One-to-Many entre User actuelle et l'entité FicheFrais, cette relation est maintenue par la propriété user dans l'entité FicheFrais.

## C. Formulaires

- Le formulaire EtatType est utilisé pour créer ou modifier des entités Etat. Il contient un seul champ libelle qui est ajouté dans la méthode buildForm. Ce champ correspond à la propriété libelle de l'entité Etat. La méthode configureOptions est utilisée pour configurer les options pour ce formulaire. Ici, elle définit l'option data\_class à Etat::class, ce qui signifie que ce formulaire est destiné à créer ou modifier des instances de la classe Etat. Il est appelé dans le controller Etat.
- Le formulaire FraisForfaitType est utilisé pour créer ou modifier des entités FraisForfait. Il contient deux champs : libelle et montant, qui sont ajoutés dans la méthode buildForm. Ces champs correspondent aux propriétés libelle et montant de l'entité



FraisForfait. La méthode `configureOptions` est utilisée pour configurer les options pour ce formulaire. Ici, elle définit l'option `data_class` à `FraisForfait::class`. Il est appelé dans le controller `FraisForfait`

- Le formulaire `LigneFraisHorsForfaitType` est utilisé pour créer ou modifier des entités `LigneFraisHorsForfait`. Il contient quatre champs : libelle, date, montant et fichefrais, qui sont ajoutés dans la méthode `buildForm`. Ces champs correspondent aux propriétés libelle, date, montant et fichefrais de l'entité `LigneFraisHorsForfait`. La méthode `configureOptions` est utilisée pour configurer les options pour ce formulaire. Ici, elle définit l'option `data_class` à `LigneFraisHorsForfait::class`. Il est appelé dans le controller `LigneFraisHorsForfait`.
- Le formulaire `MonthSelectorFormType` est utilisé pour créer un sélecteur de mois. Il contient un seul champ `selectedMonth` de type `ChoiceType`, qui est ajouté dans la méthode `buildForm`. Ce champ est utilisé pour afficher une liste de choix, qui sont fournis par l'option `choices` qui est définie comme `$mesFiches`, une variable passée à ce formulaire. L'option `label` est utilisée pour définir le libellé du champ, qui est "Sélectionner un mois" dans ce cas. L'option `choice_label` est une fonction qui prend un choix et retourne une chaîne de caractères qui sera utilisée comme libellé pour ce choix. Ici, elle prend une instance de `FicheFrais`, extrait le mois sous forme de chaîne de caractères, le convertit en une instance de `DateTimeImmutable`, puis le formate en 'M-Y' pour l'afficher. il est appelé dans les controllers `MesFichesFrais` Et `SaisirFicheFrais`.
- Le formulaire `SaisirFraisForfaitType` est utilisé pour saisir les frais forfaitisés. Il contient quatre champs : `quantiteEtape`, `quantiteKm`, `quantiteNuit` et `quantiteRepas`, qui sont ajoutés dans la méthode `buildForm`. Ces champs sont de type `IntegerType` et correspondent aux quantités d'étapes, de kilomètres parcourus, de nuitées et de repas respectivement. Chaque champ a un libellé personnalisé défini par l'option `label` et une contrainte de valeur minimale de 0 définie par l'attribut `min` dans l'option `attr`. Il est appelé dans le controller `SaisirFicheFrais`.
- Le formulaire `SaisirFraisHorsForfaitType` est utilisé pour saisir les frais hors forfait. Il contient trois champs : libellé, date et montant, qui sont ajoutés dans la méthode `buildForm`. Ces champs sont de type `TextType`, `DateType` et `IntegerType` respectivement et

correspondent au libellé, à la date et au montant des frais hors forfait. Chaque champ a un libellé personnalisé défini par l'option label. Le champ montant a également une contrainte de valeur minimale de 0 définie par l'attribut min dans l'option attr. Il est appelé dans le controller SaisirFicheFrais.

## D. Repository

Les repository sont des classes de repository Symfony qui sont utilisées pour interagir avec la base de données pour les entités concernée. Ils héritent de ServiceEntityRepository, qui fournit des méthodes pour récupérer des entités de la base de données. En plus de ces méthodes héritées, chacun des repository définit deux méthodes supplémentaires: add et remove.

## V. Tests fonctionnels

### A. Authentification d'un utilisateur à l'application

#### Description :

Assurer le bon fonctionnement du système d'authentification en autorisant l'accès aux utilisateurs ayant des identifiants valides tout en bloquant l'accès pour ceux qui n'en ont pas.

#### Conditions préalables :

- L'application web est accessible.
- L'utilisateur dispose d'un compte enregistré dans la base de données.

#### Étapes :

- 1) Accéder à la page de connexion :  
Entrer l'URL de l'application dans la barre d'adresse et appuyer sur Entrée. La page de connexion doit s'afficher.
- 2) Saisir les identifiants de connexion :

Sur la page de connexion, entrer un nom d'utilisateur et un mot de passe valides dans les champs appropriés. Appuyer sur le bouton de connexion.

3) Authentification réussie :

Si les identifiants sont valides, l'utilisateur doit être redirigé vers la page d'accueil ou une page spécifique à son rôle.

Vérifier que l'utilisateur peut accéder aux fonctionnalités de l'application qui lui sont autorisées.

4) Échec de l'authentification :

Si les identifiants ne sont pas valides, l'utilisateur doit rester sur la page de connexion.

Vérifier que l'utilisateur reçoit un message d'erreur approprié indiquant que les identifiants sont incorrects.

5) Déconnexion :

Une fois authentifié, vérifier que l'utilisateur peut se déconnecter en cliquant sur le lien de déconnexion dans la navbar.

Cliquer sur le lien de déconnexion.

Vérifier que l'utilisateur est redirigé vers la page de connexion après la déconnexion.

## B. Affichage des fiches frais

### Description :

Assurer le bon fonctionnement de l'affichage des fiches de frais pour l'utilisateur connecté.

### Conditions préalables :

- L'application web est accessible.
- L'utilisateur dispose d'un compte enregistré dans la base de données.
- L'utilisateur doit posséder au moins une fiche de frais déjà existante.

### Étapes :

- 1) Accéder à la page d'affichage des fiches de frais :  
Cliquez sur le menu "Mes fiches de frais" dans la navbar.  
Choisissez le mois à afficher souhaité dans le menu déroulant.
- 2) Vérifier l'authentification :

S'assurer que l'utilisateur est bien authentifié. S'il ne l'est pas, suivre le test fonctionnel "Authentification d'un utilisateur à l'application" pour s'authentifier.

S'il est authentifié s'assurer que le bon nom s'affiche en haut de la page.

3) Sélection d'une fiche de frais :

Utiliser le formulaire de sélection du mois pour choisir une fiche de frais spécifique.

Soumettre le formulaire.

4) Affichage des fiches de frais :

Sur la page d'affichage des fiches de frais, vérifier que la fiche de frais de l'utilisateur est correctement affichée.

S'assurer que les informations telles que le mois, l'état et toute autre donnée pertinente sont présentées de manière claire et lisible.

5) Vérification de l'affichage de la fiche de frais sélectionnée :

Vérifier que la fiche de frais sélectionnée est affichée avec tous ses détails.

## C. Saisie des fiches de frais

### Description :

Assurer le bon fonctionnement de la saisie des fiches de frais pour l'utilisateur connecté.

### Conditions préalables :

- L'application web est accessible.
- L'utilisateur dispose d'un compte enregistré dans la base de données.

### Étapes :

1) Accéder à la page de saisie des fiches de frais :

Cliquer sur le menu "Saisir mes frais" dans la navbar

2) Vérifier la création ou récupération de la fiche de frais :

Vérifier que la fiche de frais pour le mois en cours de l'utilisateur est correctement récupérée depuis la base de données.

Si aucune fiche de frais n'existe pour le mois en cours, vérifier que les formulaires qui permettent de la créer sont visibles.

- 3) Saisie des frais forfaitisés :  
Remplir le formulaire de saisie des frais forfaitisés avec des valeurs valides.  
Soumettre le formulaire.
- 4) Vérification de l'enregistrement des frais forfaitisés :  
Vérifier que les quantités saisies sont enregistrées dans la base de données si le formulaire de frais forfaitisés est soumis et valide.
- 5) Saisie des frais hors forfait :  
Remplir le formulaire de saisie des frais hors forfait avec des valeurs valides.  
Soumettre le formulaire.
- 6) Vérification de l'enregistrement des frais hors forfait :  
Vérifier que les informations sont également enregistrées dans la base de données si le formulaire de saisie des frais hors forfait est soumis et valide.
- 7) Vérification de la suppression d'un frais hors forfait :  
S'assurer que le tableau regroupant les frais hors forfaits saisis est bien visible.  
Supprimer un des frais hors forfait saisis.  
S'assurer qu'il n'est plus visible dans le tableau, et que si il n'y a aucun frais hors forfait qui existe le tableau n'est plus visible.

## D. Modification des fiches de frais

### Description :

Assurer le bon fonctionnement de la saisie des fiches de frais pour l'utilisateur connecté.

### Conditions préalables :

- L'application web est accessible.
- L'utilisateur dispose d'un compte enregistré dans la base de données.

### Étapes :

- 1) Accéder à la page de modification des fiches de frais :  
Cliquer sur le menu "Saisir mes frais" dans la navbar.
- 2) Modifier les frais forfaitisés :

Modifier les valeurs des frais forfaitisés dans le formulaire de modification.

Soumettre le formulaire.

- 3) Vérification de la modification des frais forfaitisés :  
Vérifier que les modifications sont enregistrées dans la base de données si le formulaire de modification des frais forfaitisés est soumis et valide.
- 4) Modifier les frais hors forfait :  
Modifier les informations des frais hors forfait dans le formulaire de modification.  
Soumettre le formulaire.
- 5) Vérification de l'enregistrement des frais hors forfait :  
Vérifier que les informations sont également enregistrées dans la base de données si le formulaire de saisie des frais hors forfait est soumis et valide.
- 6) Vérification de l'affichage des frais mis à jour :  
Cliquer sur le menu "Afficher mes fiches" dans la navbar.  
Vérifier que les modification ont bien été mises à jour dans l'affichage des détails de la fiche de frais.

## **E. Modification de l'état d'une fiche**

### **Description :**

Assurer le bon fonctionnement de la consultation des fiches de frais et de la modification de leur état par les comptables.

### **Conditions préalables :**

- L'application web est accessible.
- L'utilisateur dispose d'un compte enregistré dans la base de données.
- Des fiches de frais existent pour au moins un utilisateur dans la base de données.

### **Étapes :**

- 1) Accéder à la page de consultation des fiches de frais :  
Cliquer sur le menu "Modifier les fiches" qui se trouve dans le menu déroulant du nom de l'utilisateur connecté.  
Ce menu doit s'afficher seulement si l'utilisateur connecté dispose du rôle de comptable.

- 2) Sélectionner un utilisateur :  
Utiliser le formulaire pour sélectionner un utilisateur parmi ceux disponibles dans la base de données.  
Soumettre le formulaire.
- 3) Vérifier l'affichage des fiches de frais de l'utilisateur sélectionné :  
Sur la page, vérifier que les fiches de frais de l'utilisateur sélectionné sont correctement affichées dans un tableau.  
Vérifier que le tableau affiche les informations suivantes : le mois, l'état et un bouton permettant de consulter la fiche de frais.
- 4) Consulter une fiche de frais :  
Cliquer sur le bouton de consultation de la fiche de frais associée à un mois spécifique.  
Vérifier que le lien de redirection affiche bien le même tableau que l'affichage d'une fiche frais.
- 5) Modification de l'état d'une fiche :  
Sur la page de consultation de la fiche de frais, utiliser le sélecteur pour choisir le nouvel état de la fiche de frais parmi les options disponibles.  
Soumettre le formulaire de modification de l'état.
- 6) Vérification de la modification de l'état :  
Vérifier que l'état de la fiche de frais a été correctement modifié dans la base de données.  
Vérifier que l'utilisateur est redirigé vers la page de consultation de la fiche de frais après la modification.