

Rapport Projet Tutoré

Nom de code :

G.A.P.M.

Étudiants ayant réalisé le projet :

- Louis LUBINEAU
- Frédéric DABADIE
- Joël HERRANZ
- Mathieu MORGAT
- Christopher DENT
- Valentin CHRÉTIEN
- Enolah DAIJARDIN
- Taisong MA

Promotion
Licence Pro – PRISM
2022/2023

Sommaire

Contexte	3
Les fonctionnalités implémentées	4
Visiteur	4
Demandeur de services	6
Prestataire de services	13
Administrateur	24
Répartition des rôles	29
Gantt	30
Prototype – Maquette (IHM)	31
Outils technologiques utilisés	32
Arborescence	33
Architecture utilisée	34
Modélisation – Diagrammes	41
Cas d'utilisation (Use case)	41
Modèle Conceptuel de Données (MCD)	42
Diagramme de classe (UML)	43
Modèle Logique de Données (MLD)	44
Explications de codes	45
Quelques images d'intégration	48
Difficultés rencontrés	51
Bilan	52
Perspectives futures	54
Annexes	55
Définitions	55
Scénarios	57

Contexte

Une petite entreprise souhaite acquérir une application web qui lui permettra une gestion autonome des divers secteurs de son activité.

Cette entreprise se positionne dans le secteur des services à la personne.

Elle met en lien :

- Des intervenants (aide-soignants, kiné, infirmières, éducateurs, aide de vie, etc) qui peuvent être considérés comme des « prestataires de services », se rendant selon un planning au domicile des personnes en demande de services ;
- Des personnes physiques ou virtuelles (personnes en demande d'assistance, structures qui hébergent des personnes dépendantes) qui peuvent être considérés comme des « demandeurs de services » ;
- Des personnels administratifs qui gèrent les différents intervenants ;
- Un administrateur qui assure la cohésion de l'ensemble.

Les fonctionnalités implémentées

Détails de chaque interactions possibles en tant que visiteur, demandeur, intervenant et personnel administratif. Description brève du site et de ce que les utilisateurs peuvent faire.

Visiteur

Inscription

L'objectif de cette fonctionnalité est de permettre aux nouveaux utilisateurs de s'inscrire sur la plateforme afin d'accéder à toutes les fonctionnalités et services offerts, en spécifiant au moment de l'inscription son rôle (demandeur ou prestataire de services).

Requis :

1. Formulaire d'inscription : Un formulaire convivial et intuitif doit être mis en place pour recueillir les informations nécessaires à l'inscription. Les champs utilisés doivent répondre à des besoins spécifiques pour le fonctionnement de l'application, afin d'être en conformité avec le RGPD.
2. Politique de confidentialité et conditions d'utilisation : L'utilisateur doit être informé des politiques de confidentialité et des conditions d'utilisation de la plateforme ou de l'application et doit donner son consentement avant de procéder à l'inscription.

3. Gestion des erreurs : Des messages d'erreur clairs et des instructions appropriées doivent être fournis en cas de saisie incorrecte ou d'informations manquantes afin d'assister l'utilisateur lors du processus d'inscription.

4. Gestion des doublons : Un mécanisme de vérification doit être mis en place pour éviter la création de comptes en double. Dans GAPM, cette vérification est restreinte au mail. Cela peut inclure la vérification de l'unicité de l'adresse e-mail ou d'autres informations uniques fournies par l'utilisateur.

5. Authentification sécurisée : Les informations d'identification de l'utilisateur doivent être stockées de manière sécurisée en utilisant des techniques de hachage et de cryptage appropriées pour protéger la confidentialité des données.

6. Profil utilisateur : Une fois l'inscription réussie, un profil utilisateur doit être créé et associé à l'utilisateur.

Demandeur de services

Recherche d'un prestataire de services

La fonctionnalité de recherche d'un prestataire de services vise à faciliter la recherche et la sélection d'un prestataire qualifié pour répondre aux besoins spécifiques de l'utilisateur. Elle permettra aux utilisateurs de trouver des prestataires de services en offrant un service spécifique, offrant ses services dans une ville concrète et/ou ayant un nom déterminé.

Requis :

1. Base de données : la fonctionnalité devra s'appuyer sur une base de données solide pour stocker les informations relatives aux prestataires de services, y compris les profils, les évaluations, les commentaires, etc.

Prendre un rendez-vous

La fonctionnalité pour prendre un rendez-vous vise à permettre aux utilisateurs de planifier et de gérer facilement des rendez-vous avec des professionnels, des services ou des événements. Elle offre une solution pratique pour la réservation de créneaux horaires, en optimisant l'organisation et en évitant les conflits d'horaires.

Requis :

1. Calendrier et disponibilité : un système de gestion des calendriers doit être mis en place pour permettre aux prestataires de services de définir

leurs disponibilités et de les rendre accessibles aux utilisateurs. Le calendrier devrait permettre aux utilisateurs de voir les disponibilités en temps réel, d'identifier les créneaux libres et de choisir celui qui correspond le mieux à leurs besoins.

2. Base de données : la fonctionnalité devra s'appuyer sur une base de données pour stocker les informations relatives aux rendez-vous, aux utilisateurs et aux prestataires de services.

Consulter l'historique des rendez-vous

La fonctionnalité pour la consultation de l'historique des rendez-vous, vise à permettre aux utilisateurs de consulter et de suivre l'historique de leurs rendez-vous à venir, passés ou annulés. Elle offre une vue d'ensemble des rendez-vous, y compris les détails tels que les dates, les horaires, les prestataires de services, ainsi que le lieu du rendez-vous.

Requis :

1. Filtrage : les utilisateurs devraient pouvoir filtrer leur historique de rendez-vous en fonction du statut du rendez-vous (à venir, passé ou annulé).

2. Consultation des détails des rendez-vous : les utilisateurs devraient pouvoir accéder aux détails complets de chaque rendez-vous dans leur historique. Cela doit inclure le prestataire de service lié au rendez-vous, la date et heure du rendez-vous, la spécialité et lieu du rendez-vous.

3. Base de données : la fonctionnalité devra s'appuyer sur une base de données pour stocker les informations relatives aux rendez-vous des utilisateurs.

Annuler un rendez-vous

La fonctionnalité pour annuler un rendez-vous vise à permettre aux utilisateurs de résilier ou d'annuler un rendez-vous préalablement planifié. Elle offre aux utilisateurs la flexibilité d'annuler des rendez-vous qui ne peuvent pas être honorés.

Requis :

1. Sélection du rendez-vous à annuler : les utilisateurs devraient pouvoir sélectionner un rendez-vous spécifique qu'ils souhaitent annuler. Cela se fera depuis l'historique des rendez-vous, dans l'option de filtrage des rendez-vous « à venir ».

2. Notifications aux parties concernées : la fonctionnalité doit générer des notifications automatiques pour informer l'utilisateur et le prestataire de services concernées, ainsi que l'administrateur de l'application sur l'annulation du rendez-vous.

3. Mise à jour de l'historique : une fois le rendez-vous annulé, la fonctionnalité devrait mettre à jour l'historique des rendez-vous de l'utilisateur pour refléter l'annulation.

4. Base de données : la fonctionnalité doit s'appuyer sur une base de données pour stocker les informations relatives aux rendez-vous, y compris les détails des annulations.

Noter/commenter un service reçu

La fonctionnalité pour noter et commenter un service reçu vise à permettre aux utilisateurs de partager leur expérience et de fournir des commentaires sur les services qu'ils ont reçus. Elle permet aux utilisateurs de donner une évaluation quantitative (notation) et qualitative (commentaire) des prestations afin d'aider les autres utilisateurs à prendre des décisions éclairées lorsqu'ils recherchent des prestataires de services.

Requis :

1. Interface de notation/commentaire : l'interface devrait permettre aux utilisateurs de sélectionner une échelle de notation appropriée (par exemple, une échelle de 1 à 5 étoiles) et de rédiger un commentaire détaillé s'ils le souhaitent.

2. Affichage des évaluations/commentaires : la fonctionnalité devrait fournir un moyen pour les utilisateurs de consulter les évaluations et les commentaires laissés par d'autres utilisateurs. Ces évaluations et commentaires devront être affichés sur la page de profil du prestataire de services.

3. Base de données : la fonctionnalité devra s'appuyer sur une base de données pour stocker les informations relatives aux évaluations, aux commentaires et aux données des utilisateurs.

Modifier ses informations personnelles

La fonctionnalité pour la modification des informations personnelles vise à permettre aux utilisateurs de mettre à jour et de modifier facilement les informations personnelles associées à leur compte utilisateur. Elle offre aux utilisateurs la possibilité de maintenir leurs informations à jour, telles que leur nom, adresse, numéro de téléphone, adresse e-mail, etc.

Requis :

1. Modification des informations de base : les utilisateurs devraient pouvoir modifier des informations de base telles que leur nom, leur adresse, leur numéro de téléphone, leur adresse e-mail, etc. Ils devraient pouvoir mettre à jour ces informations en fonction de leurs besoins actuels.

2. Autorisations de modification : la fonctionnalité devrait mettre en place des mécanismes de sécurité pour s'assurer que seuls les utilisateurs autorisés peuvent modifier leurs informations personnelles. Cela doit inclure l'authentification de l'utilisateur à travers la connexion de celui-ci, et la vérification des droits de modification (un utilisateur ne peut modifier que leurs informations).

3. Base de données : la fonctionnalité devra s'appuyer sur une base de données pour stocker et mettre à jour les informations personnelles des utilisateurs.

Supprimer un compte personnel

La fonctionnalité pour supprimer un compte personnel vise à permettre aux utilisateurs de supprimer définitivement leur compte utilisateur et toutes les données qui y sont associées. Elle offre aux utilisateurs la possibilité de résilier leur adhésion et de supprimer toutes les informations personnelles stockées dans le système.

Requis :

1. Avertissement des conséquences de la suppression : la fonctionnalité doit afficher un avertissement détaillant les conséquences de la suppression. Cela peut inclure des informations sur la perte permanente des données, l'impossibilité de récupérer le compte et les informations associées, et les impacts sur les services ou les fonctionnalités auxquels l'utilisateur avait accès.

2. Suppression de toutes les données personnelles : lorsque la suppression du compte est confirmée, la fonctionnalité doit supprimer toutes les données personnelles de l'utilisateur stockées dans le système. Cela comprend les informations d'identification, les données de profil, les historiques, les préférences et toute autre information spécifique à l'utilisateur.

3. Base de données : la fonctionnalité doit s'appuyer sur une base de données pour stocker les informations utilisateur. Lors de la suppression d'un compte, les données associées à cet utilisateur doivent être effacées de manière sécurisée et irrécupérable.

Devenir prestataire de services

La fonctionnalité pour devenir prestataire de services vise à permettre aux utilisateurs de s'inscrire et de devenir des prestataires de services sur la plateforme. Elle offre aux individus ou aux entreprises la possibilité de proposer leurs services, d'établir leur profil professionnel et d'interagir avec les utilisateurs qui recherchent des prestataires.

Requis :

1. Inscription en tant que prestataire de services : la fonctionnalité doit fournir un processus d'inscription spécifique pour les utilisateurs souhaitant devenir des prestataires de services. Cela doit inclure la collecte d'informations spécifiques pour les prestataires de services telles que les spécialités offertes, l'adresse et la ville professionnelle de l'utilisateur pour valider l'éligibilité en tant que prestataire.

2. Base de données : la fonctionnalité nécessitera une base de données pour stocker les informations des prestataires de services, les services proposés, les réservations, les évaluations, les commentaires et d'autres données pertinentes.

Prestataire de services

Un prestataire de services est en même temps un demandeur de services. Ainsi, il hérite toutes les fonctionnalités liées à un demandeur de services sauf pour la fonctionnalité de suppression de compte personnel. La suppression d'un prestataire doit être réalisée directement par un administrateur.

Accès au planning des jours ouvrables

La fonctionnalité pour l'accès au planning des jours ouvrables vise à permettre aux prestataires de services d'accéder à un planning où ils peuvent visualiser les jours pendant lesquels ils sont disponibles ou indisponibles pour fournir leurs services, ainsi que les nombre de rendez-vous qu'ils ont par jour. Depuis ce planning, les prestataires de services ont accès aux détails de tous les rendez-vous qu'ils ont, filtrés par jour. Cela permet aux prestataires de gérer leur emploi du temps et de communiquer efficacement leurs disponibilités aux utilisateurs.

Requis :

1. Affichage du planning des jours ouvrables : la fonctionnalité doit fournir aux prestataires de services un moyen d'accéder à un planning où ils peuvent voir les jours ouvrables. Ce planning devrait afficher les jours de la semaine, du mois ou d'une période spécifique et indiquer clairement les jours où le prestataire est disponible pour fournir ses services.

2. Accès aux rendez-vous par jour : la fonctionnalité doit fournir aux prestataires de services un moyen d'accéder aux rendez-vous d'un jour déterminé via un lien directement placé sur le planning.

3. Base de données : la fonctionnalité doit utiliser une base de données pour stocker les informations sur la disponibilité des prestataires de services.

Placer des empêchements

La fonctionnalité pour placer des empêchements vise à permettre aux prestataires de services de définir des tranches horaires spécifiques où ils ne seront pas disponibles pour fournir leurs services. Cela permet aux prestataires de gérer leur emploi du temps de manière précise et d'éviter toute confusion ou réservation pendant ces périodes non disponibles.

Requis :

1. Configuration des empêchements : la fonctionnalité doit offrir aux prestataires de services la possibilité de configurer des empêchements, c'est-à-dire des tranches horaires spécifiques où ils ne seront pas disponibles pour fournir leurs services. Les prestataires doivent pouvoir indiquer les jours ouvrables concernés, ainsi que les plages horaires précises qui seront indisponibles.

2. Affichage de la disponibilité aux utilisateurs : il est crucial que les empêchements des prestataires de services soient également visibles pour les utilisateurs lorsqu'ils cherchent à réserver des services. Le système doit afficher clairement les tranches horaires où un prestataire est indisponible, afin d'éviter toute confusion ou tentative de réservation pendant ces périodes.

3. Base de données : La fonctionnalité doit utiliser une base de données pour stocker les informations sur les empêchements des prestataires de services.

Consulter l'historique des rendez-vous en tant que prestataire

La fonctionnalité pour consulter l'historique des rendez-vous en tant que prestataire vise à permettre aux prestataires de services d'accéder à un historique complet de leurs rendez-vous passés. Elle offre aux prestataires une vue d'ensemble de leurs interactions avec les clients, y compris les détails des rendez-vous, les informations des clients et les évaluations reçues.

Requis :

1. La fonctionnalité doit fournir aux prestataires de services un moyen d'accéder à leur historique complet des rendez-vous. Ils devraient pouvoir

consulter les rendez-vous du jour, des prochains jours et les rendez-vous déjà passés sous forme d'une liste, triés par date.

2. Filtrage : pour faciliter la consultation de l'historique des rendez-vous, les prestataires de services devraient pouvoir filtrer les rendez-vous par trois critères : les rendez-vous du jour, les rendez-vous à venir et les rendez-vous passés.

3. Base de données : la fonctionnalité nécessitera une base de données pour stocker les informations des rendez-vous passés, y compris les détails des rendez-vous, les informations des clients, les évaluations et les commentaires.

Signaler qu'un rendez-vous a été effectué

La fonctionnalité pour signaler qu'un rendez-vous a été effectué vise à permettre aux prestataires de services de signaler qu'un rendez-vous a été effectué avec succès. Cela permet de tenir un suivi précis des rendez-vous réalisés, d'actualiser l'historique des rendez-vous et de faciliter la gestion administrative.

Requis :

1. Marquer le rendez-vous comme effectué : la fonctionnalité doit offrir aux prestataires de services la possibilité de marquer un rendez-vous spécifique comme étant effectué. Cela peut être réalisé par le biais d'un

bouton ou d'une option dédiée dans l'interface utilisateur associée à chaque rendez-vous.

2. Mise à jour de l'historique des rendez-vous : une fois qu'un rendez-vous est signalé comme effectué, la fonctionnalité doit mettre à jour l'historique des rendez-vous du prestataire de services. Cela permettra de refléter précisément le statut du rendez-vous, de faciliter la consultation ultérieure et d'obtenir des données statistiques sur les rendez-vous réalisés.

3. Base de données : la fonctionnalité nécessitera une base de données pour stocker les informations sur les rendez-vous effectués, y compris son statut, renseigné par le prestataire de services.

Annuler rendez-vous à venir

La fonctionnalité pour annuler un rendez-vous à venir vise à permettre aux prestataires de services d'annuler efficacement les rendez-vous programmés qui ne peuvent pas être honorés. Cela permet de gérer les changements d'emploi du temps, d'informer les clients concernés et de maintenir une communication transparente pour éviter les désagréments et les confusions.

Requis :

1. Accès aux rendez-vous à venir : la fonctionnalité doit fournir aux prestataires de services un accès facile aux rendez-vous à venir programmés avec leurs clients. Ils devraient pouvoir consulter une liste

contenant ses rendez-vous à venir, triés par date et heure, pour identifier ceux qui nécessitent une annulation.

2. Annulation des rendez-vous : une fois qu'un rendez-vous à venir est identifié pour annulation, la fonctionnalité doit permettre aux prestataires de services d'initier le processus d'annulation. Cela sera réalisé par le biais d'un bouton associé à chaque rendez-vous.

3. Mise à jour de l'historique des rendez-vous : une fois qu'un rendez-vous à venir est annulé, la fonctionnalité doit mettre à jour l'historique des rendez-vous du prestataire de services pour refléter le statut d'annulation. Cela permettra de conserver un suivi précis des rendez-vous passés et annulés.

4. Base de données : la fonctionnalité nécessitera une base de données pour stocker les informations sur les rendez-vous annulés.

Emprunter un véhicule

La fonctionnalité "Emprunter un véhicule" vise à permettre aux prestataires de services de réserver et d'emprunter un véhicule dans le cadre de leurs activités professionnelles. Cela facilite la mobilité des prestataires de services, leur permettant d'effectuer leurs missions et de fournir des services à leurs clients de manière efficace et flexible.

Requis :

1. Recherche et disponibilité des véhicules : la fonctionnalité doit permettre aux prestataires de services de rechercher et de vérifier la disponibilité des véhicules mis à disposition. Les informations sur les types de véhicules, les caractéristiques, les emplacements de prise en charge et de dépôt doivent être clairement affichées.

2. Réservation de véhicules : une fois qu'un prestataire de services a sélectionné un véhicule disponible, la fonctionnalité doit lui permettre de réserver le véhicule pour une période spécifique. Cela peut être réalisé en saisissant la date et l'heure de prise en charge et de dépôt du véhicule.

3. Validation et confirmation de la réservation : la fonctionnalité doit valider la disponibilité du véhicule et confirmer la réservation au prestataire de services. Une fois la réservation confirmée, le prestataire de services doit recevoir une notification de confirmation avec les détails de la réservation.

4. Suivi et historique des emprunts : la fonctionnalité doit permettre aux prestataires de services de consulter un historique des véhicules empruntés précédemment, y compris les dates, les durées d'emprunt et les informations sur les véhicules. Cela facilite le suivi des utilisations passées et permet de planifier efficacement les futurs emprunts.

5. Intégration de système de réservation : la fonctionnalité nécessitera l'intégration d'un système de réservation pour gérer les disponibilités des véhicules et faciliter les réservations en temps réel.

Consulter l'historique des emprunts de véhicules

La fonctionnalité pour consulter l'historique des emprunts de véhicules vise à permettre aux prestataires de services d'accéder à un historique détaillé de tous les emprunts de véhicules qu'ils ont effectués dans le cadre de leurs activités professionnelles. Cela facilite la gestion des utilisations passées, permet de vérifier les informations relatives aux emprunts et facilite la planification future.

Requis :

1. Affichage de l'historique des emprunts : la fonctionnalité doit permettre aux prestataires de services de consulter un historique complet de tous les véhicules qu'ils ont empruntés. L'historique devrait inclure les détails tels que les dates d'emprunt, le modèle du véhicule et son numéro d'immatriculation.
2. Tri et filtrage des emprunts : pour faciliter la recherche d'informations spécifiques, la fonctionnalité doit permettre de filtrer les emprunts par les catégories : actuels, prochains, en attente de confirmation et passés. La liste résultant du filtre sera affichée par ordre de date croissante.
3. Affichage des détails des emprunts : la liste affichée dans l'historique devra montrer la date d'emprunt, le modèle du véhicule et le numéro d'immatriculation pour chacun des emprunts.

4. Base de données : la fonctionnalité nécessitera une base de données pour stocker et gérer les informations relatives à l'historique des emprunts de véhicules, y compris les détails des emprunts et les informations associées.

Enregistrer des dépenses et notes de frais

La fonctionnalité pour l'enregistrement des dépenses et notes de frais vise à permettre aux prestataires de services de saisir, suivre et gérer leurs dépenses professionnelles et leurs notes de frais de manière pratique et organisée. Cela facilite la gestion financière et la comptabilité des activités professionnelles.

Requis :

1. Saisie des dépenses et notes de frais : la fonctionnalité doit permettre aux prestataires de services d'enregistrer les dépenses liées à leur activité professionnelle, telles que les frais de déplacement, les repas, hébergement, etc. Ils doivent pouvoir saisir les détails de la dépense, y compris la date, la catégorie, le montant, le fournisseur ainsi que des commentaires.

2. Gestion des catégories de dépenses : la fonctionnalité doit permettre aux prestataires de services de définir et de gérer des catégories de dépenses pour organiser leurs enregistrements. Ils doivent pouvoir créer des catégories personnalisées, telles que "transport", "hébergement", "repas", etc., afin de classer leurs dépenses de manière appropriée.

3. Capture de reçus et pièces justificatives : pour faciliter la gestion des dépenses, la fonctionnalité peut permettre aux prestataires de services de capturer et de joindre des reçus et pièces justificatives à leurs enregistrements.

4. Calcul automatique des totaux et des taxes : la fonctionnalité peut inclure un calculateur automatique pour aider les prestataires de services à déterminer les totaux des dépenses. Cela facilite la gestion des montants dépensés et la génération de rapports financiers précis.

5. Base de données : la fonctionnalité nécessitera une base de données pour stocker et gérer les informations relatives aux dépenses et notes de frais enregistrées par les prestataires de services.

Consulter la situation des dépenses et notes de frais

La fonctionnalité pour la consultation de la situation des dépenses et notes de frais vise à permettre aux prestataires de services de visualiser et de suivre le statut de leurs demandes de remboursement de dépenses et de notes de frais. Cela leur permet de connaître l'état de chaque demande, de la création jusqu'au paiement, en passant par les différentes étapes d'approbation et de validation.

Requis :

1. Vue d'ensemble des demandes de remboursement : la fonctionnalité doit fournir aux prestataires de services une vue d'ensemble de leurs demandes de remboursement de dépenses et de notes de frais. Ils doivent pouvoir consulter une liste des demandes soumises, affichant des informations telles que la date de soumission, le montant demandé et le statut actuel de chaque demande.

2. Catégorisation des demandes par statut : les demandes de remboursement doivent être catégorisées en fonction de leur statut pour une visualisation plus claire. Les catégories doivent inclure "en attente", "validée", "refusée" et "payée". Les prestataires de services doivent pouvoir filtrer les demandes en fonction de ces catégories pour une consultation plus ciblée.

3. Détails des demandes : en sélectionnant une demande spécifique, les prestataires de services doivent pouvoir accéder aux détails complets de cette demande. Cela peut inclure des informations telles que les pièces jointes, la description des dépenses, les dates, les catégories et les commentaires associés à la demande.

4. Base de données : la fonctionnalité nécessitera une base de données pour stocker et gérer les informations relatives aux demandes de remboursement, y compris les détails des demandes et les statuts.

Administrateur

Ajout d'un membre du personnel

La fonctionnalité pour l'ajout d'un membre du personnel vise à permettre aux administrateurs d'ajouter de nouveaux membres du personnel au système. Cela facilite la gestion des utilisateurs et des rôles au sein de l'application, en permettant aux administrateurs d'attribuer des autorisations spécifiques à chaque membre du personnel.

Requis :

1. Création d'un compte utilisateur : la fonctionnalité doit permettre aux administrateurs de créer un compte utilisateur pour chaque nouveau membre du personnel. Ils doivent pouvoir saisir les informations nécessaires telles que le nom complet et leur adresse e-mail pour l'identification du membre du personnel.
2. Base de données : la fonctionnalité nécessitera une base de données pour stocker et gérer les informations relatives aux membres du personnel, y compris les détails des comptes utilisateurs.

Recherche des utilisateurs

La fonctionnalité pour la recherche des utilisateurs vise à permettre aux administrateurs de rechercher et de retrouver facilement les utilisateurs du

système en fonction de leur nom ou prénom. Cela permet aux administrateurs d'accéder rapidement aux informations pertinentes et de gérer efficacement les utilisateurs.

Requis :

1. Recherche par nom ou prénom : La fonctionnalité doit permettre aux administrateurs d'effectuer une recherche en saisissant un nom ou un prénom dans le champ de recherche. Le système effectue ensuite une recherche parmi les utilisateurs en vérifiant si l'expression saisie correspond à un nom ou à un prénom dans la base de données.

2. Affichage des résultats de recherche : Une fois la recherche effectuée, la fonctionnalité doit afficher les résultats correspondants aux critères de recherche spécifiés. Les résultats doivent inclure les informations pertinentes pour chaque utilisateur, telles que la photo de profil, le nom complet, email, leurs spécialités, leurs adresses, etc. Les résultats seront présentés sous forme de liste pour une consultation facile.

3. Consultation des profils utilisateur : en sélectionnant un utilisateur dans les résultats de recherche, les administrateurs doivent pouvoir accéder au profil complet de cet utilisateur. Le profil peut contenir des informations détaillées sur l'utilisateur.

4. Actions sur les profils utilisateur : la fonctionnalité peut également permettre aux administrateurs d'effectuer certaines actions sur les profils des utilisateurs. Cela doit inclure l'affichage des détails, la modification des

informations personnelles et la suppression du compte de l'utilisateur. Les administrateurs doivent disposer des autorisations appropriées pour effectuer ces actions.

5. Base de données : la fonctionnalité nécessitera une base de données pour stocker et gérer les informations relatives aux utilisateurs.

Modification et suppression des utilisateurs

La fonctionnalité "Modification et suppression des utilisateurs" vise à permettre aux administrateurs d'effectuer des modifications et des suppressions sur les profils des utilisateurs du système. Cela permet aux administrateurs de gérer efficacement les informations utilisateur et d'assurer l'intégrité des données.

Requis :

1. Modification des profils utilisateur : la fonctionnalité doit permettre aux administrateurs de modifier les informations des utilisateurs, telles que le nom, le prénom, l'adresse e-mail, le numéro de téléphone, l'adresse, etc.
2. Suppression des profils utilisateur : la fonctionnalité doit également permettre aux administrateurs de supprimer les profils des utilisateurs du système. Cela peut être nécessaire dans des situations telles que des utilisateurs inactifs, des comptes en double ou des utilisateurs qui ont demandé la suppression de leur compte. Lorsqu'un profil utilisateur est

supprimé, toutes les données associées à ce profil doivent également être supprimées ou anonymisées.

3. Base de données : la fonctionnalité nécessitera une base de données pour stocker et gérer les informations des utilisateurs.

Consultation de statistiques

La fonctionnalité "Consultation de statistiques" vise à fournir aux administrateurs un aperçu global des données et des statistiques liées au fonctionnement du site. Elle permet aux administrateurs d'analyser les tendances, de prendre des décisions éclairées et de surveiller la performance globale du service.

Requis :

1. Statistiques par spécialité : la fonctionnalité doit afficher le nombre de rendez-vous effectués par spécialité. Cela permet aux administrateurs de visualiser les spécialités les plus demandées et d'ajuster les ressources en conséquence. Les statistiques par spécialité seront présentées sous forme graphique, concrètement des camemberts, pour une compréhension claire et rapide.

2. Statistiques par jour de la semaine : la fonctionnalité doit fournir le nombre de rendez-vous enregistrés pour chaque jour de la semaine. Cela permet aux administrateurs d'identifier les jours de la semaine les plus fréquentés et d'optimiser la planification des ressources en conséquence.

Les statistiques par jour de la semaine seront présentées graphiquement sous forme de diagramme de barres, pour une analyse visuelle facile.

3. Statistiques par tranche horaire : la fonctionnalité doit afficher le nombre de rendez-vous par tranche horaire. Cela permet aux administrateurs de connaître les heures de la journée les plus demandées et de mieux répartir les rendez-vous. Les statistiques par tranche horaire seront représentées graphiquement sous forme de diagramme de barres pour une visualisation claire des tendances.

4. Taux de rendez-vous par statut : la fonctionnalité doit indiquer le taux de rendez-vous par statut, c'est-à-dire le pourcentage de rendez-vous annulés, en attente et confirmés. Cela permet aux administrateurs d'évaluer l'efficacité de la gestion des rendez-vous et de prendre des mesures pour améliorer les taux de confirmation et de réduction des annulations. Les taux de rendez-vous par statut seront représentés graphiquement sous forme de camembert pour une compréhension claire et rapide

5. Collecte de données : la fonctionnalité doit disposer d'un mécanisme pour collecter et consolider les données nécessaires à la génération des statistiques. Ces données proviendront de la base de données des rendez-vous et doivent être mises à jour en temps réel ou à intervalles réguliers pour refléter les dernières informations.

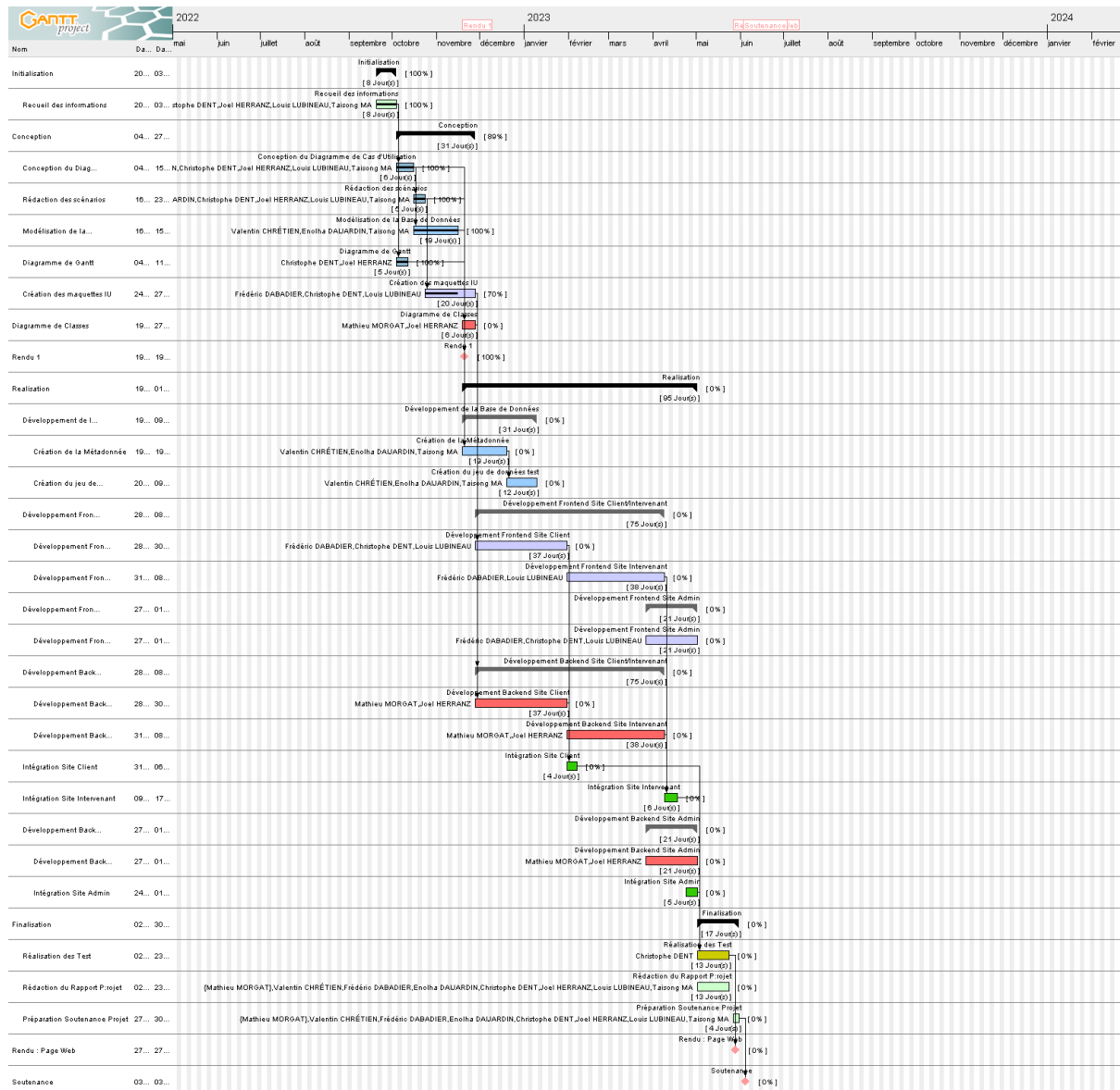
Répartition des rôles

La répartition des rôles pour la réalisation du projet.

Valentin CHRÉTIEN	- Architecte de base de données / Développeur
Frédéric DABADIE	- Développeur principal
Enolha DAIJARDIN	- Développeuse
Christophe DENT	- Testeur / Développeur
Joel HERRANZ	- Conception / Développeur / Rédaction
Louis LUBINEAU	- Chef de projet / Graphiste / Développeur principal
Taisong MA	- Conception / Développeur
Mathieu MORGAT	- Développeur / Testeur qualité

Gantt

Diagramme de Gantt pour la planification du projet des différentes réalisations.



Lien du diagramme :

GANTT_PRISM_PROJETWEB.pdf

Prototype – Maquette (IHM)

La maquette du projet, disponible ci-dessous. Elle regroupe les différentes sections du site. Nous nous sommes basés sur celle-ci pour réaliser l'intégration du site.

Elle est divisée en 5 sections :

- Mode Visiteur
- Mode Demandeur
- Pages Intervenant
- Personnel administratif
- Administrateur

[» Lien maquette Figma](#)

Outils technologiques utilisés

Voici les différents outils technologiques que nous avons utilisés pour la conception, la gestion et le développement du projet.

Gestion de projet / Versionning :

- GitHub
- Slack / Discord / Trello

Serveur web

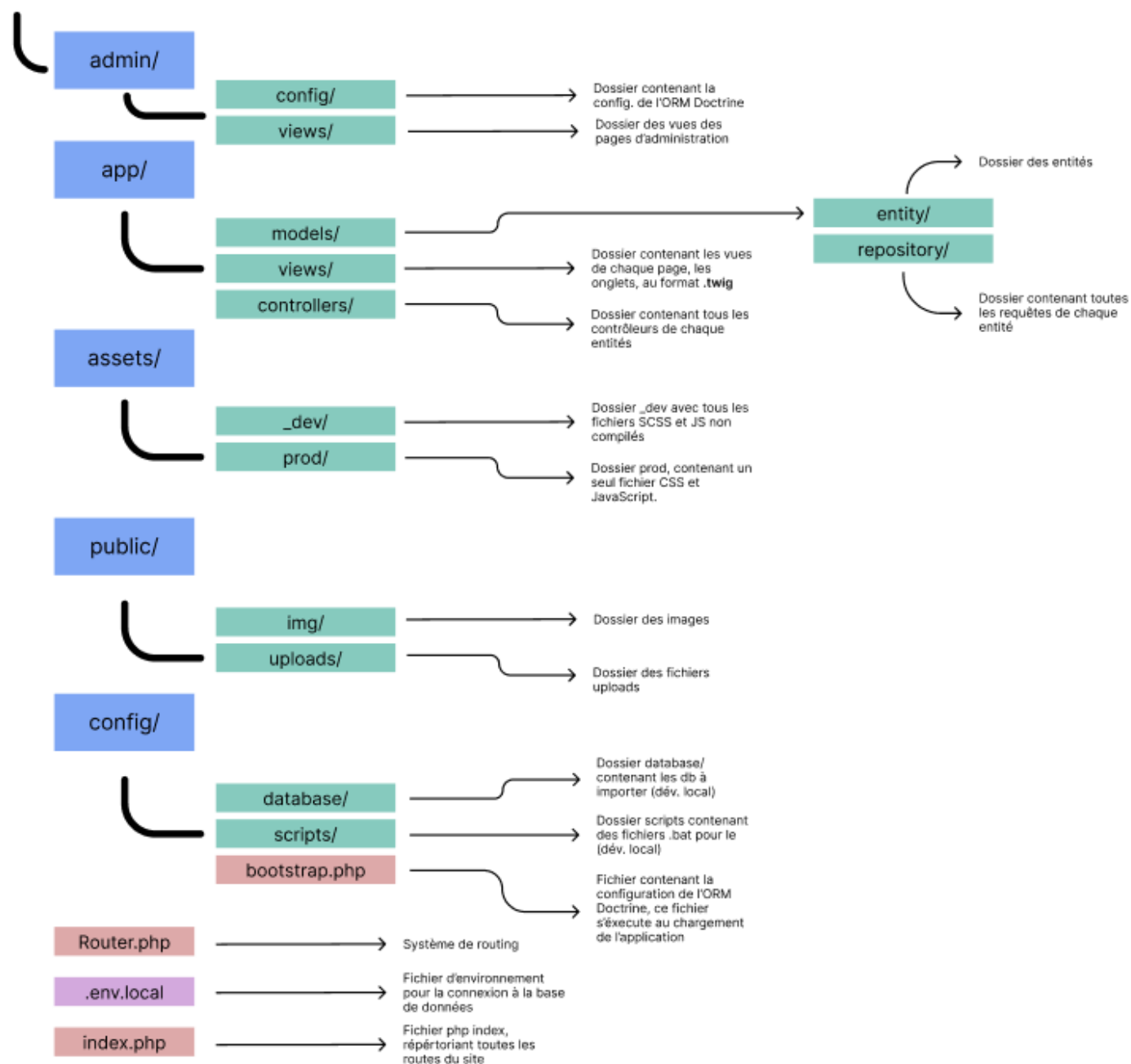
- Développement local :
 - Wampserver / Xampp
 - Docker
- Environnement de production :
 - Ionos Web Hosting

Environnement technique

- Gestion des dépendances PHP avec **Composer** (*Back-End*)
 - Moteur de Template : **Twig**
 - ORM (Object Relational Mapping) : **Doctrine/ORM**
 - Envoie de mails : **PHPMailer**
 - Variables global : **Dotenv**
 - Gestion d'images : **Intervention/Image**
- Gestion des dépendance JavaScript avec **NodeJS** / NPM (*Front-End*)
 - Pré-processeur CSS : **SCSS**
 - Compilation des fichiers : **Webpack**
 - Slider des avis : **Swiper JS**
- Utilisation d'AJAX natif pour des appels dynamiques à la BDD.

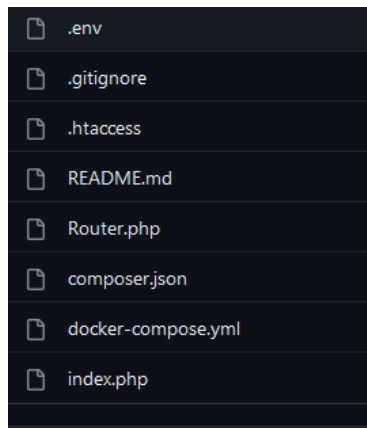
Arborescence

Voici l'arborescence utilisée pour le projet G.A.P.M. Elle est non exhaustive, et seulement à but indicatif avec les fichiers et dossiers les plus importants. L'explication de l'architecture est dans la section suivante.



Architecture utilisée

L'architecture du projet peut être décrite en détail afin de faciliter la compréhension de sa structure. Tout d'abord, examinons les fichiers présents à la racine du projet. Les voici :

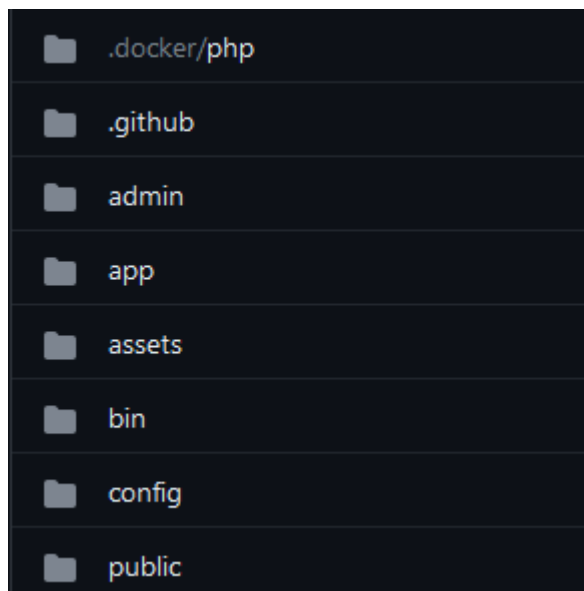


- **.env** : Ce fichier est un exemple contenant des champs à remplir, principalement les informations d'accès à la base de données du projet. L'utilisateur doit simplement copier ce fichier et le renommer en ".env.local" pour se connecter à la base de données.
- **.gitignore** : Ce fichier répertorie tous les fichiers et dossiers qui ne doivent pas être envoyés vers Git. Il inclut notamment le fichier ".env.local" afin de ne pas divulguer les informations de connexion. Il contient également les dépendances de Composer (le dossier "vendor/") ainsi que les dépendances de Node.js (le dossier "node_modules/").
- **.htaccess** : Ce fichier gère les différentes redirections du projet.
- **README.md** : Ce fichier fournit des instructions détaillées sur l'installation du projet.

- **Router.php** : Ce fichier est responsable de la gestion du système de routage.
- **composer.json** : Dans ce fichier, toutes les dépendances du projet sont répertoriées.
- **docker-compose.yml** : Ce fichier permet d'exécuter un serveur web en utilisant Docker avec Apache et phpMyAdmin.
- **index.php** : Ce fichier répertorie toutes les routes du site web et permet également de charger l'ORM Doctrine grâce au fichier "Bootstrap.php".

Après avoir passé en revue les différents fichiers à la racine, il convient de souligner leur importance pour le bon fonctionnement, l'installation et le développement de l'application.

Passons maintenant aux dossiers de l'application. Voici la liste :



Pour une visualisation plus détaillée du contenu de chaque dossier, vous pouvez consulter la section [Arborescence](#) ou le dépôt GitHub (disponible [ICI](#)).

Les deux premiers dossiers, ".docker" et ".github", renferment les fichiers de configuration spécifiques à Docker et GitHub respectivement.

Par la suite, nous avons le dossier "**admin/**", qui inclut un fichier "index.php" utilisé comme routeur, ainsi que les vues (templates) associées à chaque section de l'administration.

Le dossier "**app/**" représente le cœur du projet et suit l'architecture MVC (Modèle-Vue-Contrôleur). Il est composé des sous-dossiers suivants : **models/**, **views/** et **controllers/**.

Dans le dossier "**models/**", nous retrouvons deux sous-dossiers importants : "**entity/**" qui regroupe toutes les entités du projet, et "**repository/**" qui contient les requêtes personnalisées pour chaque entité.

En utilisant l'ORM Doctrine, chaque entité peut être associée à un "Repository" incluant des méthodes personnalisées, en plus des méthodes de base qui ne sont pas explicitement énumérées.

Le dossier "**assets**/" regroupe l'ensemble des fichiers essentiels au design du site, tels que le CSS, le JavaScript et les polices de caractères (fonts).

À l'intérieur de ce dossier, on distingue deux sous-dossiers :

- "**_dev**/" : Ce sous-dossier contient les fichiers JavaScript et SCSS (un pré-processeur de CSS) qui sont séparés en plusieurs fichiers. Ces fichiers sont maintenus de manière modulaire pour faciliter le développement. Le fichier de configuration `webpack.config.js` est responsable de les compiler en un seul fichier exploitable. Une fois compilés, ces fichiers sont déplacés vers le dossier "prod/".
- "**prod**/" : Les fichiers compilés sont stockés dans ce dossier, également connu sous le nom de "dist/" (pour distribution). Lors du déploiement en environnement de production, seuls les fichiers JavaScript et CSS compressés, les polices de caractères et certaines images sont chargés. Ainsi, seules les ressources essentielles et optimisées sont utilisées dans le cadre de la production du site.

Le dossier "**bin**/" contient le fichier "doctrine", responsable de l'intégration de l'ORM au cœur du projet. Par ailleurs, le dossier "config" comprend deux sous-dossiers, à savoir "scripts" et "databases". Ces sous-dossiers jouent un rôle essentiel dans l'amélioration du développement local. Plus précisément, le dossier "scripts" contient des scripts liés à la base de données, y compris les scripts de mise à jour et l'installation de dépendances. En parallèle, le dossier "databases" est utilisé pour gérer les aspects liés aux bases de données.

Enfin, le dossier "**public/**" est dédié au stockage des informations. Cela inclut, par exemple, les téléchargements de photos de profil ainsi que le chargement des notes de frais. Ce dossier joue un rôle crucial dans la gestion des ressources accessibles aux utilisateurs externes de l'application.

Explication brève du fonctionnement de l'application

Après avoir passé en revue les différents fichiers et dossiers de l'application, voici une brève explication de son fonctionnement et de la raison pour laquelle tous ces fichiers sont nécessaires.

Le fichier `index.php` est le fichier par défaut lorsque l'utilisateur accède au site web. Comme mentionné précédemment, ce fichier contient toutes les actions possibles sur le site. Lorsqu'un utilisateur visite une URL spécifique, le système de routage (`Router.php`) va vérifier si cette URL correspond à l'une des routes définies dans le fichier `index.php`.

Une route est définie selon le format suivant : URL, Contrôleur, Action. Par exemple, si l'utilisateur souhaite afficher tous les intervenants, il se rendra sur l'URL "intervenants". Dans le fichier `index.php`, nous avons associé la route "`intervenants`" au contrôleur "IntervenantController" et à l'action "afficherIntervenantsVue".

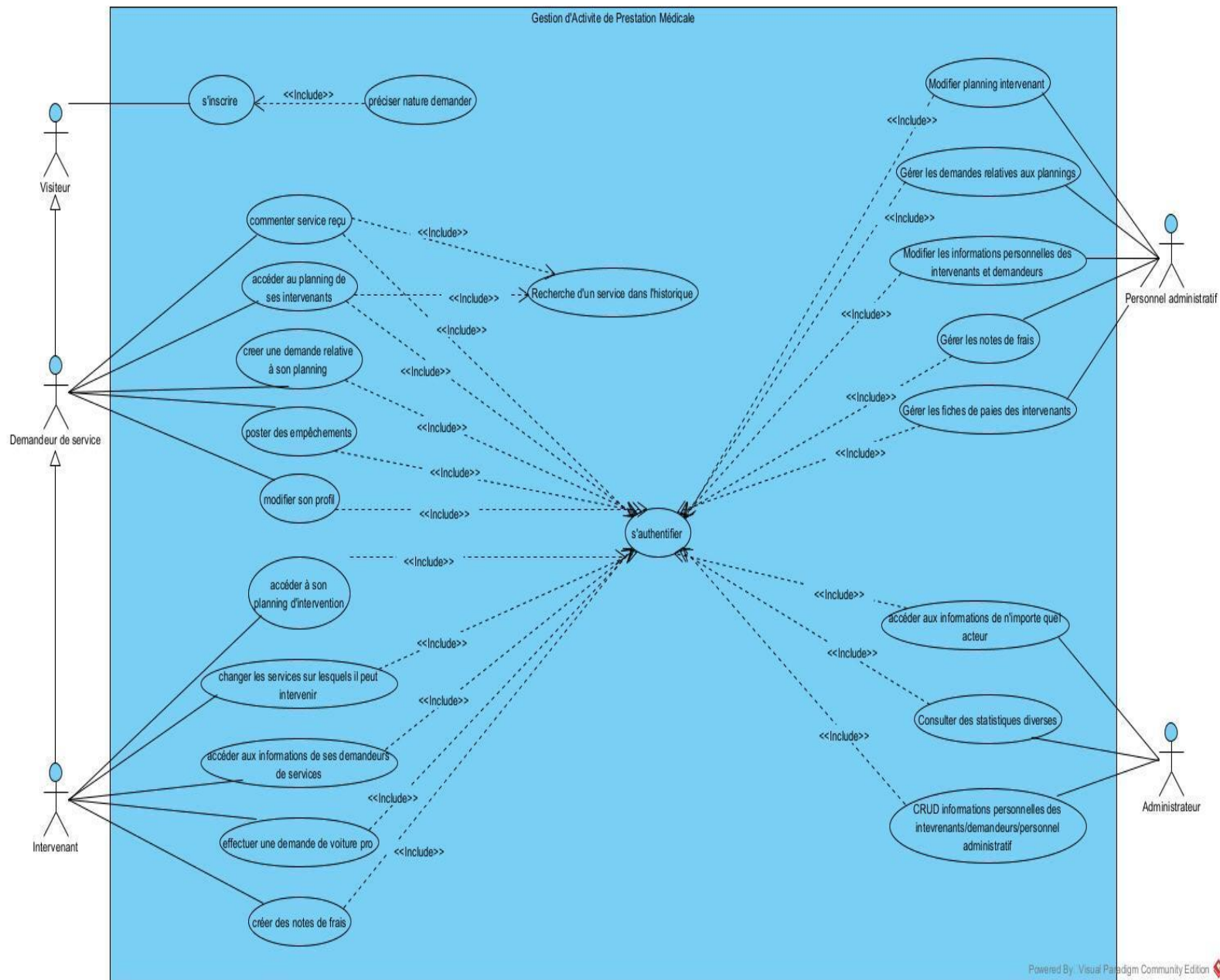
Le routeur fera appel au contrôleur des intervenants et à la méthode "afficherIntervenantsVue". Cette méthode contiendra une fonction Doctrine permettant de récupérer tous les intervenants à partir de l'entité "**Intervenant**" en utilisant la méthode "**findAll()**", en utilisant également le principe de Repository.

Nous récupérons ainsi toutes les données du modèle sous forme d'une collection d'objets (étant donné qu'il peut y avoir plusieurs intervenants). Il ne nous reste plus qu'à retourner une vue en passant les variables nécessaires, afin de les utiliser dans le template.

Le template hérite d'un "layout", un fichier de base qui contient des informations communes à toutes les pages HTML, telles que l'en-tête, le pied de page et divers liens. Ce layout inclut les fichiers du dossier "prod/", tels que le CSS et le JavaScript. Lorsque nous renvoyons notre vue "IntervenantVue.twig", celle-ci hérite du layout, ce qui signifie que le style du site est déjà présent. Ce style est généré à partir des fichiers "_dev/scss", qui, à chaque sauvegarde, sont compilés vers les fichiers du dossier "prod/".

Modélisation – Diagrammes

Cas d'utilisation (Use case)



Modèle Conceptuel de Données (MCD)

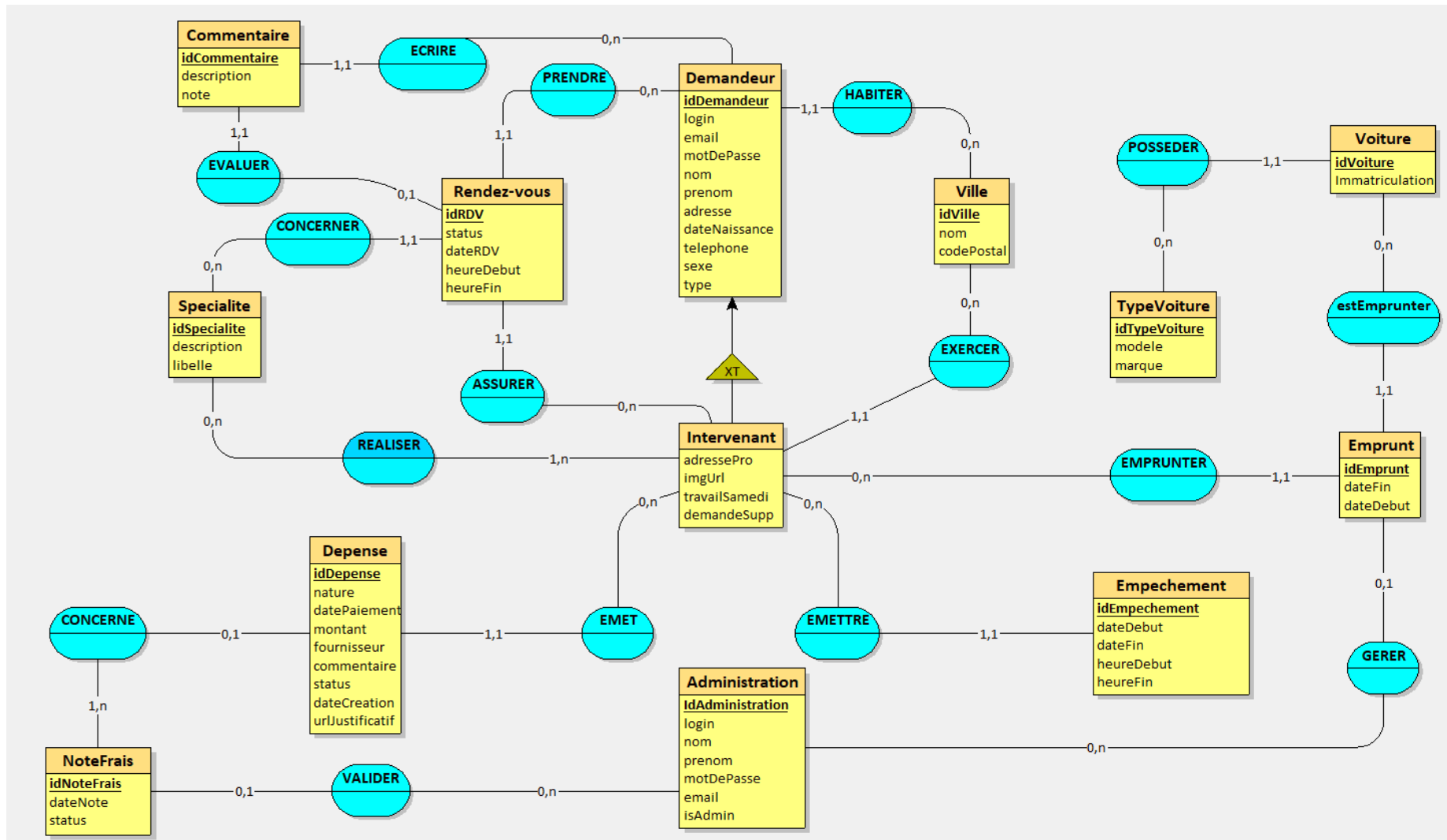
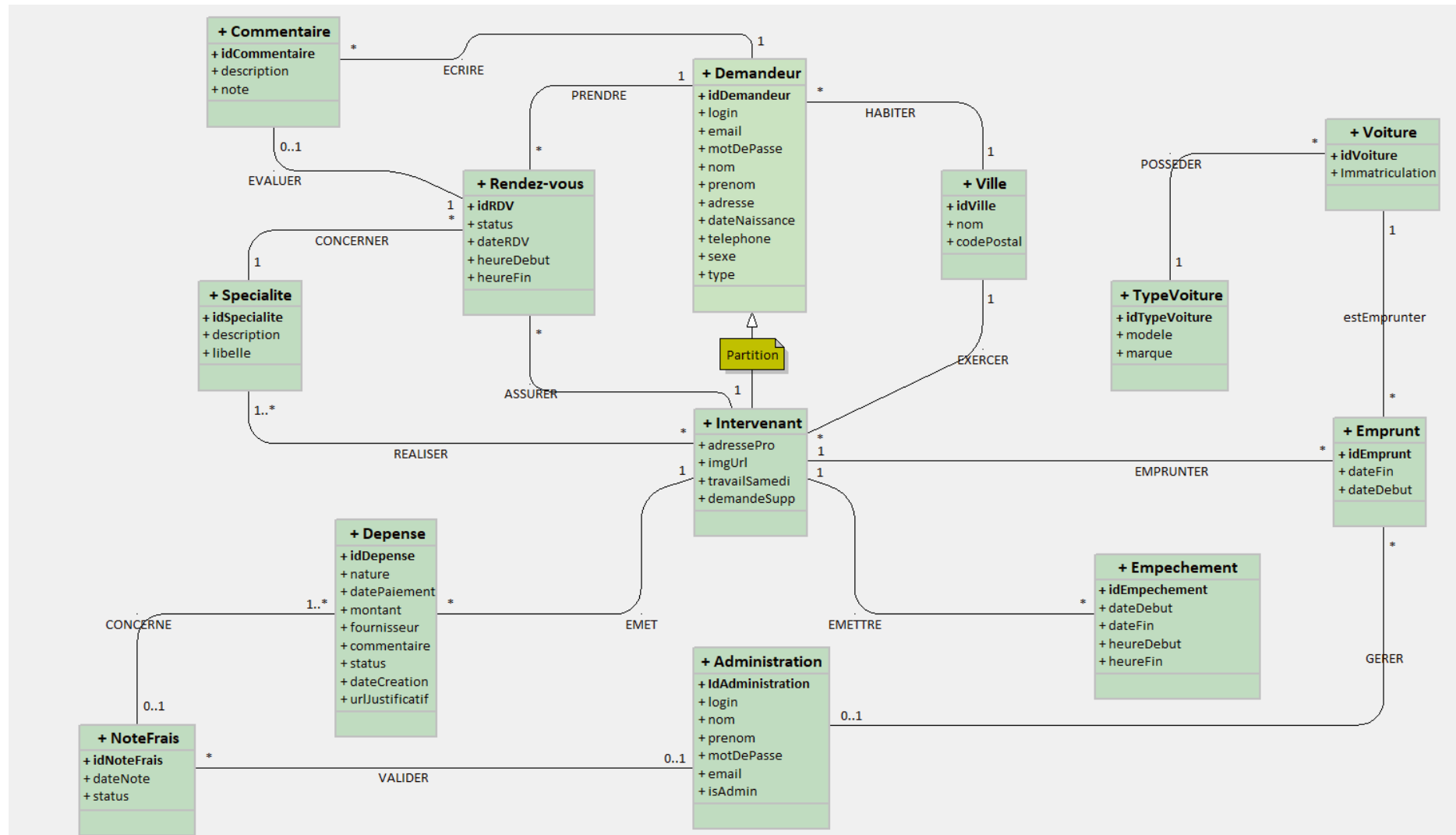
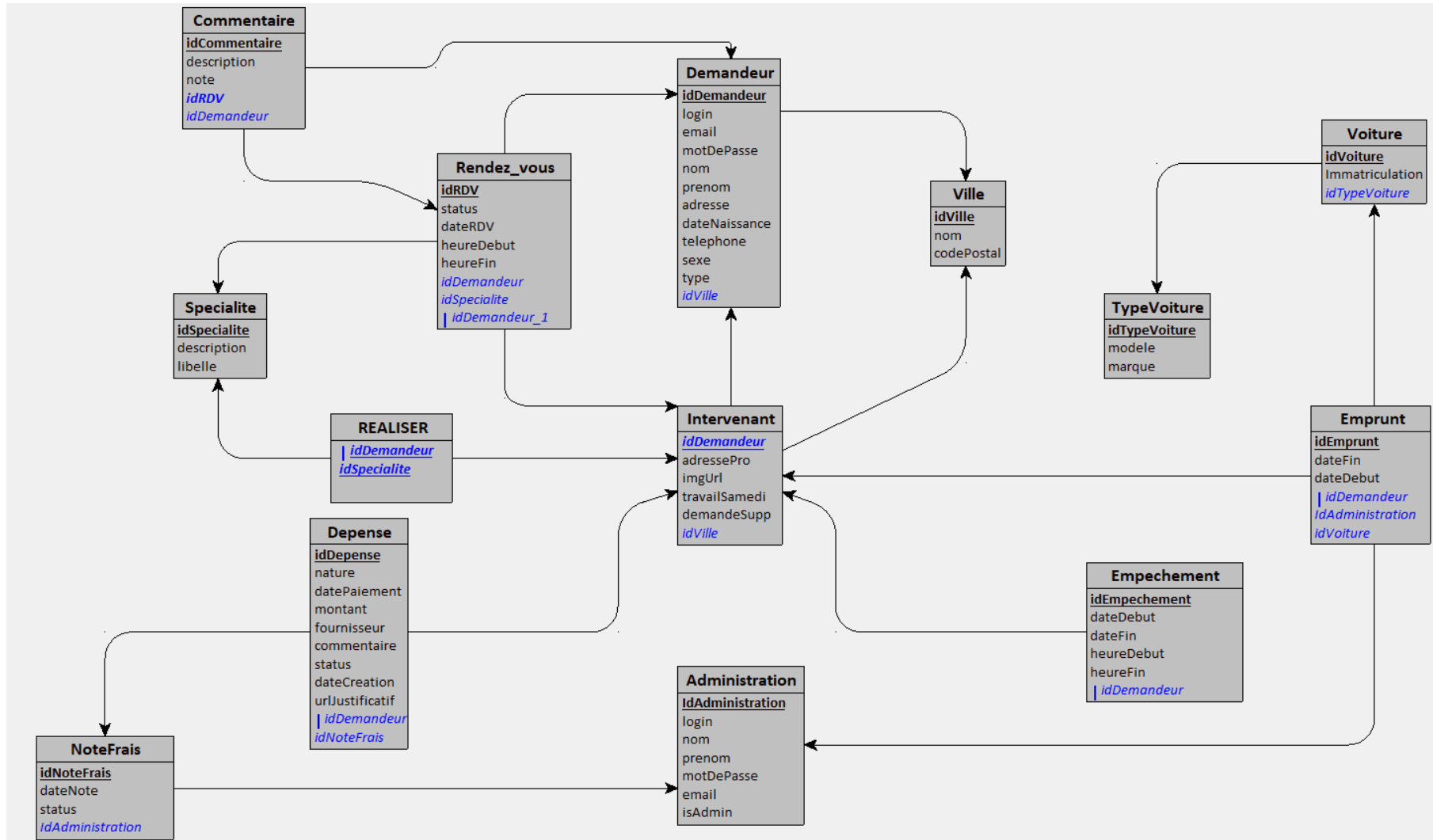


Diagramme de classe (UML)



Modèle Logique de Données (MLD)



Explications de codes

Code du routeur **index.php**

```
1  <?php
2
3  use App\controllers\Router;
4  use App\models\entity\Session;
5
6
7  require_once "Router.php";
8  require_once 'vendor/autoload.php';
9  require_once 'config/bootstrap.php';
10
11 session_name('user');
12 Session::start();
13
14 global $entityManagerFactory;
15 $entityManager = $entityManagerFactory();
16 $router = new Router();
17
18 /*
19  * Routes GET
20  */
21 $router->addRoute('demandeur', 'DemandeurController', 'index');
22 $router->addRoute('search', 'SearchController', 'index');
23 $router->addRoute('logout', 'DemandeurController', 'logout');
24 $router->addRoute('my-account', 'DemandeurController', 'displayMyAccount');
25 $router->addRoute('inscription-intervenant', 'IntervenantController', 'index');
26 $router->addRoute('notes-de-frais', 'NoteFraisController', 'displayNoteFrais');
27 $router->addRoute('mes-rendez-vous', 'RendezVousController', 'displayMyRdv');
28 $router->addRoute('liste-rdv', 'RendezVousController', 'displayMyRdvIntervenant');
29 $router->addRoute('planning', 'IntervenantController', 'displayPlanning');
30
31 /*
32  * Routes POST
33  */
34 $router->addRoute('login', 'DemandeurController', 'login');
35 $router->addRoute('register', 'DemandeurController', 'register');
36 $router->addRoute('my-account-edit', 'DemandeurController', 'update');
37 $router->addRoute('my-account-delete', 'DemandeurController', 'delete');
38 $router->addRoute('confirm-rdv', 'RendezVousController', 'createRDV');
39 $router->addRoute('ajout-avis', 'RendezVousController', 'createNoticeOnRdv');
40 $router->addRoute('upgrade-to-intervenant', 'IntervenantController', 'devenirIntervenant');
41 $router->addRoute('update-intervenant', 'IntervenantController', 'update');
42
43 $router->addRoute('intervenant-unsubscribe-request', 'IntervenantController', 'unsubscribeRequest');
44 $router->addRoute('intervenant-cancel-unsubscribe-request', 'IntervenantController', 'cancelUnsubscribe');
45
46 $router->addRoute('picture-edit', 'IntervenantController', 'updatePicture');
47 $router->addRoute('faq', 'HomeController', 'displayFAQ');
48 $router->addRoute('oubli', 'HomeController', 'displayForgotten');
49 $router->addRoute('forgotten', 'DemandeurController', 'forgottenMail');
50 $router->addRoute('toggle-mode-intervenant', 'IntervenantController', 'toggleModeIntervenant');
51 $router->addRoute('emprunter-vehicule', 'IntervenantController', 'emprunterVehicule');
52
53 /*
54  * Routes avec paramètres
55  */
56 $router->addRoute('my-account&nav=<nav>', 'DemandeurController', 'displayMyAccount');
57 $router->addRoute('notes-de-frais&nav=<nav>', 'NoteFraisController', 'displayNoteFrais');
58
59 $router->addRoute('profile&id=<id>', 'IntervenantController', 'profile');
60
```

Il permet de définir au routeur les différentes routes de l'application.

Exemple de l'Entité **Demandeur**, on peut y voir l'utilisation de L'ORM **Doctrine**.

```
1  <?php
2  namespace App\models\entity;
3
4  use App\models\repository\DemandeurRepository;
5  use Doctrine\Common\Collections\Collection;
6  use Doctrine\DBAL\Types\Types;
7  use Doctrine\ORM\Mapping as ORM;
8
9  #[ORM\Entity(repositoryClass: DemandeurRepository::class)]
10 #[ORM\InheritanceType('JOINED')]
11 #[ORM\DiscriminatorColumn(name: 'type', type: 'string')]
12 #[ORM\DiscriminatorMap(['demandeur' => Demandeur::class, 'intervenant' => Intervenant::class])]
13 #[ORM\Table(name: 'Demandeur')]
14 class Demandeur
15 {
16     #[ORM\Id]
17     #[ORM\Column(type: Types::INTEGER)]
18     #[ORM\GeneratedValue]
19     protected int $idDemandeur;
20     #[ORM\Column(unique: true)]
21     private string $login;
22     #[ORM\Column(unique: true)]
23     private string $email;
24     #[ORM\Column]
25     private string $motDePasse;
26     #[ORM\Column]
27     private string $nom;
28     #[ORM\Column]
29     private string $prenom;
30     #[ORM\Column]
31     private string $dateNaissance;
32     #[ORM\Column]
33     private string $adresse;
34     #[ORM\Column]
35     private string $telephone;
36     #[ORM\Column]
37     private string $sexe;
38     #[ORM\ManyToOne(targetEntity: Ville::class, cascade: ['persist'], fetch: 'EAGER')]
39     #[ORM\JoinColumn(name: 'idVille', referencedColumnName: 'idVille', nullable: false)]
40     private Ville $ville;
41     #[ORM\OneToMany(mappedBy: 'demandeur', targetEntity: RendezVous::class, fetch: 'LAZY')]
42     private $rendezVous;
43
44     #[ORM\OneToMany(mappedBy: 'demandeur', targetEntity: Commentaire::class, fetch: 'LAZY')]
45     private $commentaires;
46
47     private $type;
48
49
50     // Getteur & Setteur
51
52 }
```

Sujet 2 – M. Gagné.

Le code de base “layout” du projet, le fichier **base.twig**.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{{ app.request.getSchemeAndHttpHost() }}/assets/prod/css/main.min.css">
    <link rel="icon" href="{{ app.request.getSchemeAndHttpHost() }}/assets/img/logo.svg"/>
    <title>{% block title %} • GAPM</title>
  </head>
  <body class="{% if modeIntervenant or isIntervenant %}intervenant{% else %}demandeur{% endif %} {% block type %}{% endblock %}"
    data-log="{{ userLogged }}">
    <div id="loader"></div>
    {% if not no_header %}
      {% include "includes/header.twig" %}
    {% endif %}
    <div id="content">
      {% block content %}

      {% endblock %}
    </div>

    {% include "includes/footer.twig" %}
    {% include 'includes/notification.twig' %}

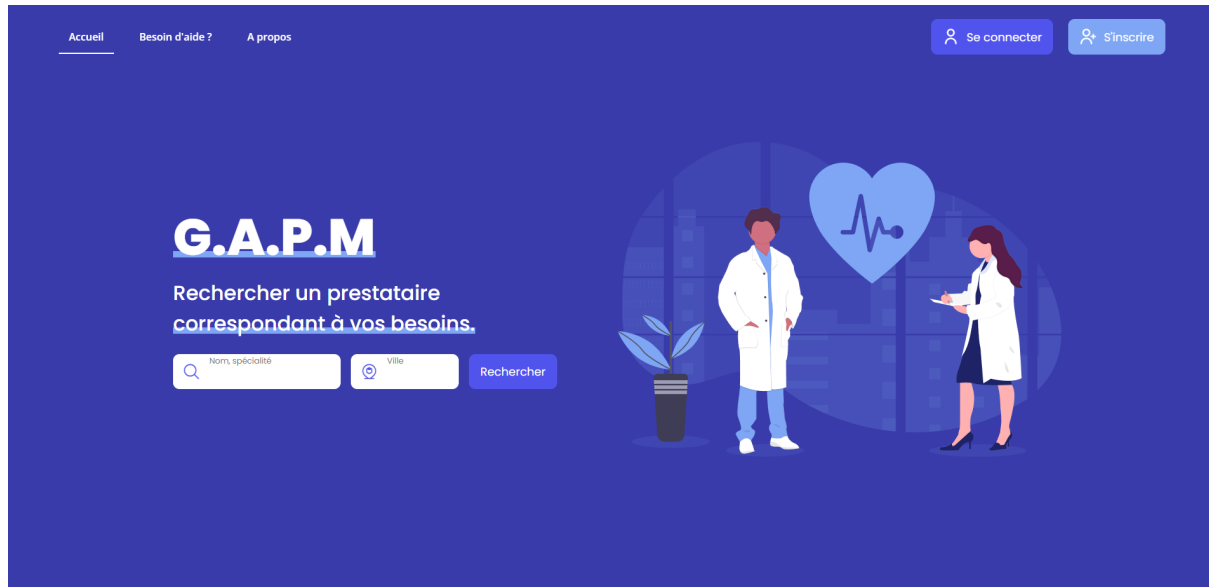
    {% if not userLogged %}
      <i id="popUp-connexion" class="popup-closed">{% include 'includes/popUpConnexion.twig' %}</i>
      <i id="popUp-inscription" class="popup-closed">{% include 'includes/popUpInscription.twig' %}</i>
    {% endif %}

    {% if userLogged %}
      <i id="popUp-prendre-RDV" class="popup-closed">{% include 'includes/popUpPrendreRDV.twig' %}</i>
      <i id="popUp-avis-RDV" class="popup-closed">{% include 'includes/popUpAvisRDV.twig' %}</i>
    {% endif %}

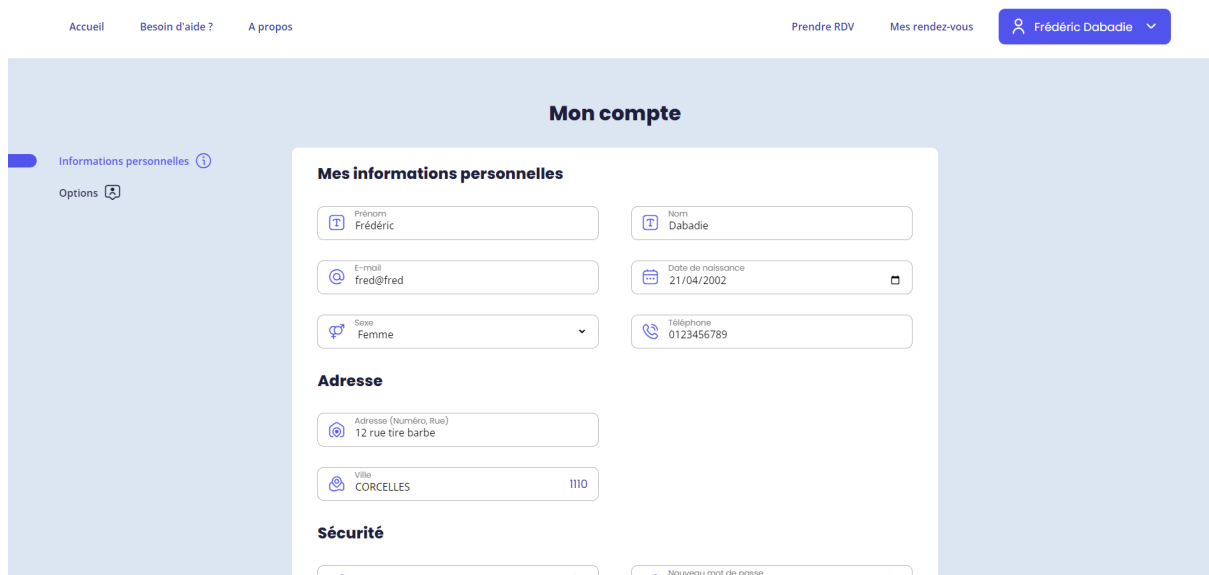
    <script src="{{ app.request.getSchemeAndHttpHost() }}/assets/prod/js/main.min.js"></script>
  </body>
</html>
```

Quelques images d'intégration

Home page :

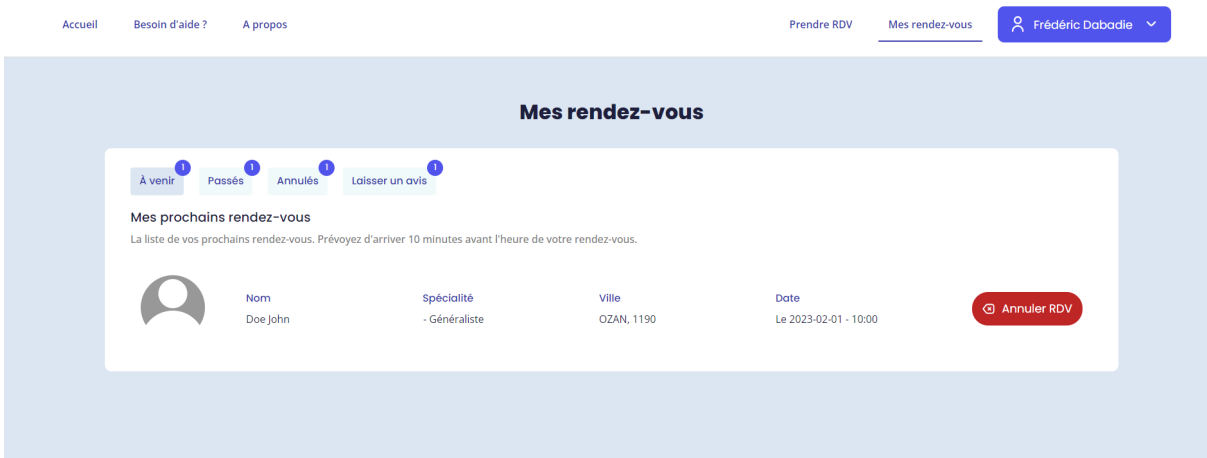


Page de mon compte :

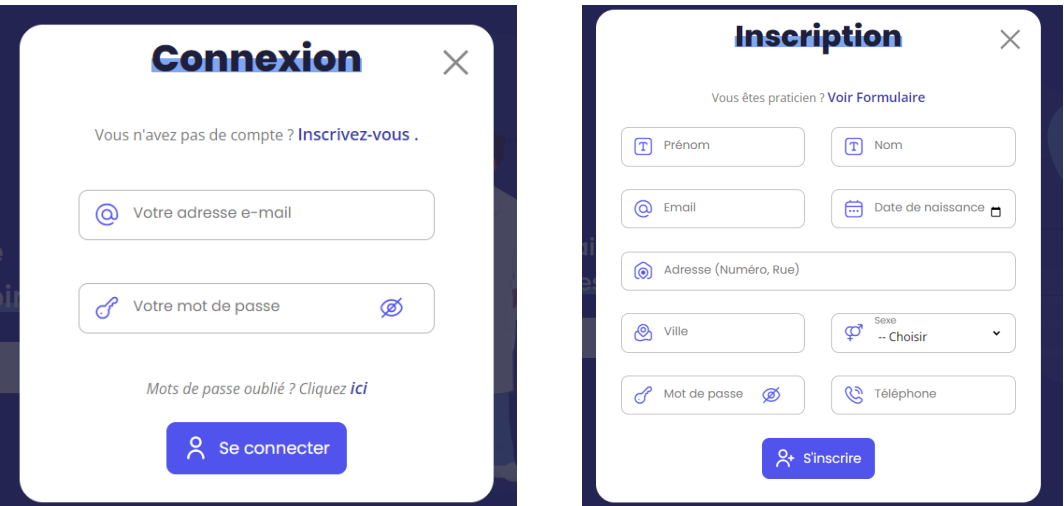


Sujet 2 – M. Gagné.

Page mes rendez-vous :



Popin de connexion / inscription :





Envoi de mail mot de passe oublié :



Sujet 2 – M. Gagné.

Envoi de mail refus d'un emprunt :

Search...





Validation d'un emprunt
to: <jack.black@example.com> a minute ago

Refus d'un emprunt
to: <jack.black@example.com> 2 minutes ago

Mot de passe oublié
to: <test@test.fr> 8 minutes ago

Refus d'un emprunt



From: No-reply <no-reply@gapm.com>
To: Black Jack <jack.black@example.com>

2023-06-07 11:46, 616 Bytes

Show Headers

HTML

HTML Source

Text

Raw

Spam Analysis




Tech Info

Votre demande d'emprunt de vehicule du 2023-06-07 au 2023-06-30 pour une Renault Mégane a été refusé par un administrateur.

Pour voir vos emprunts, cliquez ici : Voir les emprunts

Envoi de mail confirmation d'un rendez-vous :

Search...



Confirmation d'un rendez-vous
to: <jack.black@example.com> a few seconds ago


Confirmation d'un rendez-vous
to: <jack.black@example.com> 2 days ago

Confirmation d'un rendez-vous
to: <jack.black@example.com> 2 days ago

Confirmation d'un rendez-vous
to: <jack.black@example.com> 2 days ago

Validation d'une note de frais
to: <jack.black@example.com> 5 days ago

Confirmation d'un rendez-vous



From: No-reply <no-reply@gapm.com>
To: Morgat Mathieu <morgatmathieu@gmail.com>, Black Jack <jack.black@example.com>

2023-06-12 07:06, 674 Bytes

Show Headers

HTML

HTML Source

Text

Raw

Spam Analysis

Tech Info

Le rendez-vous du 2023-06-12 de 10:00 à 10:30, demandé par Mathieu Morgat avec Jack Black, a bien été pris en compte.

Pour voir vos rendez-vous, cliquez ici : Voir mes rendez-vous

Difficultées rencontrés

Les principaux problèmes rencontrés durant le projet sont les suivants :

1. Gestion de projet : Nous avons constaté une mauvaise organisation au niveau de la gestion de projet. Il n'y avait pas de personne dédiée à temps plein pour prendre en charge cette responsabilité, ce qui a entraîné des difficultés dans le suivi et la coordination des tâches.
2. Modélisation initiale : Nous avons rencontré des difficultés au début du projet pour réaliser une modélisation adéquate. Par la suite, nous avons réalisé que certaines choses n'étaient pas correctement définies dès le départ, ce qui a entraîné des ajustements et des modifications ultérieures.
3. Compréhension de l'ORM Doctrine : Au début, nous avons rencontré des difficultés dans la compréhension et l'utilisation de l'ORM Doctrine. Cependant, une fois que nous avons acquis une bonne maîtrise de cet outil et que nous l'avons pratiqué régulièrement, nous avons pu surmonter ces difficultés et progresser efficacement dans le développement du projet.

Il est important de noter que malgré ces problèmes, nous avons su faire face et les surmonter avec succès, ce qui a contribué à l'amélioration de notre apprentissage et de notre expérience dans le cadre de ce projet. Ces obstacles ont également permis de tirer des leçons importantes en termes de planification, de modélisation et d'adaptation aux outils technologiques.

Bilan

En termes de réalisations, notre équipe a réussi à créer une version fonctionnelle de l'application G.A.P.M en web. Nous avons mis en place les principales fonctionnalités telles que la recherche de praticiens, la prise de rendez-vous en ligne, la gestion des disponibilités, la gestion des profils patients et praticiens, ainsi que la gestion des rendez-vous.

Ce projet nous a permis d'approfondir nos compétences techniques en utilisant des technologies clés du développement web telles que HTML, JavaScript, SCSS, Webpack, Composer et Doctrine ORM. Nous avons acquis une meilleure compréhension de ces technologies et avons pu les appliquer concrètement dans la réalisation du projet. De plus, nous avons appris à intégrer et à utiliser ces différentes technologies de manière cohérente pour créer une application web robuste et fonctionnelle.

Sur le plan du travail de groupe, nous avons développé des compétences essentielles en matière de collaboration et de communication. La répartition des tâches a été effectuée de manière efficace, permettant à chaque membre de l'équipe de contribuer à la réalisation du projet. Nous avons également fait preuve d'une bonne coordination et d'une communication régulière pour résoudre les problèmes, partager les idées et s'assurer que chacun est sur la même longueur d'onde.

Ce projet nous a également confronté à des défis et des obstacles auxquels nous avons dû faire face ensemble. Nous avons développé notre capacité à résoudre les problèmes de manière collaborative, en trouvant des solutions efficaces et en adaptant notre approche lorsque cela était nécessaire. Nous avons également appris à gérer notre temps de manière judicieuse, en respectant les délais fixés et en nous adaptant aux contraintes du projet.

En conclusion, ce projet de développement web a été une expérience enrichissante à la fois sur le plan technique et sur le plan du travail d'équipe. Nous avons réussi à créer une version fonctionnelle de l'application en utilisant différentes technologies web, ce qui a renforcé nos compétences en développement web. De plus, notre travail de groupe a été efficace, mettant en valeur notre collaboration et notre capacité à relever des défis. Nous sommes fiers de ce que nous avons accompli et nous sommes ouverts à toute suggestion d'amélioration pour continuer à progresser dans notre travail de groupe et pour faire évoluer notre application.

Perspectives futures

Le projet G.A.P.M a atteint de nombreuses fonctionnalités clés, mais il existe également des perspectives futures pour améliorer davantage l'application. Voici quelques idées de développement :

- **Cessation d'activité pour les intervenants** : Une fonctionnalité importante à envisager serait d'ajouter la possibilité pour les intervenants de mettre fin à leur activité. Cela permettrait aux praticiens de signaler leur cessation d'activité de manière claire et précise, garantissant ainsi que les demandeurs ne prennent pas de rendez-vous avec des praticiens inactifs.
- **Gestion des fiches de paie** : Une amélioration potentielle consisterait à intégrer un système de gestion des fiches de paie pour les intervenants. Cela faciliterait le suivi et le calcul des revenus, simplifiant ainsi la gestion financière pour les praticiens.
- **Localisation des praticiens sur leur profil (utilisation de l'API Google Maps)** : Une fonctionnalité intéressante serait d'intégrer une carte interactive sur les profils des praticiens, en utilisant l'API Google Maps. Cela permettrait aux demandeurs de visualiser l'emplacement géographique des praticiens, facilitant ainsi le choix d'un praticien en fonction de sa proximité géographique.
- **Système de cabinet ou de mode libéral, avec choix du type et du jour de rythme** : Une amélioration potentielle consisterait à mettre en place un système permettant aux intervenants de spécifier s'ils exercent en cabinet ou en mode libéral. De plus, il serait intéressant de permettre aux praticiens de définir leurs jours de disponibilité ou

de rythme de travail, offrant ainsi une plus grande flexibilité dans la prise de rendez-vous.

En explorant ces perspectives d'amélioration, le projet G.A.P.M pourrait continuer à évoluer et à répondre aux besoins changeants des intervenants et des demandeurs. Ces développements futurs permettraient de renforcer l'application et d'améliorer l'expérience globale des utilisateurs.

Annexes

Définitions

ORM (Object-Relational Mapping) : L'ORM est un acronyme pour Object-Relational Mapping, qui est une technique de programmation permettant de mapper les objets d'une application avec les tables d'une base de données relationnelle. Il facilite l'accès et la manipulation des données en fournissant une interface objet pour interagir avec la base de données, en évitant ainsi d'écrire des requêtes SQL directement.

Moteur de template : Un moteur de template est un outil logiciel qui permet de générer des documents dynamiques à partir de modèles pré-définis. Il facilite la séparation entre la logique de présentation et la logique métier d'une application en permettant d'insérer des variables, des boucles, des conditions, et d'autres fonctionnalités dans le modèle. Le moteur de template interprète le modèle et génère le document final en remplaçant les parties variables par les valeurs appropriées.

Github : Github est une plateforme web qui offre des fonctionnalités de gestion de versions et de collaboration pour les projets de développement de logiciels. Il utilise le système de contrôle de version Git, permettant aux développeurs de partager, de collaborer et de suivre les modifications apportées au code source d'un projet. Github offre également des fonctionnalités telles que le suivi des problèmes, les demandes de fusion (pull requests) et l'hébergement de sites web statiques.

Lien du dépôt du projet : [GitHub](#)

Scénarios

Inscription d'un visiteur

Identification

But : Décrire les étapes permettant à l'acteur principal de s'inscrire au site web en tant que demandeur de services.

Acteur principal : Visiteur

Acteur secondaire : Aucun

Portée : Application GAPM

Intervenant : Entreprise demandant l'application (client)

Garantie en cas de succès : Le compte est enregistré en bdd et le Visiteur obtient accès au reste des fonctionnalités du site mis à disposition aux demandeurs de service.

Objectif : Inscription au site d'un visiteur.

Pré-requis : Aucun

Séquencement

Le cas d'utilisation commence lorsqu'un visiteur décide de s'inscrire sur le site.

Préconditions

Aucunes

Enchaînement nominal :

Sujet 2 – M. Gagné.

- (0. Le visiteur choisit de s'inscrire sur le site)
- 1. Le système demande des informations nécessaires pour l'inscription sur la personne.
- 2. Le visiteur remplit les informations et confirme.
- 3. Le système demande si le visiteur souhaite devenir un intervenant ou un demandeur de service.
- 4. Le visiteur indique qu'il souhaite devenir un demandeur de services et confirme.
- 5. Le système envoie une confirmation par e-mail à la personne.
- 6. Le visiteur confirme son e-mail et se connecte à son compte.
- 7. Le site redirige vers la page d'accueil et il peut ainsi naviguer sur le site.

Enchaînements d'exception

- 2.1 L'e-mail ou le mot de passe entrés par le visiteur ne sont pas valides, ou ne correspondent pas aux normes.
- 2.2 Le système indique au visiteur de rectifier les champs demandés.
- 2.3 L'utilisateur rectifie les informations et confirme.
- 6.1 Le visiteur ne valide pas son compte à temps et le lien expire.
- 6.2 Le système exige à l'utilisateur de vérifier son e-mail.
- 6.3 L'utilisateur appuie sur « Ré-envoyer une confirmation »

Enchaînements alternatifs

- 4 L'utilisateur indique qu'il souhaite devenir intervenant.
- 5 Le système continue avec le scénario pour l'inscription en tant qu'intervenant.

Modifier son profil

Identification

But : Décrire les étapes permettant à l'acteur principal de modifier son profil sur le site web en tant que demandeur de services.

Acteur principal : Demandeur de service

Acteur secondaire : Aucun

Portée : Application GAPM

Intervenant : Entreprise demandant l'application (client)

Garantie en cas de succès : Les modifications apportées au compte sont prises en compte par le système.

Objectif : Modification des informations du compte d'un demandeur de service

Pré-requis : Être connecté en tant que demandeur de services.

Enchaînement nominal :

- (0. Le demandeur choisit de modifier son profil)
1. Le demandeur clique sur son icône de profil
2. Le système lui affiche son profil
3. Le demandeur clique sur « modifier les informations »
4. Le système affiche des champs modifiables.
5. Le demandeur remplit les champs à changer puis appuie sur le bouton « enregistrer »
6. Le système enregistre les modifications

7. Le système retourne un message de succès

Enchaînements d'exception

5.1 Certaines informations ne sont pas conformes

5.2 Le système indique au visiteur de rectifier les champs demandés.

5.3 L'utilisateur rectifie les informations et confirme.

Commenter service reçu

Identification

But : Décrire les étapes permettant à l'acteur principal de commenter un service reçu en tant que demandeur de services.

Acteur principal : Demandeur de service

Acteur secondaire : Intervenant

Portée : Application GAPM

Intervenant : Entreprise demandant l'application (client)

Garantie en cas de succès : Le commentaire du demandeur de service sera pris en compte par le système

Objectif : Commenter un service reçu par un intervenant

Pré-requis : Être connecté en tant que demandeur de services et avoir reçu un service

Enchaînement nominal :

(0. Le demandeur choisit de commenter un service reçu)

Sujet 2 – M. Gagné.

1. Le demandeur choisit un service reçu
2. Le système lui affiche un modal de commentaire
3. Le demandeur remplit la zone de texte et confirme
4. Le système stocke le commentaire et ferme le modal.

Recherche d'un service dans l'historique

Identification

Nom du cas d'utilisation : « Recherche d'un service dans l'historique »

But : décrire les étapes permettant au demandeur de chercher un service

Acteur principal : demandeur de services

Acteur secondaire : néant

Portée : application GAPM

Intervenant : Entreprise demandant l'application (client)

Garantie en cas de succès : les détails du service reçu sont affichés

Séquencement

Le cas d'utilisation commence lorsqu'un demandeur de services interagit pour accéder à son historique des services

Préconditions

Le demandeur de services est déjà connecté/identifié

Enchaînement nominal :

1. Le système affiche l'historique des services (informations affichées : id du service reçu, date, nom de l'intervenant, type de service reçu, statu : « À venir » / « Passé » / « Annulé » / « Raté ») ainsi que les options de recherche d'un service (par date, par type de service, par nom d'intervenant)

Sujet 2 – M. Gagné.

2. Le demandeur de services choisit « Recherche par date »
3. Le système affiche un champ pour l'entrée de la date
4. Le demandeur de services renseigne une date
5. Le système affiche une liste contenant tous les services avec cette date (informations affichées : voir point 1 de la séquence nominale)
6. Le demandeur de services choisit un service
7. Le système affiche une fiche contenant les détails du service

Enchaînements alternatifs

A1 : Le demandeur de services choisit une recherche par type de service

L'enchaînement démarre après le point 1 de la séquence nominale :

2. Le demandeur de services choisit « Recherche par type de service »
3. Le système affiche tous les types de services offerts
4. Le demandeur de services choisit un type de service
5. Le système affiche une liste contenant tous les services de ce type (informations affichées : voir point 1 de la séquence nominale)

La séquence nominale reprend au point 6

A2 : Le demandeur de services choisit une recherche par nom de l'intervenant

L'enchaînement démarre après le point 2 de la séquence nominale :

2. Le demandeur de services choisit « Recherche par nom de l'intervenant »
3. Le système affiche un menu avec toutes les lettres de l'alphabet
4. Le demandeur de services choisit une des lettres affichées

Sujet 2 – M. Gagné.

5. Le système affiche une liste contenant tous les intervenants avec un nom commençant par la lettre choisie par l'utilisateur.
6. Le demandeur de services choisit un intervenant
7. Le système affiche une liste avec tous les services demandés par cet utilisateur et en même temps offerts par l'intervenant choisit (informations affichées : voir point 1 de la séquence nominale)

Commenter service reçu

Identification

Nom du cas d'utilisation : « Commenter service reçu »

But : décrire les étapes permettant au demandeur de services de commenter un service reçu après utilisation, en mode d'avis sur ceci

Acteur principal : demandeur de services

Acteur secondaire : néant

Portée : application GAPM

Intervenant : Entreprise demandant l'application (client)

Garantie en cas de succès : le commentaire est sauvegardé dans la base de données

Séquencement

Le cas d'utilisation commence lorsqu'un demandeur de services interagit avec le système pour la création du commentaire

Préconditions

Le demandeur de services est déjà connecté/identifié

Le service sélectionné a un statut « Passé » et n'a pas été déjà commenté

Enchaînement nominal :

1. Le demandeur de services demande l'ajout d'un commentaire sur le service sélectionné
2. Le système vérifie que le service sélectionné ait un statut « Passé » et qu'il n'ait pas été déjà commenté
3. Le système renvoie le demandeur de service vers le formulaire de création de commentaires
4. Le demandeur de services saisit son commentaire (renseigne une note sur 5 et rédige son commentaire)

Sujet 2 – M. Gagné.

5. Le demandeur de services valide son commentaire
6. Le système vérifie qu'au moins une des informations demandées ont été renseignées.
7. Le système sauvegarde les informations renseignées
8. Le système renvoie l'un message informant que les informations transmises ont été sauvegardées avec succès
9. Le système redirige l'utilisateur vers la fiche du service commenté

Enchaînements alternatifs

A1 : Le commentaire ne contient pas d'informations

L'enchaînement démarre après le point 6 de la séquence nominale :

7. Le système renvoie l'un message informant au demandeur de services que le commentaire n'a pas été sauvegardé car au moins une des informations demandées doit être renseigné (note et/ou commentaire)

La séquence nominale reprend au point 3

Enchaînements d'exception

E1 : Le demandeur de services choisit un commentaire n'a pas un statut « Passé » et/ou il a été déjà commenté

L'enchaînement démarre après le point 2 de la séquence nominale :

3. Le système informe au demandeur de services que l'enregistrement de commentaires pour le service sélectionné est indisponible car son statut n'est pas « Passé » ou parce qu'il a été déjà commenté
4. Le système sollicite au demandeur de services de sélectionner un service qui remplisse ces conditions
5. Appel du cas « Recherche d'un service dans l'historique » au point 5

Accéder aux informations de ses demandeurs de services

Identification

But : décrire les étapes permettant à un intervenant de changer les services qu'il propose

Acteur principal : intervenant

Portée : application GAPM

Garantie en cas de succès : la liste de service est mise à jour en base de données.

Séquencement

Le cas d'utilisation commence lorsqu'un intervenant interagit avec le système pour changer les services qu'il propose.

Préconditions

L'intervenant est déjà connecté/identifié

Enchaînement nominal :

1. L'intervenant choisit dans une liste les différents services qu'il veut pouvoir proposer
2. Le système vérifie que les services sélectionnés correspondent au profil de l'intervenant.
3. Le système enregistre en bdd les modifications et affiche un message de succès.

Enchaînements alternatifs

A1 : Les services sélectionnés ne sont pas compatibles avec le profil de l'intervenant

L'enchaînement démarre après le point 1 de la séquence nominale :

1. Le système réinitialise le formulaire et envoie un message expliquant qu'il ne peut pas sélectionner ces options à moins de changer son profil.

Effectuer une demande de voiture pro

Identification

But : décrire les étapes permettant à un intervenant de demander la location d'une voiture professionnelle.

Acteur principal : intervenant

Portée : application GAPM

Garantie en cas de succès : L'intervenant reçoit une confirmation que la demande est envoyée et sera traitée.

Séquencement

Le cas d'utilisation commence lorsqu'un intervenant interagit avec le système pour demander une voiture professionnelle.

Préconditions

L'intervenant est déjà connecté/identifié

Enchaînement nominal :

1. Le système affiche les critères nécessaires pour un formulaire de demande de voiture
2. L'intervenant remplit et envoie le formulaire
3. Le système envoie un mail/message à l'intervenant pour lui confirmer la bonne prise en compte de sa demande.

Créer des notes de frais

Identification

But : décrire les étapes permettant à un intervenant de créer des notes de frais

Acteur principal : intervenant

Portée : application GAPM

Garantie en cas de succès : Notes de frais créée(s) et lien vers des document pdf

Séquencement

Le cas d'utilisation commence lorsqu'un intervenant interagit avec le système pour créer une ou des notes de frais.

Préconditions

L'intervenant est déjà connecté/identifié

Enchaînement nominal :

1. Le système affiche les critères nécessaires pour un formulaire de demande de notes de frais
2. L'intervenant remplit et envoie le formulaire
3. Le système envoie un mail/message à l'intervenant pour lui confirmer la bonne prise en compte de sa demande.

Authentification des membres du personnel d'administration

Identification

But : Authentification des membres du personnel admin

Acteur principal : Personnel Administratif

Séquencement

Le cas d'utilisation commence quand l'utilisateur accède à l'interface de connexion du back office et qu'il dispose des identifiants.

Préconditions

Avoir des identifiants de personnel administratif.

Enchaînement nominal :

0. L'utilisateur arrive sur l'interface de connexion du Back-Office, il dispose d'identifiants)
1. Le système retourne un formulaire (identifiant, mot de passe)
2. L'utilisateur entre ses informations de connexion et appuie sur le bouton "Se connecter"
3. Le système retourne l'interface du Back-Office :
4. L'interface du Back-office est composée d'un Menu (à gauche) et du contenu relatif au menu (à droite)

Enchaînements alternatifs

1. L'utilisateur rentre des informations de connexions erronées.
2. Le système retourne un message indiquant que ses informations de connexion sont fausses.
3. L'utilisateur ne peut pas aller plus loin sur le site, il reste bloqué sur la page de connexion.

Gérer les demandes relatives aux planning des intervenants

Identification

But : Gérer les demandes relatives aux planning des intervenants.

Acteur principal : Personnel Administratif

Pré-requis : Être connecté à l'interface d'administration

Séquencement

Le cas d'utilisation commence quand l'utilisateur accède à l'interface de connexion du back office et qu'il dispose des identifiants.

Préconditions

Avoir des identifiants de personnel administratif.

Enchaînement nominal :

0. L'utilisateur choisi l'onglet de navigation sur le quel il souhaite aller (ici Gérer les demande relatifs à aux planning des intervenant)
1. Le système retourne une page affichant les différente demande de chaque intervenant
2. L'utilisateur peut rechercher une demande
3. L'utilisateur peut s' il le souhaite modifier le statuts d'avancements de la demande.
4. Sur chaque demande le système renvoie un bouton qui redirige sur la page de gestion des plannings des l'intervenant concernée
5. Le système renvoie un message de validation / erreur et réaffirme la demande courante .

Enchaînements alternatifs

1. Il peut rechercher une demande par un système de filtre et d'un champ de recherche
2. Le système retourne toutes les demandes selon les filtres/le champ de recherche.

Enchaînements alternatifs 2

1. L'utilisateur souhaite modifier les statuts.
2. Le système retourne une interface de modification du statut, selon la demande cliquée.

Gérer les informations personnelles des intervenants

Identification

Acteur : Personnel administratif

Séquencement

Le cas d'utilisation commence quand l'administrateur est connecté au back office.

Préconditions

Être connecté à l'interface d'administration.

Enchaînement nominal :

1. L'utilisateur choisi l'onglet de navigation sur le quel il souhaite aller
(Gérer les données personnelles des intervenants)
2. Le système retourne une page indiquant chaque intervenant et ses données personnelles

3. L'utilisateur a la possibilité de modifier ces informations en cliquant sur le bouton "Modifier"
4. Le système retourne une page pré-complétée avec les informations personnelles de l'intervenant cliqué.
5. L'utilisateur modifie et sauvegarde ses modifications
6. Le système met à jour les informations de l'intervenant dans la base de données.

Enchaînements alternatifs

- Les champs de données personnelles doivent respecter une certaine validité (exemple : numéro de téléphone, email, adresse).
1. L'utilisateur entre des informations n'ayant aucun rapport avec le champ.
 2. Le système retourne les erreurs et le format à adapter pour chaque champ.

Gérer les notes de frais

Identification

But : Gérer les notes de frais

Acteur principal : Personnel Administratif

Pré-requis : Être connecté à l'interface d'administration

Séquencement

Le cas d'utilisation commence quand l'utilisateur accède à l'interface de connexion du back office et qu'il dispose des identifiants.

Préconditions

Avoir des identifiants de personnel administratif.

Enchaînement nominal :

1. L'utilisateur choisi l'onglet de navigation sur le quel il souhaite aller (ici Gérer les notes de frais)
2. Le système retourne une page affichant les différentes demandes de notes de frais de chaque intervenant
3. L'utilisateur a la possibilité de voir la note de frais avec les différents documents justificatifs.
4. L'utilisateur a la possibilité de valider ou refuser la note de frais ces informations en cliquant sur le bouton "valider" ou "refuser"