
Application de Gestion de Frais de déplacement



Tables des matières

Tables des Matières	2/33
Historique du Document	3/33
Description Général du Projet	
• Le contexte du projet	4/33
• Le domaine étudié	5/33
• Le fonctionnement du système	6/33
• Le périmètre du système	8/33
• Les extensions possibles	8/33
Modélisation Conceptuelle	
• Modèle de Dictionnaire	9/33
• Modèle Entité-Association (MEA)	11/33
• Modèle Logique de données (MLD)	14/33
• Script de création de la base	15/33
Modélisation UML	
• Diagrammes des cas d'utilisations	16/33
• Diagrammes de séquence des différents cas d'utilisations	20/33
• Diagramme de classe	22/33
• Présentation de l'interface	24/33
• Présentation des principales requêtes développées	30/33

- Présentation du code développé dans l'application

32/33

Historique du document

Cette partie permettra d'identifier les différentes phases du projet ainsi que les avancements de celui-ci via des versions

Version	Date de mise	Auteur	Modification
---------	--------------	--------	--------------

	à jour		essentielles
V1.0	08/09/2024	Giovanni Miro	Description du contexte
V1.1	20/09/2024	Giovanni Miro	Modélisation de la base de données
V1.2	01/02/2025	Giovanni Miro	Développement de l'application et des principales requêtes

Description du sujet du projet

Contexte du projet

Galaxy Swiss Bourdin (GSB) est un laboratoire pharmaceutique international issu de la fusion en 2009 entre deux grands acteurs du secteur : l'américain Galaxy, spécialisé dans les maladies virales comme le SIDA et les hépatites, et l'européen Swiss Bourdin, centré sur des médicaments plus conventionnels. Ce regroupement a donné naissance à une entité majeure dans l'industrie pharmaceutique, avec un siège administratif situé à Paris, en France, et son siège social à Philadelphie, aux États-Unis.

L'industrie pharmaceutique étant un secteur lucratif mais souvent critiqué, GSB s'efforce d'optimiser ses activités tout en consolidant sa position de leader. La fusion a permis des économies d'échelle en production et en distribution, ainsi qu'une restructuration visant à combiner les forces et les produits concurrents des deux entités d'origine.

GSB déploie son influence à travers un vaste réseau de visiteurs médicaux, avec 480 en France métropolitaine et 60 dans les départements et territoires d'outre-mer, répartis en 6 secteurs géographiques (Paris-Centre, Sud, Nord, Ouest, Est, et les DOM-TOM). L'entreprise est engagée dans l'amélioration du suivi de l'activité de visite médicale en France, un marché stratégique.

GSB est aujourd'hui reconnu pour ses traitements contre les maladies virales ainsi que pour ses produits pharmaceutiques plus traditionnels, ce qui lui confère une marque forte et une présence significative sur le marché global de la santé.

Cette entreprise m'a contacter pour leur fournir une application de gestion de note de frais pour ces employés car aujourd'hui elle n'en possède aucun. Ceci permettra de faciliter la gestion de celle-ci d'un point de vue utilisateur comme d'un point de vue comptabilité.

Domaine étudié

Domaine de l'entreprise :

Le laboratoire Galaxy Swiss Bourdin (GSB) évolue dans l'industrie pharmaceutique, un secteur très lucratif caractérisé par de fréquentes fusions et acquisitions. GSB est spécialisé dans le développement de traitements pour les maladies virales, telles que le SIDA et les hépatites, ainsi que dans les médicaments conventionnels. L'entreprise cherche à optimiser sa production et sa distribution tout en naviguant dans un environnement de régulation et de critique croissante envers la transparence de l'industrie.

Domaine de développement :

Pour développer ce logiciel, je vais m'orienter vers plusieurs langages de programmation :

- HTML / CSS : Pour permettre d'avoir une interface web ainsi qu'un rendu écran agréable
- MYSQL : Pour pouvoir stocker et manipuler les diverses informations rentrées
- PHP : Pour faire la passerelle en HTML et MYSQL et ainsi manipuler les données à travers le logiciel

Pour l'édition des fiches de frais, nous utiliserons la technologie FPDF qui se trouve être la solution la plus rapide et efficace.

Il y aura aussi une gestion d'envoi de mail pour les notes de frais envoyées au comptable. Ceci demandera donc une fonctionnalité d'envoi de mail qui sera pré-remplie mais qui demandera une validation client.

Par conséquent pour que tout fonctionne correctement, l'application sera seulement développée pour le service web sur ordinateur.

Fonctionnement du système :

L'application se compose de différentes fonctionnalités :

De la gestion de note de frais	Cette fonctionnalité permettra d'insérer ces notes de frais au fur et à mesure, de les modifier ou d'en créer.
De la gestion d'historique de note de frais	Cette fonctionnalité permettra de voir les différentes notes de frais des derniers mois du point de vue utilisateurs
Validation ou non des notes de frais	Cette fonctionnalité concernera la comptabilité et les agents et permettra de validé ou non les notes de frais.
De la gestion pour la comptabilité	Cette fonctionnalité permettra à la comptabilité de posséder toute les notes de frais de ces employés pour ainsi avoir un visuel constant sur les notes de frais en cours, validé ou refusé.
De la gestion d'utilisateur	Ceci permettra de faire une gestion des utilisateurs de bloquer ou limiter certain utilisateur.
De la génération d'état des notes de frais	Cette fonctionnalité permettra de générer des pdf de ces notes de frais et de l'état de ces remboursements
De la recherche avec filtre pour les notes de frais	Vous pourrez rechercher avec un filtre les différentes notes de frais que vos avez rempli

a. Types de frais

Lors d'un déplacement, de nombreux frais seront à prévoir : ils seront gérés de manière indépendante selon la nature du frais.

Les repas seront remboursés automatiquement avec un plafond de 20 euros par jour, en France métropolitaine

Les hôtels sont gérés par un forfait à 75 euros la nuit

Les déplacements vont être gérés de plusieurs manières différentes :

Les billets d'avions vont être remboursé en totalité

Les billets de TGV seront remboursés en totalité et les billets de TER sont remboursés à hauteur de 5€ maximum par jour.

Déplacement en voiture 2 situations distinctes :

- Si le véhicule est un véhicule privé, le remboursement sera effectué en fonction de la distance effectuée .**
- Si le véhicule appartient à l'entreprise, le remboursement inclura seulement l'essence utilisé.**

Pour chaque frais devant être remboursé, la note des frais acquittées doit être fournie dans la demande de remboursement au format .jpg ou .png.

b. Etat des demandes

Les demandes pourront être classé en 4 états différents :

En attente: Après la création des demandes, elles seront mises en attente durant la période concernant cette note afin de permettre à l'utilisateur de la modifier en cas de besoin

En validation état où la comptabilité choisit d'accepter ou non le remboursement

Accepté le remboursement est accepté et l'utilisateur en est prévenu

Refusé le remboursement est refusé et l'utilisateur en est prévenu

Lorsqu'une demande est acceptée ou refusé un mail automatique sera envoyé à la personne concernée l'informant ainsi de la fin de la transaction.

Périmètre du système :

Il y aura deux parties dans ce logiciel :

Une interface Utilisateurs :

Cette interface pourra permettre au Utilisateurs de gérer leurs notes de frais, de les consulter et de voir si elles sont validés ou non. De signaler des erreurs de saisie après validation. Renseignez les différentes factures et autres dans la note de frais.

Une interface Comptable :

Cette interface permettra de faire la validation des notes de frais, de visualiser toute les notes de frais des utilisateurs. De recueillir les signalements fait pas les utilisateurs en cas d'erreur et de modifier le statue des notes.

Un interface Administrateur :

Celui-ci permettra donc d'ajouter ou supprimer des utilisateurs ou comptable et de bloquer les comptes.

Extensions :

Des extensions sont envisageablent comme l'ajout d'un fonctionnalité de statistiques pour mieux savoir ou les grosses dépenses se situent, ou l'envoi des notes de frais par mail depuis le logiciel

Modélisation Conceptuelle

Modèle de dictionnaire

Nom Conceptuel	Nom Logique	Type (E,Ca,Co)	Nature	Longueur	Identifiant	Exemple de valeur
code_employe	Code employe	E	N	INT	OUI	1,2,3
nom	Nom Employe	E	A	Varchar(50)	NON	Zaiter
prenom	Prenom employe	E	A	Varchar(50)	NON	David
num_tel	Numéro téléphone	E	AN	Varchar(15)	NON	0769372556

cp	Code Postal	E	N	INT	NON	68210
ville	Ville employé	E	A	Varchar(50)	NON	Mulhouse
cp_ville	Code postal ville	E	N	INT	OUI	68210
nom_ville	Nom ville	E	A	Varchar(50)	NON	Mulhouse
num_mission	Numéro Mission	E	N	INT	OUI	1,2,3
nom_mission	Nom Mission	E	A	Varchar(50)	NON	Voyage d'affaire
date_debut	Date de début	E	DATE	DATE	NON	12/05/2005
date_fin	Date de fin	E	DATE	DATE	NON	25/10/2015
nb_repas	Nombre de repas	E	N	INT	NON	12,15,20
destination	Destination	E	AN	Varchar(80)	NON	15 rue des Près
code_employe	Code employe	E	N	INT	NON	1,2,3
id_hebergement	Id Hébergement	E	N	INT	OUI	1,2,3
nom_hotel	Nom Hotel	E	A	Varchar(50)	NON	Hotel Ibis
num_mission	Numéro Mission	E	N	INT	NON	1,2,3
nbr_nuit_facture	Nombre nuits facturé	E	N	INT	NON	1,2,3
id_transport	ID Transport	E	N	INT	OUI	1,2,3
libelle_transport	Libelle_Transport	E	A	Varchar(50)	NON	1,2,3
nb_chevaux	Nombre chevaux	E	N	INT	NON	1,2,3
coeff_km	Coefficient KM	E	N	DECIMAL	NON	1,5
prix_total	Prix total	E	N	DECIMAL	NON	12,53
num_mission	Numéor Mission	E	N	INT	OUI	1,2,3
id_transport	ID de transport	E	N	INT	OUI	1,2,3

Modèle entiter-association (M-E-A)

Présentation du Modèle E-A

1. Entités principales :

- **Employé** : Représente les employés de l'entreprise. Chaque employé possède des attributs comme Code_employe (identifiant unique), nom, prenom, num_tel (numéro de téléphone), cp (code postal de la ville de résidence) et ville (nom de la ville de résidence).
- **Ville** : Représente les différentes villes. Les villes sont identifiées par le cp_ville (code postal unique) et le nom_ville (nom de la ville).
- **Mission** : Représente une mission d'un employé. Elle inclut des attributs comme Num_mission (numéro unique de la mission), nom_mission (nom de la mission), date_debut, date_fin et nb_repas (nombre de repas fournis pendant la mission).
- **Hébergement** : Représente les lieux où un employé peut être hébergé pendant une mission. Elle contient des informations comme Id_hebergement (identifiant unique de l'hébergement), Nom_hotel (nom de l'hôtel ou du lieu d'hébergement) et nbr_nuit_facture (nombre de nuits facturées).
- **Transport** : Représente les moyens de transport utilisés pour une mission. Elle inclut des attributs comme Id_transport (identifiant unique du transport) et Libelle_transport (type de transport utilisé, par exemple, voiture, avion).
- **Voiture** : Représente les caractéristiques des voitures utilisées comme moyen de transport. Elle contient des attributs comme Nb_chevaux (nombre de chevaux fiscaux de la voiture),

coeff_km (coefficient par kilomètre pour la facturation), et prix_total (coût total du transport).

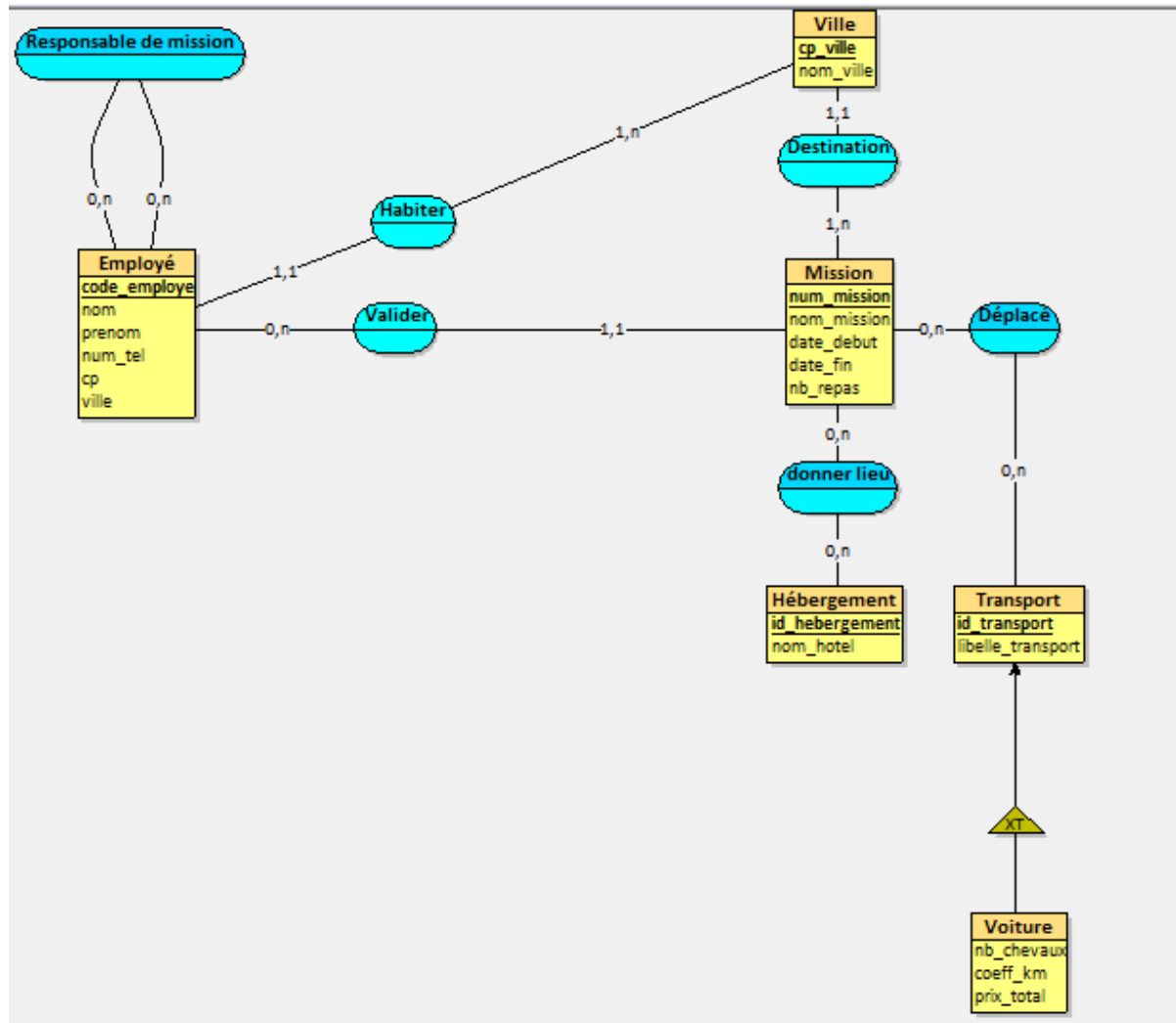
2. Relations principales :

- **Responsable de mission** : Relation entre un **Employé** et une **Mission**, indiquant qu'un employé est responsable de plusieurs missions (1,n), mais chaque mission ne peut avoir qu'un seul responsable (1,1).
- **Habiter** : Relation entre un **Employé** et une **Ville**. Un employé réside dans une ville spécifique (1,n pour Ville -> Employé), et chaque employé peut habiter dans une seule ville (1,1 pour Employé -> Ville).
- **Destination** : Relation entre une **Mission** et une **Ville**. Chaque mission se déroule dans une ville (1,1 pour Mission -> Ville), mais plusieurs missions peuvent avoir lieu dans la même ville (1,n pour Ville -> Mission).
- **Donner lieu** : Relation entre une **Mission** et un **Hébergement**, indiquant qu'une mission peut inclure plusieurs nuits d'hébergement facturées (0,n), mais un hébergement peut être associé à plusieurs missions (0,n).
- **Déplacer** : Relation entre une **Mission** et un **Transport**, montrant qu'un moyen de transport est utilisé pour une mission (0,n). Un transport peut être utilisé pour plusieurs missions (0,n).

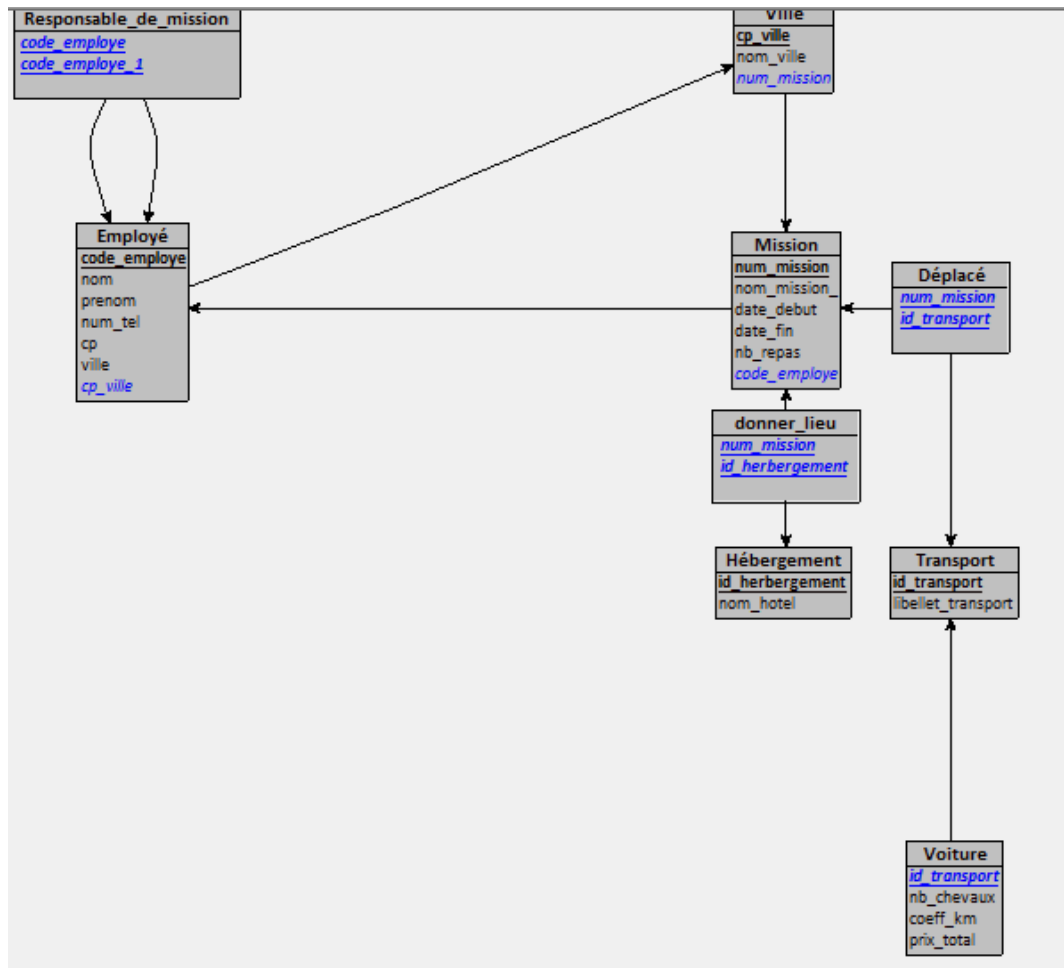
3. Choix de Modélisation :

1. **Modélisation des employés et des villes** : L'entité **Employé** est bien définie avec des attributs nécessaires à la gestion des employés, tels que les informations personnelles (nom, prénom, téléphone) et la relation avec la **Ville** où ils résident. Cela permet une bonne gestion des employés dans différentes villes.
2. **Séparation des missions et des hébergements** : Le fait de séparer les entités **Mission** et **Hébergement** permet de modéliser efficacement la possibilité d'avoir plusieurs hébergements associés à une mission, ou inversement, plusieurs missions liées à un même lieu d'hébergement.
3. **Modélisation des transports** : L'entité **Transport** permet de définir les moyens de transport utilisés pour les missions. La sous-catégorie **Voiture** permet d'ajouter des détails spécifiques au transport en voiture, comme le calcul du coût total en fonction du nombre de chevaux et des kilomètres parcourus.
4. **Contraintes sur les relations** :
 - Chaque **Employé** peut être responsable de plusieurs missions, illustré par la relation Responsable de mission qui a une cardinalité 1,n entre Employé et Mission.
 - Une **Mission** peut avoir plusieurs hébergements, illustré par la cardinalité 0,n dans la relation Donner lieu.

- La relation **Déplacer** modélise les déplacements en liant chaque **Mission** avec les différents moyens de **Transport** utilisés, permettant une flexibilité dans le choix des moyens de transport pour chaque mission.



Modèle Logique de donnée (MLD)



Script de création de la base

```

CREATE DATABASE GestionMissions;
USE GestionMissions;

-- Table : Ville
CREATE TABLE Ville (
  cp_ville VARCHAR(10) PRIMARY KEY,

```

```

    nom_ville VARCHAR(50) NOT NULL
);

-- Table : Employé
CREATE TABLE Employe (
    Code_employe INT PRIMARY KEY AUTO_INCREMENT,
    nom VARCHAR(50) NOT NULL,
    prenom VARCHAR(50) NOT NULL,
    num_tel VARCHAR(15),
    cp_ville VARCHAR(10),
    FOREIGN KEY (cp_ville) REFERENCES Ville(cp_ville)
);

-- Table : Mission
CREATE TABLE Mission (
    Num_mission INT PRIMARY KEY AUTO_INCREMENT,
    nom_mission VARCHAR(100) NOT NULL,
    date_debut DATE NOT NULL,
    date_fin DATE NOT NULL,
    nb_repas INT,
    destination VARCHAR(10),
    Code_employe INT,
    FOREIGN KEY (destination) REFERENCES Ville(cp_ville),
    FOREIGN KEY (Code_employe) REFERENCES Employe(Code_employe)
);

CREATE TABLE Hebergement (
    Id_hebergement INT PRIMARY KEY AUTO_INCREMENT,
    Nom_hotel VARCHAR(100) NOT NULL
);

CREATE TABLE Mission_Hebergement (
    Num_mission INT,
    Id_hebergement INT,
    nbr_nuit_facture INT,
    PRIMARY KEY (Num_mission, Id_hebergement),
    FOREIGN KEY (Num_mission) REFERENCES Mission(Num_mission),
    FOREIGN KEY (Id_hebergement) REFERENCES Hebergement(Id_hebergement)
);

-- Table : Transport
CREATE TABLE Transport (
    Id_transport INT PRIMARY KEY AUTO_INCREMENT,
    Libelle_transport VARCHAR(50) NOT NULL
);

CREATE TABLE Voiture (
    Id_transport INT PRIMARY KEY,
    Nb_chevaux INT,
    coeff_km DECIMAL(5,2),
    prix_total DECIMAL(10,2),
    FOREIGN KEY (Id_transport) REFERENCES Transport(Id_transport)
);

```

```
);

CREATE TABLE Mission_Transport (
    Num_mission INT,
    Id_transport INT,
    PRIMARY KEY (Num_mission, Id_transport),
    FOREIGN KEY (Num_mission) REFERENCES Mission(Num_mission),
    FOREIGN KEY (Id_transport) REFERENCES Transport(Id_transport)
);
```

Diagrammes des cas d'utilisations

1. Acteur : Employé

Cas d'utilisation :

- **Se connecter** : L'employé entre ses identifiants pour accéder au système.
- **Consulter ses informations personnelles** : L'employé peut visualiser ses informations personnelles telles que son nom, son numéro de téléphone et sa ville de résidence.
- **Demander une modification de ses informations** : L'employé peut soumettre une demande pour modifier ses informations personnelles, telles que le numéro de téléphone ou la ville de résidence.
- **Consulter ses missions** : L'employé peut voir la liste des missions dont il est responsable ou auxquelles il participe.
- **Consulter les détails d'une mission** : L'employé peut consulter les informations détaillées d'une mission, telles que la date de début et de fin, la ville de destination, les moyens de transport et les hébergements associés.
- **Recevoir une notification** : L'employé reçoit des notifications (par email ou dans le système) sur les missions qui lui sont assignées, ou sur les changements concernant ces missions (dates, transports, hébergements, etc.).

2. Acteur : Gestionnaire

Cas d'utilisation :

- **Se connecter** : Le gestionnaire entre ses identifiants pour accéder au système.
- **Consulter les missions en cours** : Le gestionnaire accède à une vue d'ensemble des missions en cours avec les employés responsables et les villes de destination.
- **Créer une nouvelle mission** : Le gestionnaire peut créer une nouvelle mission en spécifiant le nom, la date de début et de fin, la ville de destination et l'employé responsable.
- **Assigner un employé responsable à une mission** : Le gestionnaire peut affecter un employé en tant que responsable d'une mission.
- **Ajouter/modifier les transports pour une mission** : Le gestionnaire choisit et attribue un ou plusieurs moyens de transport pour une mission spécifique (voiture, avion, etc.).

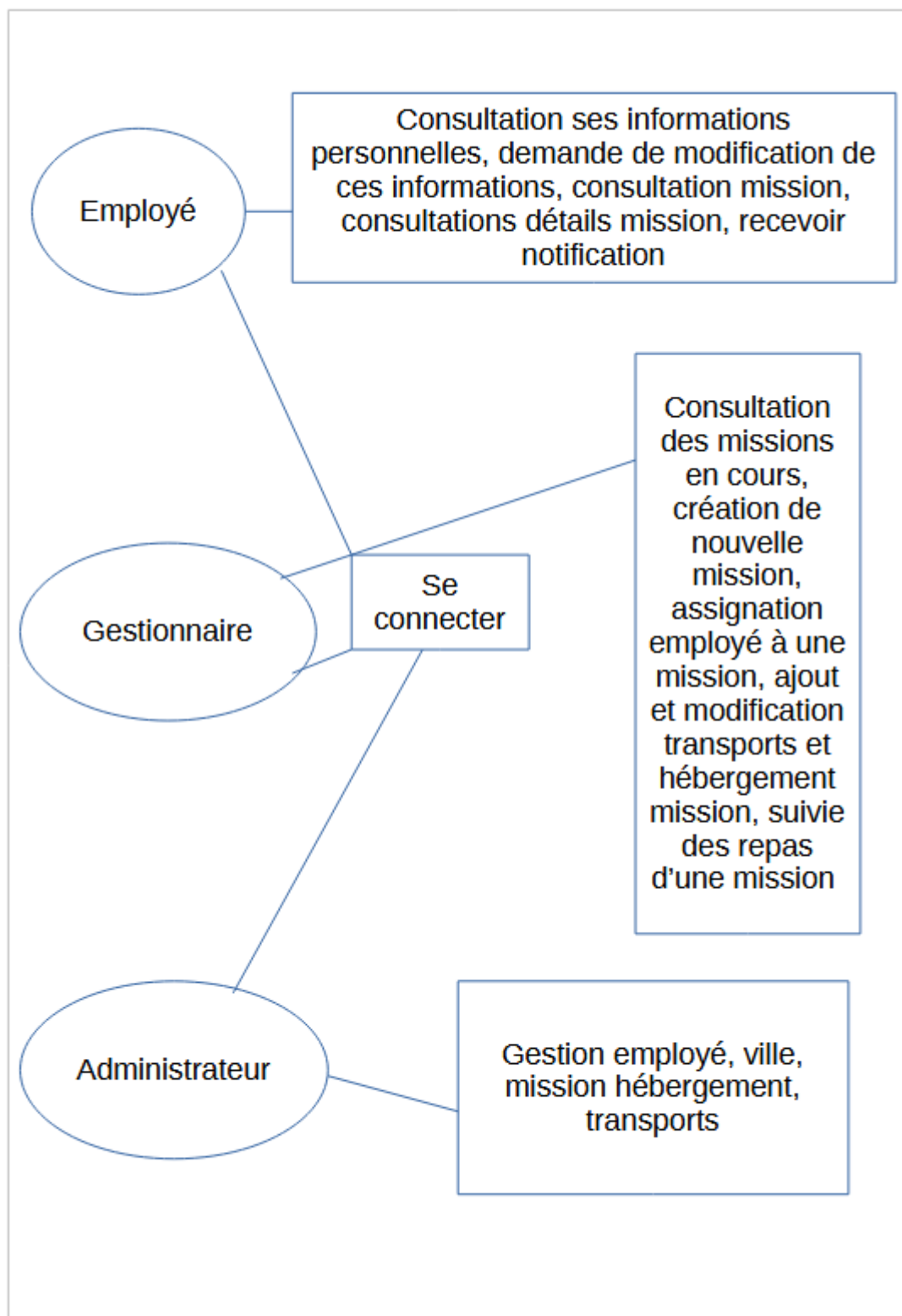
- **Ajouter/modifier des hébergements pour une mission** : Le gestionnaire sélectionne les hébergements nécessaires pour la mission et indique le nombre de nuits facturées.
- **Modifier une mission en cours** : Le gestionnaire peut mettre à jour les dates de début ou de fin d'une mission, la ville de destination, ou d'autres détails.
- **Suivre les hébergements et repas d'une mission** : Le gestionnaire consulte et ajuste les informations relatives aux repas fournis et aux nuits d'hébergement facturées pour une mission donnée.

3. Acteur : Administrateur

Cas d'utilisation :

- **Se connecter** : L'administrateur entre ses identifiants pour accéder au système.
- **Gérer les employés** :
 - **Ajouter un employé** : L'administrateur crée un nouveau profil d'employé avec les informations nécessaires (nom, prénom, numéro de téléphone, etc.).
 - **Modifier un employé** : L'administrateur peut mettre à jour les informations d'un employé.
 - **Supprimer un employé** : L'administrateur peut retirer un employé du système si celui-ci quitte l'entreprise.
- **Gérer les villes** :
 - **Ajouter une ville** : L'administrateur ajoute une nouvelle ville en spécifiant son nom et son code postal.
 - **Modifier une ville** : L'administrateur peut mettre à jour les informations d'une ville existante (nom, code postal).
 - **Supprimer une ville** : L'administrateur retire une ville qui n'est plus utilisée dans les missions.
- **Gérer les missions** :
 - **Créer/modifier une mission** : L'administrateur peut créer une mission ou modifier les détails d'une mission (nom, date de début/fin, ville de destination).
 - **Supprimer une mission** : L'administrateur peut supprimer une mission obsolète ou annulée.
- **Gérer les hébergements** :
 - **Ajouter un hébergement** : L'administrateur enregistre un nouvel hébergement (nom de l'hôtel ou du lieu d'hébergement, nombre de nuits facturées).
 - **Modifier un hébergement** : L'administrateur peut mettre à jour les informations liées à un hébergement.
 - **Supprimer un hébergement** : L'administrateur supprime un hébergement qui n'est plus utilisé.
- **Gérer les transports** :

- **Ajouter/modifier un moyen de transport** : L'administrateur ajoute ou modifie un moyen de transport (par exemple, avion, voiture).
- **Ajouter/modifier une voiture** : Si une voiture est ajoutée comme moyen de transport, l'administrateur peut spécifier des détails comme le nombre de chevaux fiscaux et le coefficient kilométrique pour le calcul des coûts.
- **Supprimer un transport** : L'administrateur supprime un transport qui n'est plus utilisé.



Diagrammes de séquence des différents cas d'utilisations

Pour chaque cas d'utilisation, un **diagramme de séquence** représente les interactions entre les différents objets et acteurs dans un ordre temporel. Voici des exemples de **diagrammes de séquence** pour certains des cas d'utilisation principaux du système de gestion des notes de frais.

Cas d'utilisation : Se connecter

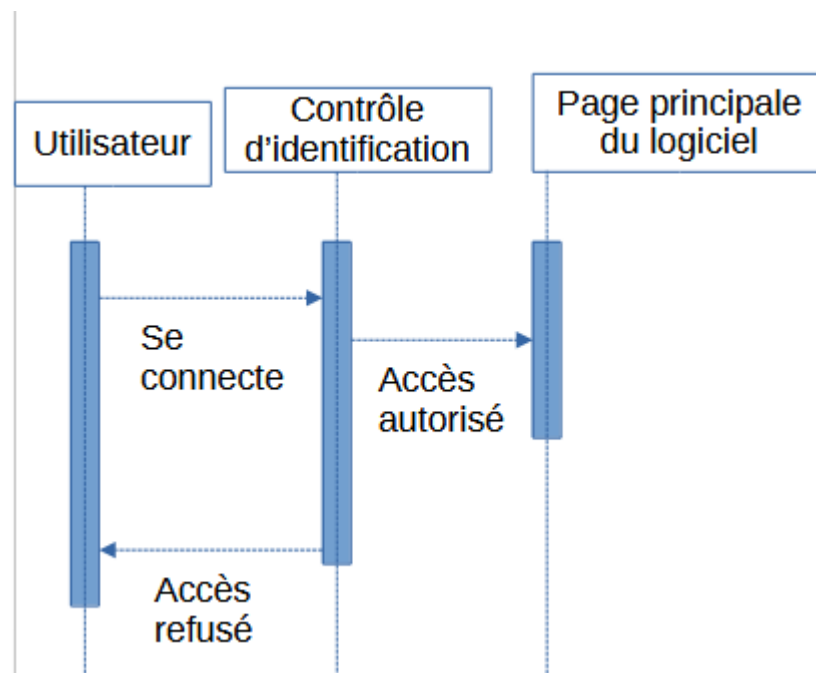
Description :

L'utilisateur entre ses identifiants pour accéder au système.

Diagramme de séquence :

Acteur (Utilisateur) : L'utilisateur entre ses identifiants et demande à se connecter.

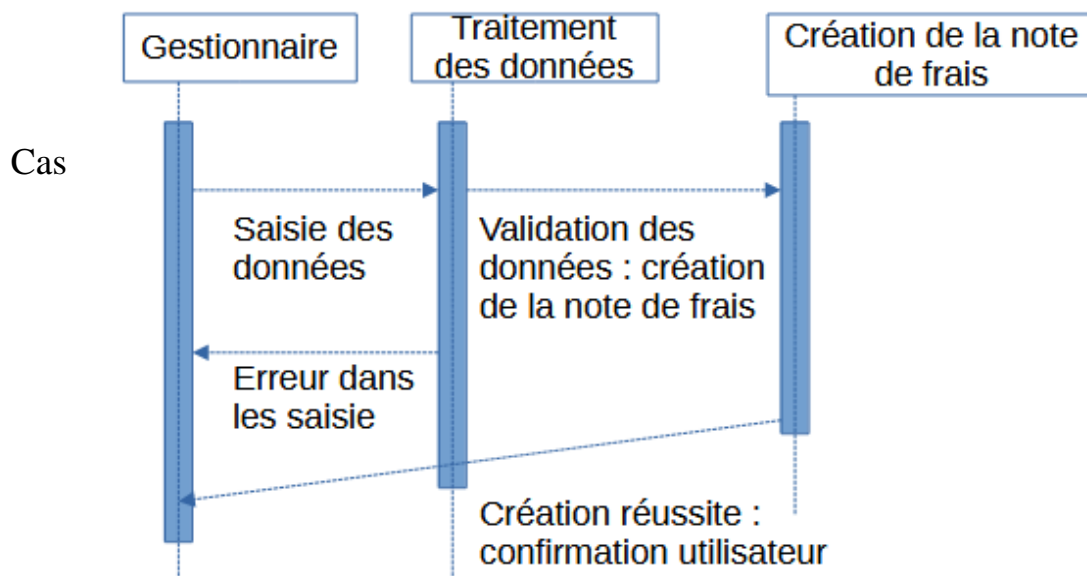
1. **Interface utilisateur (UI) :** L'interface utilisateur transmet les identifiants au système d'authentification pour vérification.
2. **Système d'authentification :** Le système d'authentification vérifie les informations d'identification fournies (nom d'utilisateur et mot de passe) et renvoie un statut de succès ou d'échec.



Cas d'utilisation : Création note de frais

Diagramme de séquence :

1. **Gestionnaire** : Le gestionnaire crée une nouvelle note de frais et saisit les détails nécessaires, tels que le type de transport, la date, le nombre de nuité.
2. **Interface Gestionnaire (UG)** : L'interface Gestionnaire transmet les informations saisies au système de gestion des notes de frais.
3. **Système de gestion des notes de frais** : Le système de gestion enregistre la note de frais avec les informations fournies dans la base de données.
4. **Système de gestion des notes de frais** : Le système renvoie une confirmation Gestionnaire indiquant que la note de frais a été enregistrée avec succès et envoie une confirmation à l'utilisateur de la création d'une note de frais



d'utilisation : Gérer les utilisateurs (Administrateur)

Description :

L'administrateur ajouté des comptes utilisateurs.

Diagramme de séquence :

- **Administrateur** : Demande de créer ou modifier un utilisateur.
- **Interface Administrateur (UA)** : Envoie les informations au système de gestion des utilisateurs.
- **Système de gestion des utilisateurs** : Enregistre ou modifie les informations dans la base de données.
- **Base de données** : Confirme que l'opération a été effectuée.
- **Système de gestion des utilisateurs** : Transmet une confirmation à l'interface.
- **Interface utilisateur (UI)** : Affiche la confirmation à l'administrateur.

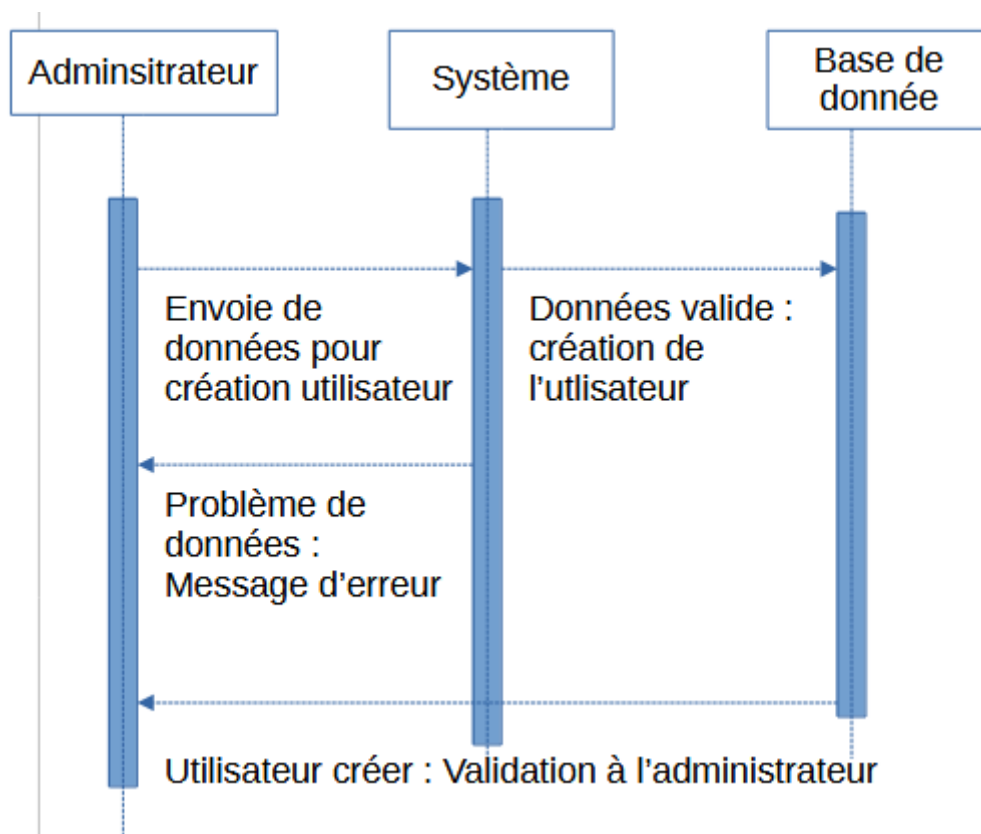
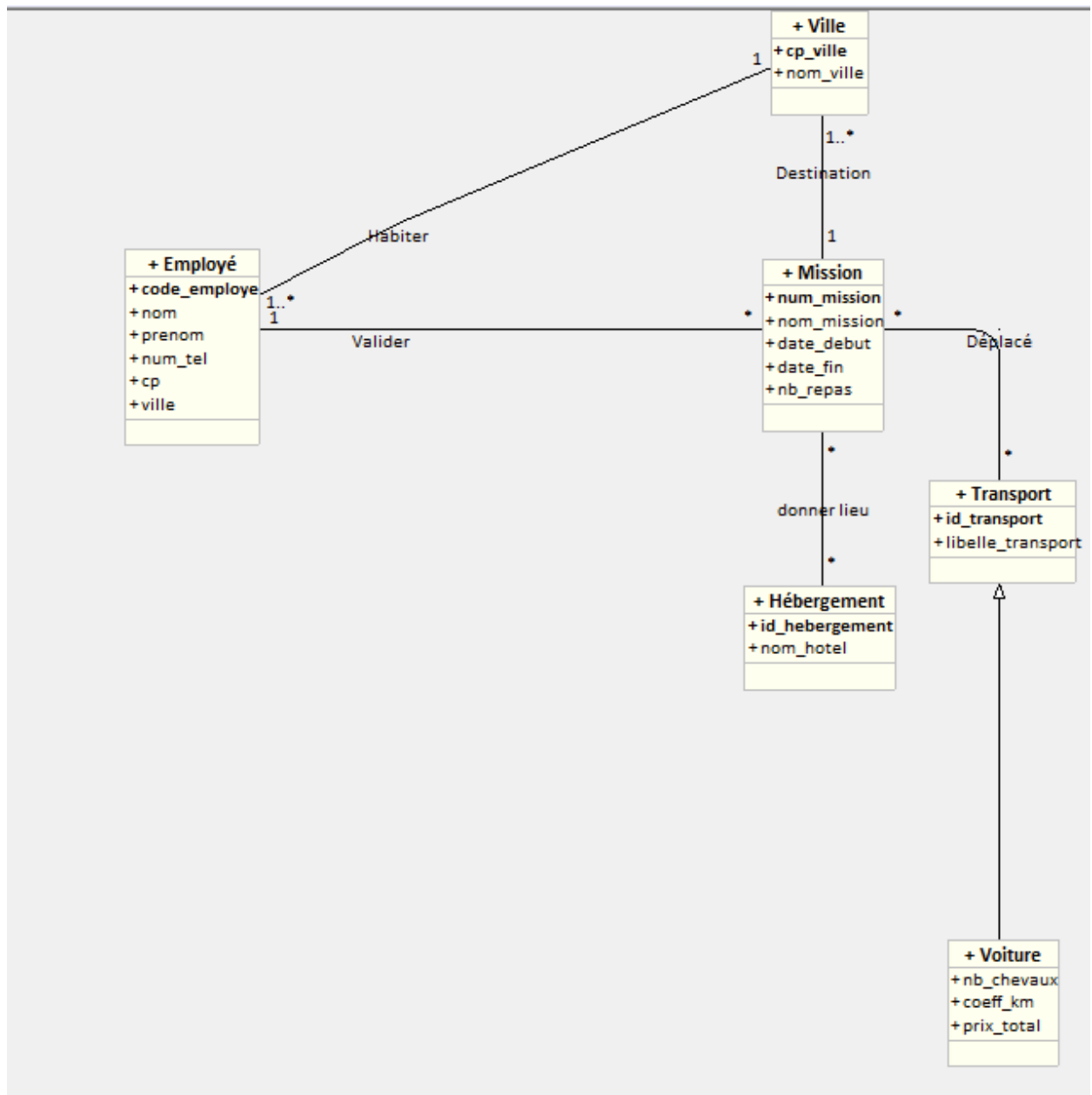


Diagramme de classe



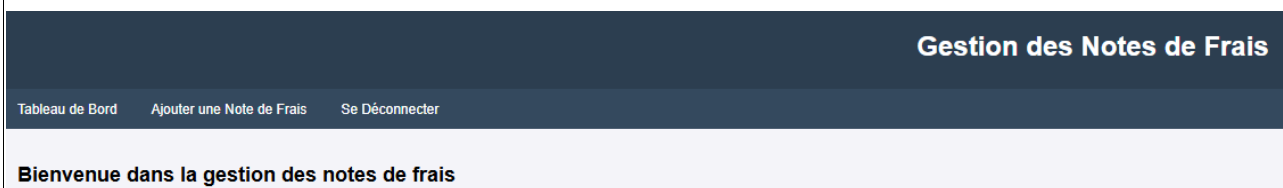
Présentation de l'interface :

A. Le Menu Général

Tout d'abord le menu de connexion qui nous permet de nous connecter à l'application avec un rôle qui est défini au sein de la base de données.



Ensuite, nous avons le menu principal qui permet de nous orienter vers plusieurs pages :



Nous pouvons accéder au tableau de bord ou tout se passera, nous avons aussi ajouter une note de frais qui nous permettra de créer une nouvelle note de frais, puis un menu qui nous permettra de nous déconnecter.

Détails de la Mission Seminaire Guadeloupe

Tableau de Bord

Ajouter une Note de Frais

Se Déconnecter

Note de frais suivante

Note de frais précédente

Informations sur la Mission

Détails

Nom de la Mission : Seminaire Guadeloupe

Employé : Miro Giovanni

Lieu : Burnhaupt 68210

Date de Début : 13/01/2025

Date de Fin : 18/01/2025

Nombres de repas : 0

Nombres de nuits : 2

Hébergement : Les 3 Mouflons

Déplacement : Tram Renault Megane 2016

Ajouter des frais

Supprimer des frais

Confirmer la note de frais

Justificatif 1 : Choisir un justificatif

Justificatif 2 : Choisir un justificatif

Hébergement

Hôtel : Les 3 Mouflons

Coût : 150 €

Repas

Nombre de Repas : 0

Coût Total : 0 €

Déplacement

Nombre de Déplacement: 2

Coût : 39.32 €

Status

En Cours

Total

Coût Total Mission : 189.32 €

© 2024 Gestion des Notes de Frais. Tous droits réservés.

Le tableau de bord permet de centraliser toute les informations nécessaire et essentiel au premier regard. Il y a tout d’abord les détails qui sont en haut à gauche :

Détails

Nom de la Mission : Seminaire Guadeloupe

Employé : Miro Giovanni

Lieu : Burnhaupt 68210

Date de Début : 13/01/2025

Date de Fin : 18/01/2025

Nombres de repas : 0

Nombres de nuits : 2

Hébergement : Les 3 Mouflons

Déplacement : Tram Renault Megane 2016

Il y a ensuite les différents boutons, ajouter frais, supprimer frais et Confirmer la note qui permette de manipuler la note de frais :

Ajouter des frais

Supprimer des frais

Confirmer la note de frais

L'on peut aussi consulter les justificatifs mis à tout moment :

Informations sur la Mission

Détails

Nom de la Mission : Seminaire Guadeloupe

Employé : Miro Giovanni

Lieu : Burnhaupt 68210

Date de Début : 13/01/2025

Date de Fin : 18/01/2025

Nombres de repas : 0

Nombres de nuits : 5

Hébergement : Les 3 Mouflons

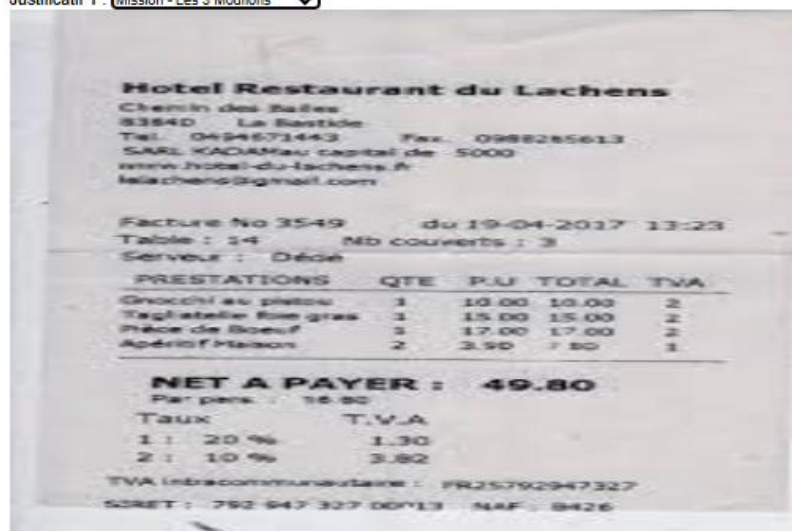
Déplacement : Tram Renault Megane 2016

Ajouter des frais

Supprimer des frais

Confirmer la note de frais

Justificatif 1 : Mission - Les 3 Mouflons



En pour finir, le montant total de chaque catégorie, le statut, ainsi que le montant total :

Hébergement

Hôtel : Les 3 Mouflons

Coût : 375 €

Repas

Nombre de Repas : 0

Coût Total : 0 €

Déplacement

Nombre de
Déplacement: 2

Coût : 39.32 €

Status

En-Cours

Total

Coût Total Mission :
414.32 €

B. Les différents formulaires

Lorsque l'on clique sur Ajouter des frais, l'on appelle une nouvelle page qui va permettre de rentrer l'hébergement, les déplacements et le nombre de repas :

Saisir Hébergement Seminaire Guadeloupe	Saisir Déplacement Seminaire Guadeloupe	Saisir Repas Seminaire Guadeloupe
Nom de l'Hôtel <input type="text"/>	Type de Transport <input type="text" value="Transport"/>	Nombres de repas <input type="text"/>
Nombres de nuits <input type="text"/>	Voiture <input type="text" value="Voiture"/>	<input type="button" value="Enregistrer Repas"/>
Ajouter un justificatif <input type="button" value="Choisir un fichier"/> Aucun fichier choisi	Ajouter un justificatif <input type="button" value="Choisir un fichier"/> Aucun fichier choisi	
<input type="button" value="Enregistrer Hébergement"/>	<input type="button" value="Enregistrer Déplacement"/>	

Et chaque bouton enregistrer permettra de valider et de mettre à jour la base de données avec les données saisies.

Il y a ensuite la suppression des frais avec le bouton supprimer des frais qui va permettre de supprimer tout les frais que l'on a pu rajouter dessus :

Suppression hébergement Seminaire Guadeloupe

Nom de l'hébergement: Les 3 Mouflons
Nombres de nuits: 5

Suppression repas Seminaire Guadeloupe

Nombres de repas: 4

Suppression déplacement Seminaire Guadeloupe

Déplacement 9: Tram 5 euros

Déplacement 11: Renault Megane 2016 34.32 euros

Et pour finir le dernier bouton permet de passer le status de la note de frais à A-Valider pour que la comptabilité puisse l'accepter ou la rejeter.

Mon Espace

Tableau de Bord

Ajouter une Note de Frais

Se Déconnecter

Nom : Miro

Prénom : Giovanni

Ville : MONACO

Code Postal : 98000

Email : giovanni6846@gmail.com

Numéro de téléphone : 0769372349

Rôle : 3

Modification des données

Ville

MONACO

Email

giovanni6846@gmail.com

Numéro de téléphone

0769372349

Modifier les informations

L'utilisateur peut aussi gérer ces données personnelles en les modifiant à tout moment via le menu Mon Espace.

Côté administrateur, l'on peut gérer tous les utilisateurs et l'on peut créer des utilisateurs :

Informations des utilisateurs

Id	Nom	Prénom	Ville	Mot de passe	Email	Numéro de téléphone	Rôle	
1	Miro	Giovanni	MONACO	0000	giovanni6846@gmail.com	0769372349	3	Modifier
3	Comptable1	Comptable1	Burnhaupt	0000	comptable1@gmail.com	0665457856	2	Modifier
4	Administrateur	Administrateur		0000			1	Modifier
8	Utilisateur1	User	ARTEMARE	User	utilisateur1@gmail.com	0788985867	3	Modifier
9	Utilisateur2	User	CHALAMONT	User	utilisateur2@gmail.com	0612563548	3	Modifier
10	Utilisateur3	User	MONACO	User3	hobq19650@alienter.com	0623658936	3	Modifier
12	Bianca	Liz	MULHOUSE	0000	karen@gmail.fr	0769372349	3	Modifier
13	Deeraj	Paupiah	MONTPELLIER	0000	pevayir628@almaxen.com	0612563548	3	Modifier

Création d'utilisateur

Nom	Prénom	Mot de passe	Rôle	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Créer

Et pour finir côté comptabilité, le comptable peut valider ou non une note de frais via deux boutons présents sur la page du tableau de bord comptabilité.

Informations sur la Mission

Détails

Nom de la Mission : Migration client

Employé : Miro Giovanni

Lieu : RENNES 35000

Date de Début : 03/02/2025

Date de Fin : 07/02/2025

Nombres de repas : 21

Nombres de nuits : 5

Transports : Les 3 Mouflons

Déplacement : Renault Megane 2016

Accepter

Rejeter

Impression PDF

Hébergement

Hôtel : Les 3 Mouflons

Coût : 375 €

Repas

Nombre de Repas : 21

Coût Total : 420 €

Transport

Nombre de transports: 1

Coût : 330 €

Statut

A-Validé

Total

Coût Total Mission : 1125 €

Et l'employé sera informé par mail de la validation ou le rejet de sa note de frais et pourra la visualiser dans son espace.

Présentation des principales requêtes développées :

Le projet a été développé en orienté objet avec l'utilisation d'un ORM (Doctrine), j'ai donc recréé chaque table de ma base de données dans une classe pour pouvoir manipuler les objets ensemble. Dès que le mapping est fait, l'on fait des recherches dans la base à l'aide de différents critères :

```
public static function déplacement($mission)
{
    global $entityManager;
    $déplacement = $entityManager->getRepository( className: Deplacer::class)->findBy([
        'mission' => $mission
    ]);
    if ($déplacement == null) {
        return False;
    } else {
        return $déplacement;
    }
}
```

Ici, l'on va chercher tous les enregistrements de la table déplacement qui ont comme numéro de mission \$mission qui vaut l'id de la mission qui à appeler cette fonction.

Voici par exemple la requête de connexion pour s'identifier :

```
public static function connexion($pseudo, $mdp)
{
    global $entityManager;
    $identifiant = $entityManager->getRepository( className: Employe::class)->findOneBy(['prenom' => $pseudo, 'mdp' => $mdp]);
    if ($identifiant == NULL)
    {
        return false;
    }
    else
    {
        $_SESSION['id'] = $identifiant->getId();
        $_SESSION['typeConnexionBack'] = "connecter";
        $_SESSION['role'] = $identifiant->getRole();
        return true;
    }
}
```

Ou alors la requête pour chercher la note de frais suivante que ça soit pour l'utilisateur lambda ou la comptabilité :

```
public static function mission_($idMission, $role)
{
    global $entityManager;
    if ($role == 2){
        $findMission = $entityManager->getRepository('Mission::class')->findBy(['status' => "A-Valider"]);
        if ($findMission == null) {
            return False;
        } else {
            $mission_s = null;
            foreach ($findMission as $mission) {
                if ($mission->getId() > $idMission) {
                    $mission_s = $mission;
                    break;
                }
            }
            return $mission_s;
        }
    } else {
        $findMission = $entityManager->getRepository('Mission::class')->findBy(['employee' => $_SESSION['id']]);
        if ($findMission == null) {
            return False;
        } else {
            $mission_s = null;
            foreach ($findMission as $mission) {
                if ($mission->getId() > $idMission) {
                    $mission_s = $mission;
                    break;
                }
            }
            return $mission_s;
        }
    }
}
```

Présentation du code développé dans l'application :

Pour établir le lien avec la base de données, l'on utilise le fichier Bootstrap.PHP qui lui va s'occuper de créer la connexion avec la base :

```
// the connection configuration
$dbParams = [
    'host' => '127.0.0.1',
    'driver' => 'pdo_mysql',
    'user' => 'root',
    'password' => '',
    'dbname' => 'gestionmissions',
];

$config = new Configuration();
$config->setProxyDir( dir: __DIR__ . '/proxies'); // Répertoire pour les fichiers proxy
$config->setProxyNamespace( ns: 'MyApp\Proxies');
$config->setAutoGenerateProxyClasses( autoGenerate: true);

$config = ORMSetup::createAttributeMetadataConfiguration($paths, $isDevMode);
$connection = DriverManager::getConnection($dbParams, $config);
$entityManager = new EntityManager($connection, $config);
```

\$dbParams contient tous les paramètres pour se connecter à la base de données

\$connection va lancer la liaison entre la base de données et l'utilisateur.

Et \$entityManager lui va retourner l'objet de connexion pour que l'on puisse faire nos requêtes.

Cela fonctionne sur le même principe qu'un singleton car nous créons un objet (new EntityManager) et il restera actif pour toutes les requêtes diverses et variées.

Un autre script intéressant serait de voir comment toutes les informations remontent dans l'affichage de la page web.

Nous avons les contrôleurs et les programmes vue précédemment qui effectuent des requêtes et qui retournent un array ou un objet directement.

Nous allons donc envoyer l'array ou l'objet à la page html :

```
$justificatif = \App\Modele\Modele_Mission::mission_justificatif($mission->getId());
$Vue->setEntete(new \App\Vue\Vue_Structure_Entete_TDB());
$Vue->addToCorps(new \App\Vue\Vue_Tableau_de_Bord($mission,$justificatif));
```

De ce code-là, nous avons ensuite créer un constructeur d'objet spécifique pour la page pour que l'on puisse manipuler les objets comme l'on veut dans l'affichage.

Par exemple, pour les détails de la note de frais, cela va s'écrire comme ça :

```
<h3>Détails</h3>
<p><strong>Nom de la Mission :</strong> <?= htmlspecialchars($this->mission->getNomMission()) ?></p>
<p><strong>Employé :</strong> <?= htmlspecialchars($this->mission->getEmploye()->getNom()) ?>
    <?= htmlspecialchars($this->mission->getEmploye()->getPrenom()) ?></p>
<p><strong>Lieu :</strong> <?= htmlspecialchars($this->mission->getVille()->getNomVille()) ?>
    <?= htmlspecialchars($this->mission->getVille()->getCpVille()) ?></p>
<p><strong>Date de Début :</strong> <?= htmlspecialchars($this->mission->getDateDebut()->format( format: 'd/m/Y')) ?></p>
<p><strong>Date de Fin :</strong> <?= htmlspecialchars($this->mission->getDateFin()->format( format: 'd/m/Y')) ?></p>
<p><strong>Nombres de repas :</strong> <?= htmlspecialchars($this->mission->getNbRepas()) ?></p>
<p><strong>Nombres de nuits :</strong> <?= htmlspecialchars($this->mission->getNbNuit()) ?></p>
<p><strong>Hébergement :</strong> <?php if ($this->mission->getHebergement() == Null){
    echo "";
} else {
    echo(htmlspecialchars($this->mission->getHebergement()->getNomHotel()));
} ?></p>
<p><strong>Déplacement :</strong>
    <?php
    $content = '';
    foreach ($this->mission->getDeplacement() as $deplacer) {
        $content .= htmlspecialchars($deplacer->getTransport()->getLibelleTransport()) . ' ';
    }
    echo $content;
    ?>
</p>
.v>
```

Avec le \$this->mission qui va être porteur du de notre objet mission et ensuite ->getEmploye pour appeler l'employeur liée à la mission. Mais comme dans la définition des classes expliquer précédemment ; getEmployer va nous permettre d'accéder à toutes les informations de la table employer qui concerne l'employeur liée à la mission !