

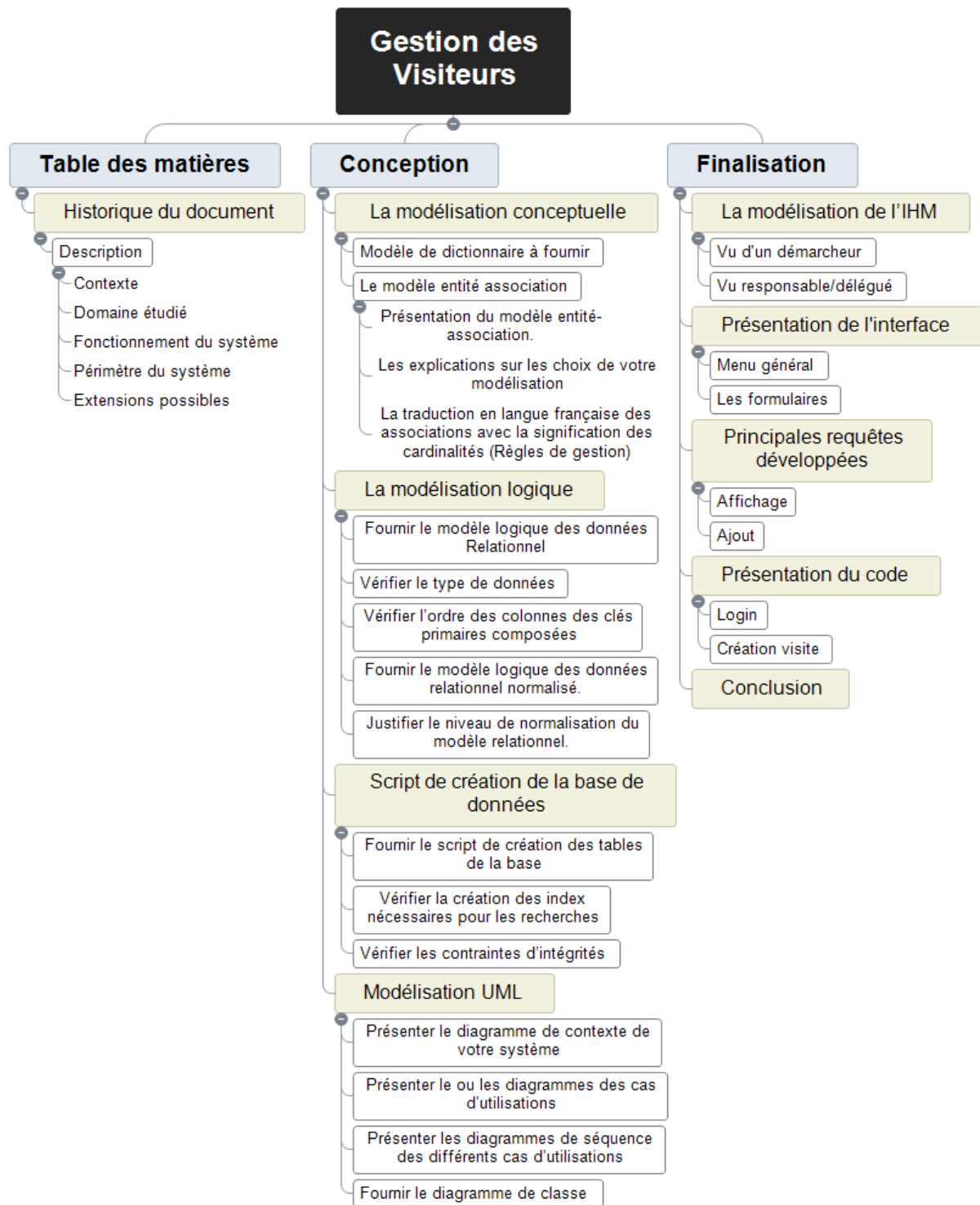
12/01/2024

Gestion des visites

Actualiser et rafraîchir la connaissance des
professionnels de santé sur les produits du
laboratoire

BRISSAC FLORIAN
GSB

GESTION DES VISITEURS



1	Table des matières	3
1.1	Historique du document.....	3
1.1.1	Description	5
2	Conception.....	6

2.1	La modélisation conceptuelle.....	6
2.1.1	Modèle de dictionnaire à fournir	6
2.1.2	Le modèle entité association.....	7
2.2	La modélisation logique	9
2.2.1	Fournir le modèle logique des données Relationnel.....	9
2.2.2	Fournir le modèle logique des données relationnel normalisé.	10
2.2.3	Justifier le niveau de normalisation du modèle relationnel.....	10
2.3	Script de création de la base de données	12
2.3.1	Fournir le script de création des tables de la base.....	12
2.3.2	Vérifier la création des index nécessaires pour les recherches	15
2.3.3	Vérifier les contraintes d'intégrités.....	16
2.4	Modélisation UML	16
2.4.1	Présenter le diagramme de contexte de votre système	16
2.4.2	Présenter le ou les diagrammes des cas d'utilisations.....	16
2.4.3	Présenter les diagrammes de séquence des différents cas d'utilisations.....	17
2.4.4	Fournir le diagramme de classe.....	18
3	Finalisation	19
3.1	La modélisation de l'IHM.....	19
3.1.1	Vu d'un démarcheur.....	20
3.1.2	Vu responsable/délégué.....	22
3.2	Présentation de l'interface	27
3.2.1	Menu général	27
3.2.2	Les formulaires	28
3.3	Principales requêtes développées.....	29
3.3.1	créer visite	29
3.3.2	liste demarcheurs	30
3.4	Présentation du code	31

1 TABLE DES MATIERES

1.1 HISTORIQUE DU DOCUMENT

Numéro de Versio	Date de Publication	Description des Modification	Responsable
V0	15/09/2023	debut du rappot, vreation page de garde	
V0.1	20/09/2023	structuration du rapport, sommaire	
V1	22/09/2023	relecture de la parie contexte, domaine etduie et fonction du système	
V2	22/09/2023	modelisation conceptuel du projet sur looping	
V2.1	06/10/2023	correction du model conceptuel, modele avec heritage de la classe employer	
V2.2	20/10/2023	conception du modele logique et explication	
V2.3	22/10/2023	justification de normalisation	
V2.3.1	23/10/2023	modification du modele relationnel. Corredction du modele	
V2.4	24/10/2023	creation de la base de donnée dans phpmyadmin	
V2.5	25/10/2023	verifivation des table, liaison des clefs entrangere, ajustement	
V2.6	26/10/2023	modification des nom du modele relationnel. Corredction du modele, ajustement	
V2.7	27/10/2023	justification de normalisation 3NF	
V2.8	28/10/2023	creation diagramme de contexte,	
V2.8.1	29/10/2023	modification du digramme de contexte et debut de diagramme de cas	
V2.8.2	30/10/2023	premiere version du diagrammes de sequence, ajustement diagramme de contexte et de cas d'utilisation	
V2.8.3	31/10/2023	explication diagramme de cas d'utilisation, modification du diagramme (couleurs, organisation)	

1.1.1 DESCRIPTION

1.1.1.1 CONTEXTE

Dans le cadre de ma formation en BTS SIO (Services Informatiques aux Organisations), mon projet consiste à développer une application pour gérer les visites des conseillers médicaux d'un laboratoire pharmaceutique. Les visites se font en France qui est divisé en 6 secteurs. Ces conseillers médicaux sont chargés de rencontrer des professionnels de la santé, tels que médecins et pharmaciens, afin de les informer sur les produits du laboratoire. Mon objectif est d'aider l'entreprise à harmoniser la gestion de ses conseillers médicaux suite à une fusion récente avec une autre entreprise.

1.1.1.2 DOMAINE ETUDIE

Le domaine que j'étudie concerne la gestion des visites médicales dans le secteur pharmaceutique. Cela implique la planification des visites, la collecte d'informations sur les professionnels de la santé, le suivi des interactions lors des visites, et la gestion des données régionales et sectorielles.

1.1.1.3 FONCTIONNEMENT DU SYSTEME

Le système que je vais développer sera une application informatique. Elle permettra aux conseillers médicaux en fonction de leurs rôles de planifier, modifier ou supprimer leurs visites, d'enregistrer des informations importantes sur les professionnels de la santé visités, de suivre leurs activités en temps réel, et de générer des rapports. Le système aidera également à organiser la hiérarchie des régions et des secteurs géographiques.

1.1.1.4 PERIMETRE DU SYSTEME

Dans le cadre de mon projet, je vais me concentrer sur les aspects suivants :

1. La création et la gestion des profils des conseillers médicaux. (add, sup, edit)
2. La planification des visites et la définition des objectifs.
3. La collecte de données sur les professionnels de la santé visités.
4. Le suivi des visites et la saisie de notes importantes.
5. La génération de rapports d'activité pour les conseillers médicaux.
6. L'organisation des données régionales et sectorielles pour une meilleure gestion.
7. Intégration de données sur les produits pharmaceutiques et leurs mises à jour (add, sup, edit).
8. L'envoi de notification par mail

1.1.1.5 EXTENSIONS POSSIBLES

À l'avenir, je pourrais envisager d'ajouter les fonctionnalités suivantes :

1. Analyse des données collectées pour obtenir des informations commerciales utiles.
2. Intégration de fonctionnalités de communication interne pour les conseillers médicaux.
3. Gestion des déplacements et des frais liés aux visites.
4. Gestion de location de véhicule et de co-voiturage.

Mon projet vise à améliorer l'efficacité et la coordination des activités des conseillers médicaux dans le but de favoriser les prescriptions des produits pharmaceutiques du laboratoire.

2 CONCEPTION

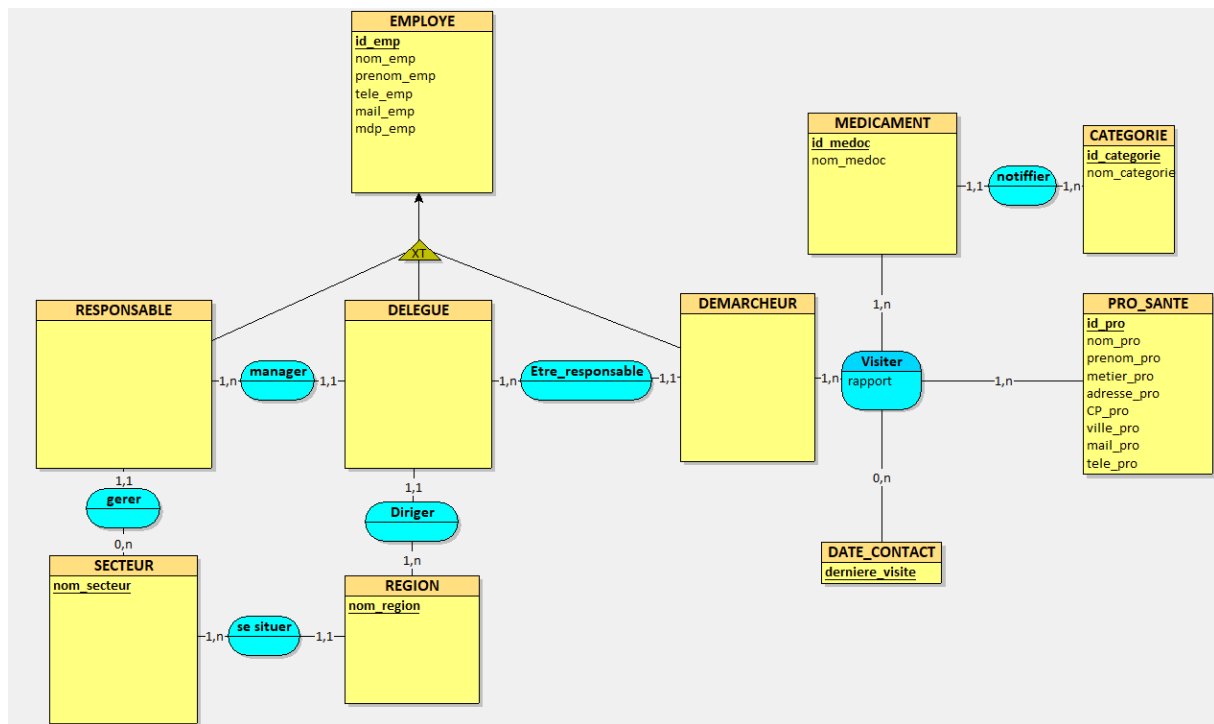
2.1 LA MODELISATION CONCEPTUELLE

2.1.1 MODELE DE DICTIONNAIRE A FOURNIR

Nom Conceptuel	Nom Logique	Type	Nature	Longueur	Remarque
id_emp	identifiant_employe	E	N	50	Identifiant
nom_emp	nom_employe	E	A	20	Nom
prenom_emp	prenom_employe	E	A	20	Prénom
tele_emp	telephone_employe	E	N	10	Téléphone
mail_emp	mail_employe	E	AN	30	Mail
mdp_emp	mdp_responsable	E	AN	20	Mot de passe
id_secteur	identifiant_secteur	E	N	2	Identifiant
nom_secteur		E	A	20	Nom
id_region	identifiant_region	E	N	10	Identifiant
nom_region		E	A	20	Nom
id_medoc	identifiant_medicament	E	N	10	Identifiant
nom_medoc	nom_medicament	E	A	50	Nom
id_categorie	identifiant_categorie	E	N	10	Identifiant
nom_categorie		E	A	20	Nom
id_pro	identifiant_professionnel_de_sante	E	N	10	Identifiant
nom_pro	nom_professionnel_de_sante	E	A	20	Nom
prenom_pro	prenom_professionnel_de_sante	E	A	20	Prénom
metier_pro	metier_professionnel_de_sante	E	A	20	Métier
adresse_pro	adresse_professionnel_de_sante	E	AN	100	Adresse
CP_pro	code_postale_professionnel_de_sante	E	N	5	
ville_pro	ville_professionnel_de_sante	E	A	20	
mail_pro	mail_professionnel_de_sante	E	AN	30	
tele_pro	tele_professionnel_de_sante	E	N	10	Téléphone
derniere_visite		E	DATE	30	
rapport		Co	A	20	Généré

2.1.2 LE MODELE ENTITE ASSOCIATION

2.1.2.1 PRESENTATION DU MODELE ENTITE-ASSOCIATION.



2.1.2.2 LES EXPLICATIONS SUR LES CHOIX DE VOTRE MODELISATION

SECTEUR :

Un SECTEUR est identifié par son nom, est géré par un seul RESPONSABLE de secteur, est composé d'une ou plusieurs REGION qui sont chacune dirigées par 1 DELEGUE régional qui sont gérées par le RESPONSABLE de secteur.

REGION :

Une REGION est identifiée par son nom, se situe dans 1 SECTEUR. Une REGION est dirigée par 1 DELEGUE régionale.

EMPLOYEE :

Un EMPLOYEE est identifié par un identifiant et possède les attributs suivants : id, nom, prénom, téléphone, mail, mot de passe. Les tables RESPONSABLE, DELEGUE et DEMARCHEUR hériteront des attributs d'EMPLOYEE.

RESPONSABLE :

Un RESPONSABLE est identifié par les attributs d'EMPLOYEE. Dirige 1 seul SECTEUR et gère 1 ou plusieurs DELEGUES régionaux qui se trouvent dans le SECTEUR.

DELEGUE :

DELEGUE est identifier par les attribue d'EMPLOYER. 1 DELGUE est responsable 1 ou plusieurs DEMARCHEUR. 1 DELEGUE dirige 1 seul REGION

DEMARCHEUR :

Une DEMARCHEUR est identifier par les attribue d'EMPLOYER. 1 DEMARCHEUR visite 1 ou plusieurs PRO_SANTE à 1 ou plusieurs DATE_CONTACT pour présenter 1 ou plusieurs MEDICAMENT.

PRO_SANTE :

Un PRO_SANTE est identifié par un identifiant et possède les attributs suivants : id, nom, prénom, métier, adresse, code postal, ville, téléphone et mail.

Plusieurs PRO_SANTE peut être visiter par plusieurs DEMARCHEUR à plusieurs dates. Il peut être visiter plusieurs fois à la DATE_CONTACT (la même journée) par 1 ou plusieurs DEMARCHEUR.

Durant la visite un DEMARCHEUR peut présenter 1 ou plusieurs MEDICAMENT

Pour chaque visite

MEDICAMENT :

Un MEDICAMENT est identifié par un identifiant et possède l'attribut de nom. 1 seul MEDICAMENT est notifier dans 1 CATEGORIE.

CATEGORIE :

Une CATEGORIE est identifier par un identifiant est possède l'attribut nom. Dans 1 seule CATEGORIE il y a 1 ou plusieurs MEDICAMEN qui y sont notifier

DATE_CONTACT :

DATE_CONTACT est identifier par un identifiant dernier_visite. Il y a plusieurs DEMARCHEUR peuvent visiter plusieurs PRO_SANTE à 1 ou plusieurs DATE_CONTACT.

Visiter :

DEMARCHEUR rend visite a 1 ou plusieurs PRO_SANTE pour lui présenter 1 ou plusieurs MEDICAMENT a 1 ou plusieurs DATE_CONTACT

2.1.2.3 LA TRADUCTION EN LANGUE FRANÇAISE DES ASSOCIATIONS AVEC LA SIGNIFICATION DES CARDINALITES (REGLES DE GESTION)

Manager :

Association entre RESPONSABLE et DELEGUE. Un RESPONSABLE peut être associé à plusieurs DELGUE (1.n)

Gérer :

Association entre RESPONSABLE et SECTEUR. Un RESPONSABLE peut être associé à un seul SECTEUR (1.1)

Se situer :

Association entre SECTEUR et REGION. Un SECTEUR peut être associé à plusieurs REGION (1.n)

Diriger :

Être responsable :

Notifier :

Association entre MEDICAMENT et CATEGORIE

- MEDICAMENT est associé à une seule CATEGORIE ; CATEGORIE peut être associée à plusieurs MEDICAMENT

Visiter :

2.2 LA MODELISATION LOGIQUE

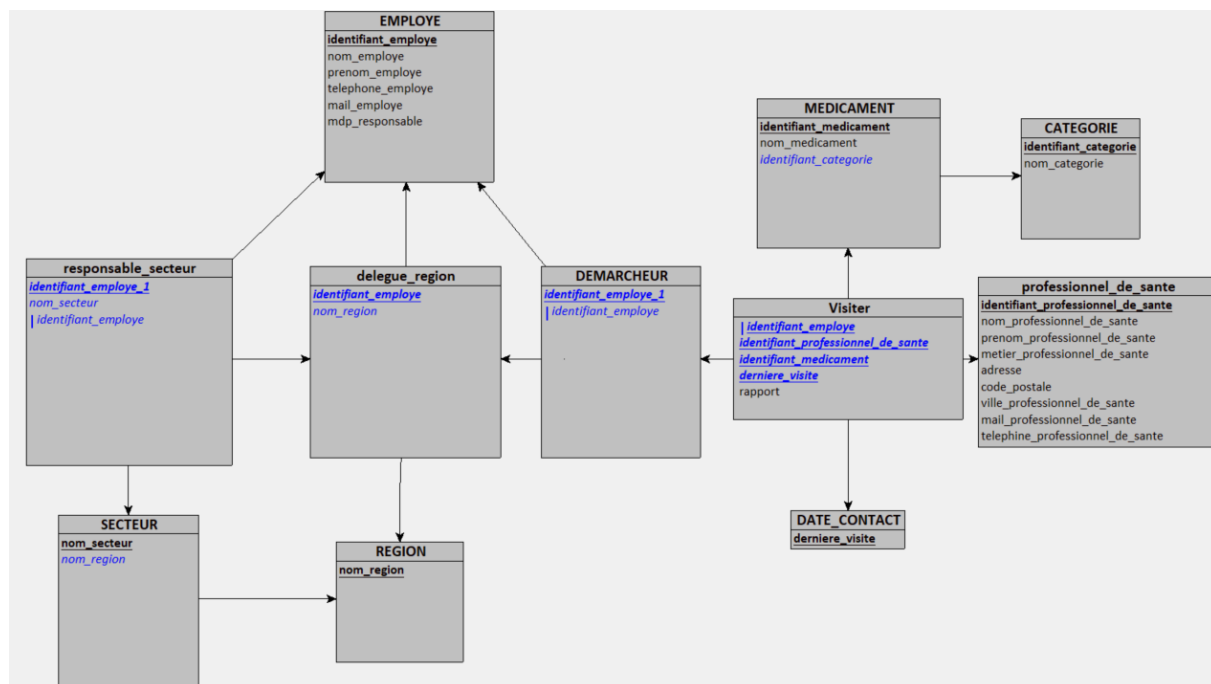
2.2.1 FOURNIR LE MODELE LOGIQUE DES DONNEES RELATIONNEL

PROFESSIONNEL_DE_SANTE = (identifiant professionnel de sante, nom_professionnel_de_sante, prenom_professionnel_de_sante, metier_professionnel_de_sante, adresse, code_postale INT, ville_professionnel_de_sante, mail_professionnel_de_sante, telephone_professionnel_de_sante) ;

MEDICAMENT = (identifiant medicament, nom_medicament, #identifiant_categorie) ;

CATEGORIE = (identifiant categorie, nom_categorie) ;

2.2.2 FOURNIR LE MODELE LOGIQUE DES DONNEES RELATIONNEL NORMALISE.



2.2.3 JUSTIFIER LE NIVEAU DE NORMALISATION DU MODELE RELATIONNEL.

La normalisation d'une base de données est un processus visant à organiser les données de manière à minimiser la redondance, à garantir l'intégrité référentielle et à faciliter la maintenance. Pour ce faire, j'ai divisé les données en plusieurs tables, en veillant à respecter certaines règles appelées "formes normales". Dans ce modèle, j'ai effectué une normalisation jusqu'à la troisième forme normale (3NF).

La table Professionnel_de_sante stocke les informations sur les professionnels de santé. Elle est en 3NF car chaque attribut ne dépend que de la clé primaire identifiant_professionnel_de_sante.

Par exemple, le nom et le prénom d'un professionnel de santé dépendent de son identifiant, mais ne sont pas influencés par d'autres attributs. Si un professionnel de santé change d'adresse, cela n'affecte que son enregistrement, ce qui simplifie la gestion des données.

La table Secteur contient des informations sur les secteurs d'activité. Elle est en 3NF car l'attribut nom_secteur est la clé primaire, et il n'y a pas de dépendance entre les autres attributs. Chaque secteur est défini de manière indépendante, sans influence mutuelle.

La table Date_Contact stocke les dates de contact. Elle est 3NF, car la clé primaire est l'attribut unique derniere_visite, et il n'y a pas de dépendances entre les autres attributs. Chaque date de contact est une entité distincte et n'a pas d'impact sur les autres.

La table *Categorie* représente les catégories de médicaments. Elle est en 3NF, car chaque catégorie est indépendante et n'est pas influencée par d'autres attributs. Par exemple, le nom de la catégorie ne dépend que de la clé primaire *identifiant_categorie*.

La table *Region* n'est pas en 3NF, car le nom de la région (attribut "*nom_region*") dépend directement de la clé primaire (*nom_region*), mais elle est également liée au nom du secteur par le biais de la clé étrangère (*nom_secteur*). Cela crée une dépendance transitive, car le nom de la région dépend du nom du secteur. Pour respecter la 3NF, il serait nécessaire de créer une nouvelle table pour stocker les associations entre les régions et les secteurs.

La table *Employe* est en 3NF car chaque employé est identifié de manière unique par son *identifiant_employe*, et les autres attributs (*nom*, *prénom*, *téléphone*, etc.) dépendent uniquement de cet identifiant. La clé primaire est *l'identifiant_employé*.

La table *medicament* n'est pas en 3NF car le nom du médicament (attribut "*nom_medicament*") dépend directement de la clé primaire (*identifiant_medicament*), mais il est également lié à la catégorie par le biais de la clé étrangère (*identifiant_categorie*). Cela crée une dépendance transitive, car le nom du médicament dépend de la catégorie à laquelle il appartient. Pour atteindre la 3NF, il serait nécessaire de créer une nouvelle table distincte pour les médicaments et leurs catégories.

La table "*Responsable_Secteur*" présente une dépendance transitive similaire. L'attribut "*nom_secteur*" dépend directement de la clé primaire (*identifiant_employe*), mais il est également lié au nom du secteur par le biais de la clé étrangère (*nom_secteur*), créant ainsi une dépendance transitive. Pour atteindre la 3NF, il faudrait également créer une nouvelle table pour gérer les relations entre les employés et les secteurs.

La table *Delegue_Region* est en 3NF car chaque délégué de région est identifié de manière unique par son *identifiant_employe_1*, et les liens vers son responsable de secteur et la région à laquelle il est assigné sont maintenus via des clés étrangères.

La table *Demarcheur* est en 3NF car chaque démarcheur est identifié de manière unique par son *identifiant_employe_1*, et son lien vers le délégué de région est maintenu par des clés étrangères.

La table Visiter représente les visites effectuées par les démarcheurs. Les données sont en 3NF car la clé primaire composite (identifiant_employe, identifiant_professionnel_de_sante, identifiant_medicament, derniere_visite) garantit que chaque visite est unique, et les attributs ne dépendent que de cette combinaison.

2.3 SCRIPT DE CREATION DE LA BASE DE DONNEES

2.3.1 FOURNIR LE SCRIPT DE CREATION DES TABLES DE LA BASE

```
CREATE TABLE professionnel_de_sante (  
    identifiant_professionnel_de_sante INT AUTO_INCREMENT,  
    nom_professionnel_de_sante VARCHAR (20),  
    prenom_professionnel_de_sante VARCHAR (20),  
    metier_professionnel_de_sante CHAR (20),  
    adresse VARCHAR (100),  
    code_postale INT,  
    ville_professionnel_de_sante CHAR (20),  
    mail_professionnel_de_sante VARCHAR (30),  
    telephone_professionnel_de_sante INT,  
    PRIMARY KEY(identifiant_professionnel_de_sante)  
);
```

```
CREATE TABLE SECTEUR(  
    nom_secteur CHAR (20),  
    PRIMARY KEY (nom_secteur)  
);
```

```
CREATE TABLE DATE_CONTACT (  
    derniere_visite DATETIME,  
    PRIMARY KEY (derniere_visite)  
);
```

```
CREATE TABLE CATEGORIE (  
    identifiant_categorie INT AUTO_INCREMENT,  
    nom_categorie VARCHAR (20),  
    PRIMARY KEY (identifiant_categorie)  
);
```

```
CREATE TABLE REGION (  
    nom_region VARCHAR (20),  
    nom_secteur CHAR (20) NOT NULL,  
    PRIMARY KEY (nom_region),  
    FOREIGN KEY (nom_secteur) REFERENCES SECTEUR (nom_secteur)  
);
```

```
CREATE TABLE EMPLOYE (  
    identifiant_employe INT AUTO_INCREMENT,  
    nom_employe VARCHAR (20),  
    prenom_employe VARCHAR (20),  
    telephone_employe VARCHAR (10),  
    mail_employe VARCHAR (30),  
    mdp_responsable VARCHAR (20),  
    PRIMARY KEY (identifiant_employe)  
);
```

```
CREATE TABLE MEDICAMENT (  
    identifiant_medicament INT AUTO_INCREMENT,  
    nom_medicament CHAR (50),  
    identifiant_categorie INT NOT NULL,  
    PRIMARY KEY (identifiant_medicament),  
    FOREIGN KEY (identifiant_categorie) REFERENCES CATEGORIE(identifiant_categorie)
```

);

```
CREATE TABLE responsable_secteur (  
    identifiant_employe INT,  
    nom_secteur CHAR (20) NOT NULL,  
    PRIMARY KEY (identifiant_employe),  
    FOREIGN KEY (identifiant_employe) REFERENCES EMPLOYE (identifiant_employe),  
    FOREIGN KEY (nom_secteur) REFERENCES SECTEUR (nom_secteur)  
);
```

```
CREATE TABLE delegue_region (  
    identifiant_employe_1 INT,  
    identifiant_employe INT NOT NULL,  
    nom_region VARCHAR(20) NOT NULL,  
    PRIMARY KEY (identifiant_employe_1),  
    FOREIGN KEY (identifiant_employe_1) REFERENCES EMPLOYE (identifiant_employe),  
    FOREIGN KEY (identifiant_employe) REFERENCES responsable_secteur  
(identifiant_employe),  
    FOREIGN KEY (nom_region) REFERENCES REGION (nom_region)  
);
```

```
CREATE TABLE DEMARCHEUR (  
    identifiant_employe_1 INT,  
    identifiant_employe INT NOT NULL,  
    PRIMARY KEY (identifiant_employe_1),  
    FOREIGN KEY (identifiant_employe_1) REFERENCES EMPLOYE (identifiant_employe),  
    FOREIGN KEY (identifiant_employe) REFERENCES delegue_region  
(identifiant_employe_1)  
);
```

```

CREATE TABLE Visiter (
    identifiant_employe INT,
    identifiant_professionnel_de_sante INT,
    identifiant_medicament INT,
    derniere_visite DATETIME,
    rapport VARCHAR (20),
    etat_visite VARCHAR (20),

    PRIMARY KEY (identifiant_employe, identifiant_professionnel_de_sante,
    identifiant_medicament, derniere_visite),

    FOREIGN KEY(identifiant_employe) REFERENCES
    DEMARCHEUR(identifiant_employe_1),

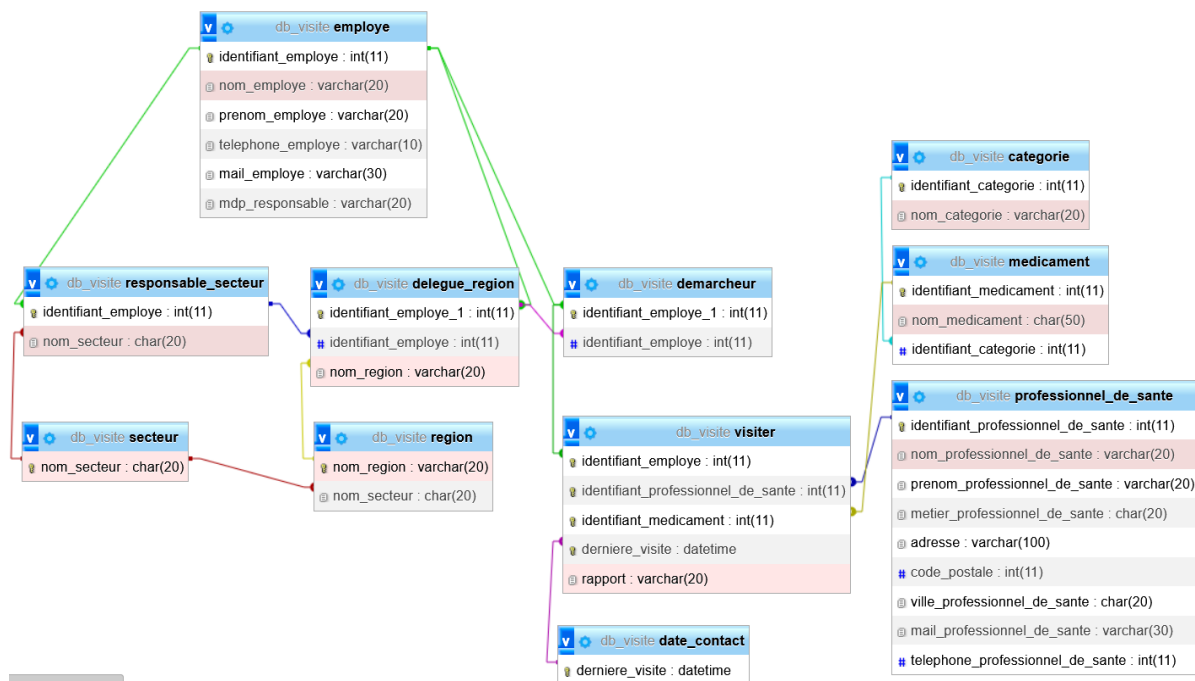
    FOREIGN KEY(identifiant_professionnel_de_sante) REFERENCES
    professionnel_de_sante(identifiant_professionnel_de_sante),

    FOREIGN KEY (identifiant_medicament) REFERENCES MEDICAMENT
    (identifiant_medicament),

    FOREIGN KEY(derniere_visite) REFERENCES DATE_CONTACT(derniere_visite)
);

```

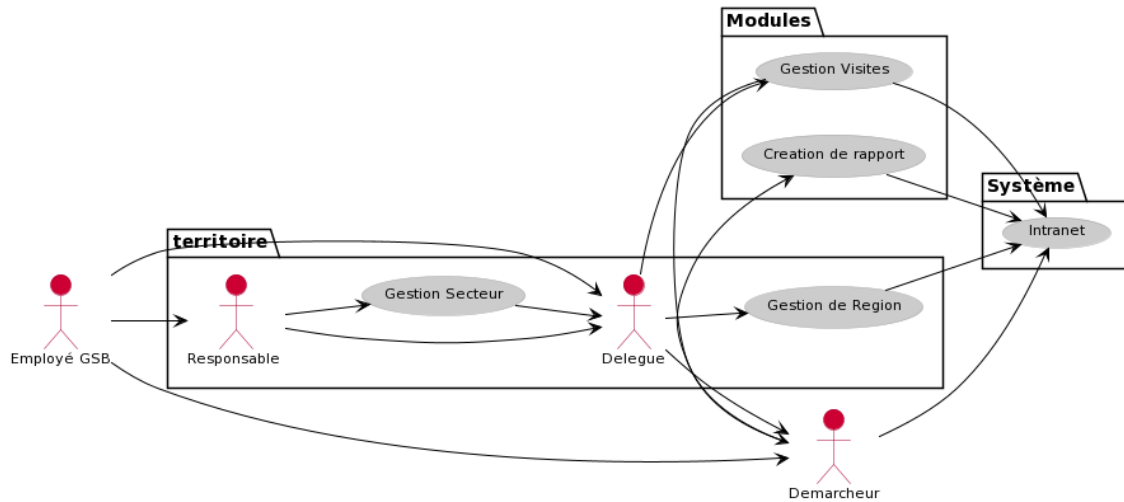
2.3.2 VERIFIER LA CREATION DES INDEX NECESSAIRES POUR LES RECHERCHES



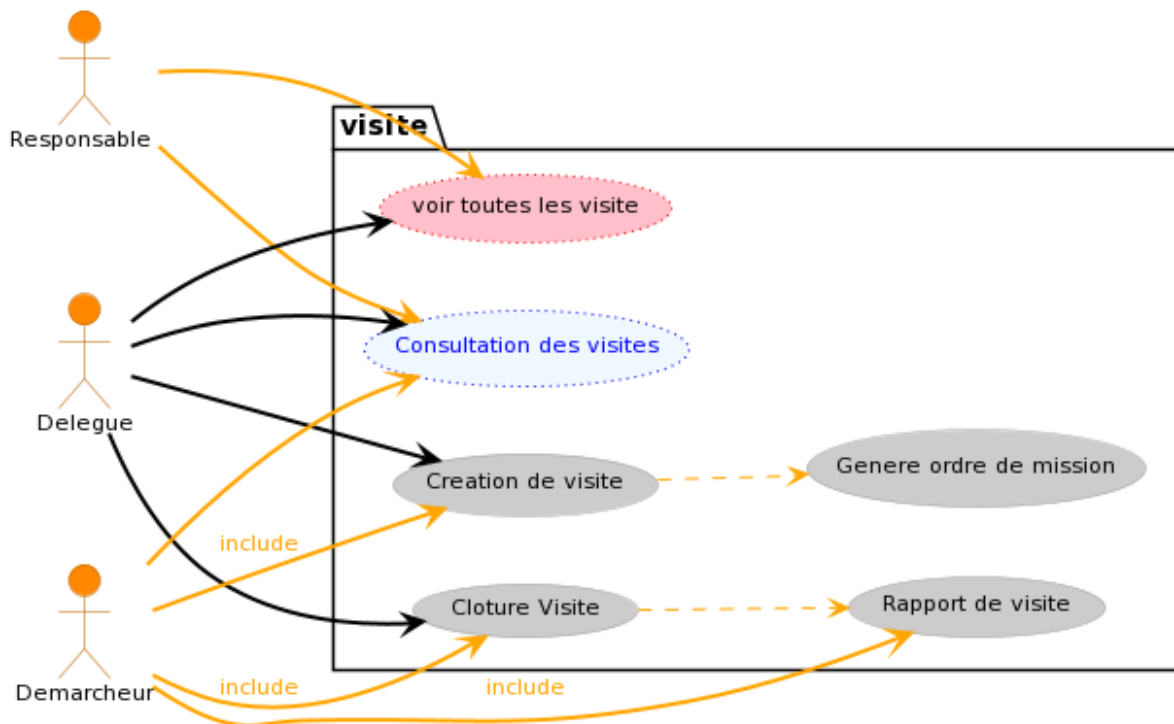
2.3.3 VERIFIER LES CONTRAINTES D'INTEGRITES

2.4 MODELISATION UML

2.4.1 PRESENTER LE DIAGRAMME DE CONTEXTE DE VOTRE SYSTEME



2.4.2 PRESENTER LE OU LES DIAGRAMMES DES CAS D'UTILISATIONS



Objectif :

Ce cas d'utilisation vise à mettre en lumière les fonctionnalités essentielles du système de gestion des visites spécifique à l'entreprise GSB, tout en présentant les différents niveaux d'accès des acteurs au sein du système.

Conditions :

L'accès à l'ensemble des fonctionnalités est sécurisé par le biais d'un système de connexion. Les autorisations et les niveaux d'accès sont configurés en fonction des rôles spécifiques de chaque employé au sein de l'entreprise.

Scénario :

- Le démarcheur, en tant que représentant commercial de GSB, a pour mission de rendre visite aux professionnels de santé afin de leur présenter des médicaments. Pour ce faire, il peut créer une visite en sélectionnant le professionnel de santé à visiter, les médicaments à présenter, et la date de la visite.
- De plus, le démarcheur peut accéder à un document d'ordre de mission qui détaille les autorisations et les conditions requises pour mener à bien sa mission. Son responsable est notifié par e-mail dès la création de la visite, lui fournissant ainsi toutes les informations nécessaires.
- Une fois la visite créée, le démarcheur peut la consulter à tout moment et la clôturer une fois qu'elle est achevée, en fournissant un rapport de visite.

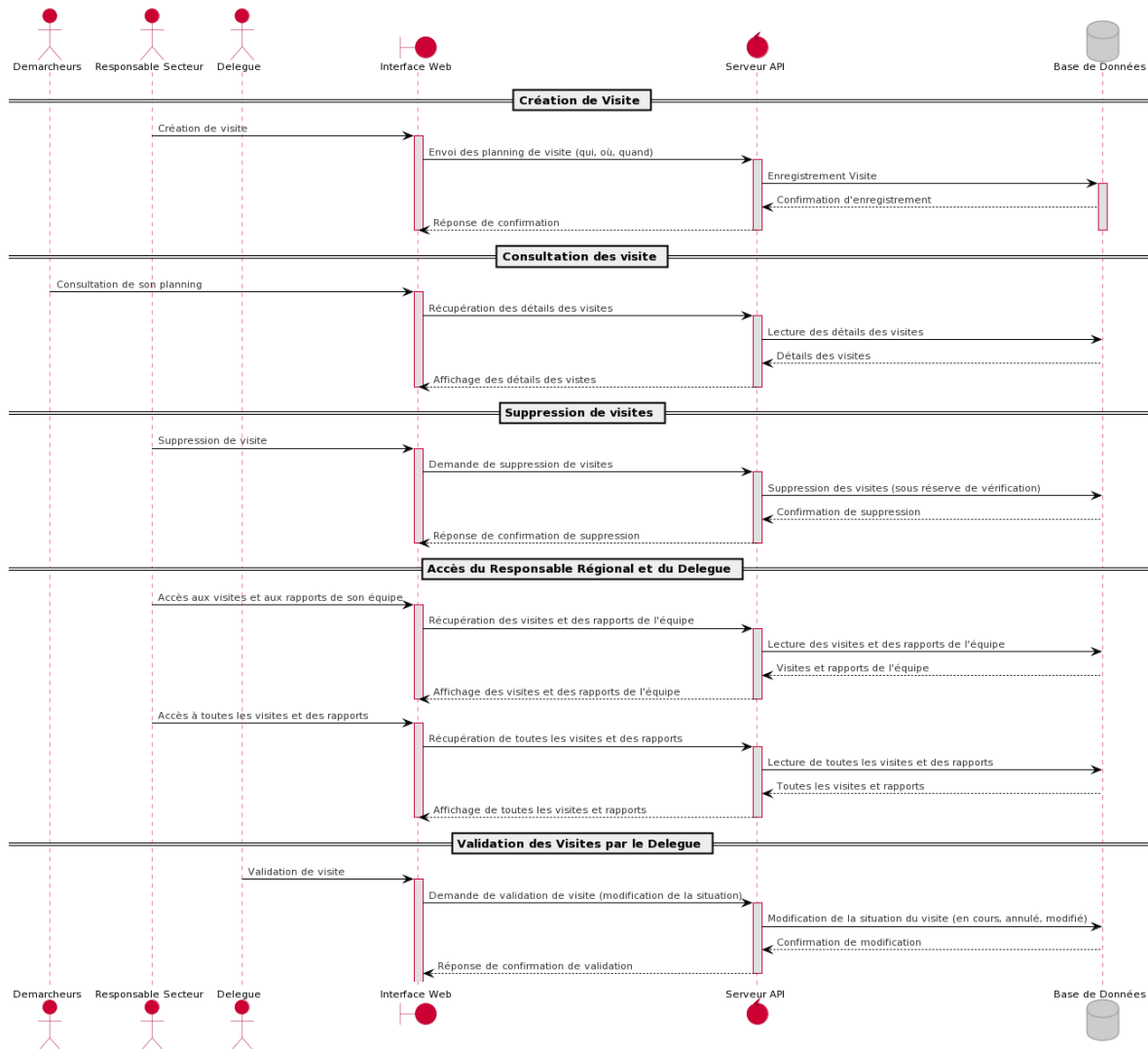
Le Responsable Régional :

- Le responsable régional supervise les délégués de son équipe pour atteindre les objectifs de l'entreprise, en veillant à leur performance, à leur motivation et à leur développement. Il dispose d'un accès complet à l'ensemble des visites effectuées par les démarcheurs.

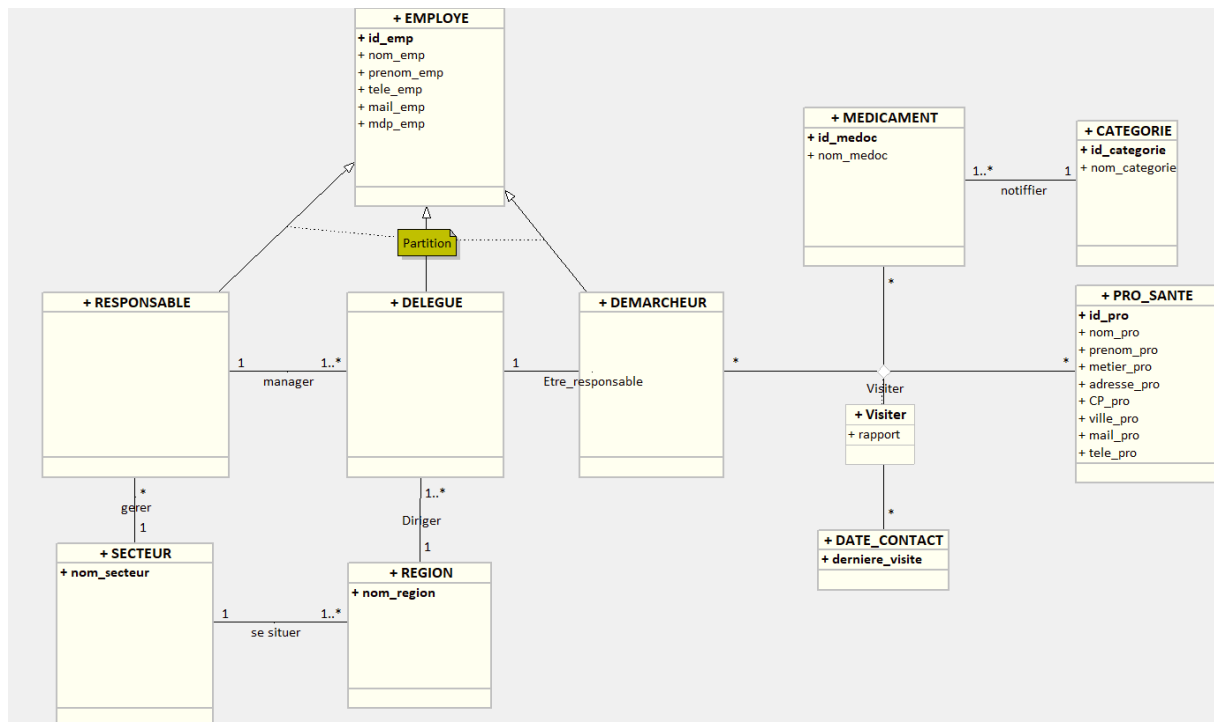
Le Délégué :

- Le délégué a un accès complet à toutes les visites effectuées par les démarcheurs de son équipe, lui permettant de suivre les visites réalisées et de consulter les rapports de clôture pour un suivi exhaustif.

2.4.3 PRESENTER LES DIAGRAMMES DE SEQUENCE DES DIFFERENTS CAS D'UTILISATIONS



2.4.4 FOURNIR LE DIAGRAMME DE CLASSE

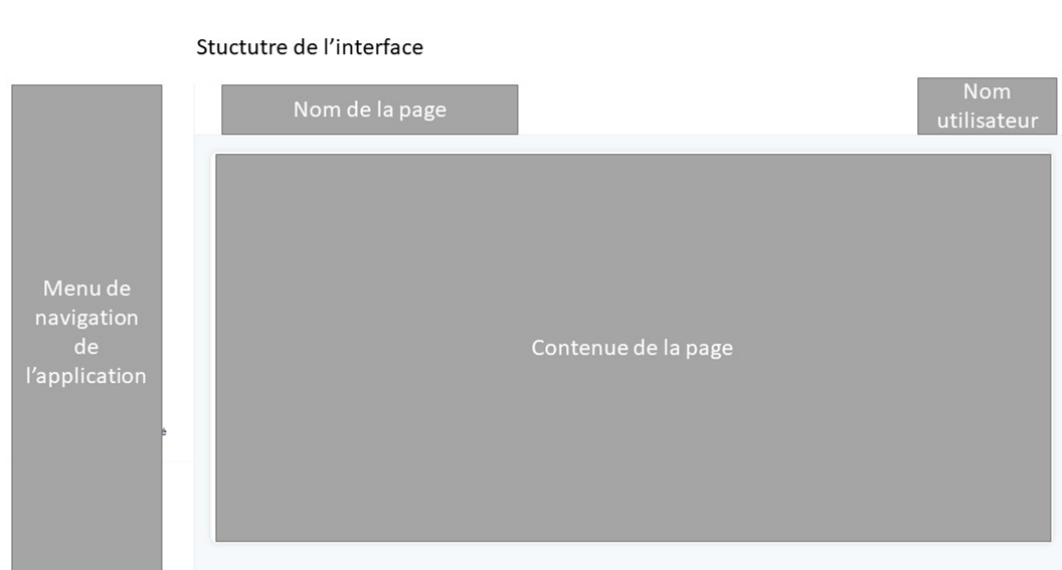


3 FINALISATION

3.1 LA MODELISATION DE L'IHM

Pour mon application web, j'ai choisi une architecture de base simple et uniforme. Cela simplifie l'expérience de l'utilisateur, lui permettant de rapidement prendre en main l'application et de retrouver une cohérence visuelle et fonctionnelle tout au long de la navigation.

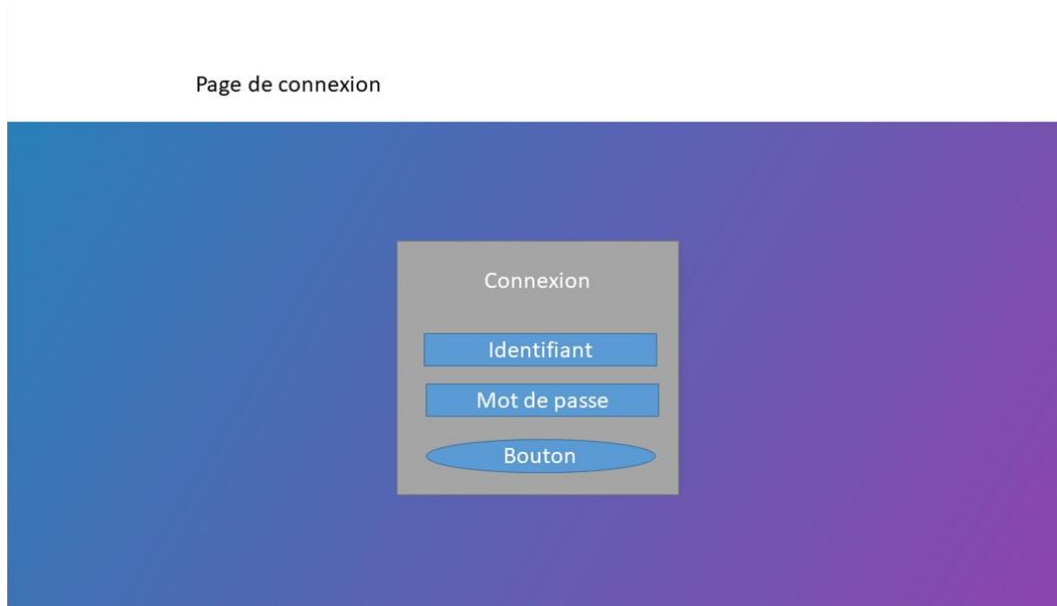
Maquette principale de l'application



Ainsi, voici l'interface homme-machine (IHM) de la structure principale de l'application. Elle comprend un menu de navigation, des informations sur la page actuelle et l'utilisateur connecté. Quant au contenu, il variera d'une page à une autre, pouvant prendre la forme de tableaux ou de formulaires selon les besoins spécifiques.

Page de connexion à l'application web :

Pour des raisons de sécurité chaque utilisateur doit se connecter l'aide de son adresse mail et de son mot de passe



L'utilisateur pourra interagir avec 2 champs texte et un bouton afin de valider et se connecter

3.1.1 VU D'UN DEMARCHEUR



Lorsqu'un démarcheur se connecte, il arrive sur la page d'accueil avec un message personnalisé de bienvenue. Son menu de navigation est composé de 5 boutons et une étiquette avec son nom d'utilisateur est visible en haut à droite.

Le démarcheur a accès à :

Vu d'un démarcheur pour ces visites



- Accueil
- Liste visites
- Liste médicaments
- Liste Professionnels
- Déconnexion

127.001.4000/visites

Nom utilisateur

Mes visites

Total visites

Date visite	Heure	Adresse	Médicament	Professionnel	Statut de la visite	Rapport

Un tableau récapitulatif de ces visites,

Listes des médicaments

Nom utilisateur

Les Médicaments

Nombre de médicament : 5

Fonction de recherche dans le tableau

Rechercher...

Nom	Catégorie

Bouton de navigation de l'application

La liste des médicaments commercialisé par GSB,

Listes des Professionnels de la santé

Nom utilisateur

Professionnels Santé Nombre : 11

Rechercher...  Fonction de recherche dans le tableau

Nom prénom	Métier	Adresse	Code Postale	Ville	Mail	Téléphone

Bouton de navigation de l'application

Et à la liste des professionnels de santé.

3.1.2 VU RESPONSABLE/DELEGUE

Vu d'un responsable ou d'un délégué

 Bouton de navigation

Accueil

Liste visites

Ajout visite

Liste démarcheurs

Ajout démarcheur

Liste médicaments

Ajout médicament

Liste Professionnels

Ajout Professionnel

Déconnexion

Accueil


Nom utilisateur

Message de bienvenu + nom de la personne connecté

Lorsqu'un délégué se connecte, il arrive sur la page d'accueil avec un message personnalisé de bienvenue. Son menu de navigation est composé de 10 boutons et une étiquette avec son nom d'utilisateur est visible en haut à droite.

Le délégué a accès à :

Vu responsable/délégué de toutes les visites



- Accueil
- Liste visites
- Ajout visite
- Liste démarcheurs
- Ajout démarcheur
- Liste médicaments
- Ajout médicament
- Liste Professionnels
- Ajout Professionnel
- Déconnexion

Nom utilisateur

Mes visites

Total visites

Démarcheur	Date visite	Heure	Adresse	Médicament	Professionnel	Statut de la visite	Rapport

La liste de toutes les visites des démarcheurs qu'il a planifiée,

Formulaire de création d'un visite

Bouton de navigation de l'application

Nom utilisateur

Créer Visite

Région

Employé

Professionnel de Santé

Médicament

Date de visite

Créer

Au formulaire de création d'une nouvelle visite,

Liste des démarcheurs pour responsable/délégué

Bouton de navigation de l'application

Nom utilisateur

Démarcheurs Nombre : 6

Nom prénom	Mail	Téléphone

La liste de tous les démarcheurs de sa région,

Formulaire de création d'un démarcheur

Bouton de navigation de l'application

Nom utilisateur

Ajouter Démarcheur

Nom

Prénom

Téléphone

Mail

Mot de passe

Région

Attribution d'un région

Delegue de Région

Attribution d'un délégué

Créer

Au formulaire de création d'un nouveau démarcheur,

Listes des médicaments

Nom utilisateur

Les Médicaments Nombre de médicament : 5

Rechercher...  Fonction de recherche dans le tableau

Nom	Catégorie

Bouton de navigation de l'application

La liste de tous les médicaments commercialiser par GSB,

Formulaire de création d'un médicament

Nom utilisateur

Création Médicament

Nom

Categorie

Choix de la catégorie

Envoyer

Bouton de navigation de l'application

Au formulaire d'ajout d'un nouveau médicament à promouvoir,

Listes des Professionnels de la santé

Nom utilisateur

Professionnels Santé Nombre : 11

Rechercher...  Fonction de recherche dans le tableau

Nom prénom	Métier	Adresse	Code Postale	Ville	Mail	Téléphone

Bouton de navigation de l'application

La liste de tous les professionnels de sa région,

Formulaire de création d'un professionnel de la santé

Nom utilisateur

Ajouter Professionnel De Santé

Nom

Prénom

Métier

Adresse

Code postal

Ville

Téléphone

Mail

Envoyer

Bouton de navigation de l'application

Au formulaire d'ajout d'un nouveau professionnel

3.2 PRESENTATION DE L'INTERFACE

3.2.1 MENU GENERAL



Menu des
démarcheurs



Menu des délégués

L'application étant développée pour réaliser les quatre opérations de base pour la persistance des données CRUD (Create Read Update Delete), le menu s'adapte en fonction des droits attribués.

Les démarcheurs ayant l'autorité la plus faible dans l'application, ils ont le droit de visionnage (Read) et peuvent consulter les listes qui représentent les données stockées dans la base de données.

Les délégués ont l'autorité la plus élevée et peuvent créer, lire, mettre à jour ou supprimer des données de la base de données via l'application.

C'est pour cela que le menu des délégués fournit plus d'options afin de répondre à leurs besoins.

3.2.2 LES FORMULAIRES

Formulaire de création d'un visite

Nom utilisateur

Créer Visite

Région

Employé

Professionnel de Santé

Médicament

Date de visite

Créer

Bouton de navigation de l'application

Pour crée une nouvelle visite le délégué devra remplir un formulaire. Afin de fluidifier la tâche, le formulaire récupère l'info liée a la région du délégué. Cela permet de pré-remplir la région, d'affiche les démarcheurs et les professionnels exerçant dans cette région. Le délégué pourra sélectionner le médicament et la date de la visite.

Formulaire de création d'un démarcheur

Nom utilisateur

Ajouter Démarcheur

Nom

Prénom

Téléphone

Mail

Mot de passe

Région

Attribution d'un région

Delegue de Région

Attribution d'un délégué

Créer

Bouton de navigation de l'application

Formulaire de création d'un professionnel de la santé

Nom utilisateur

Ajouter Professionnel De Santé

Nom

Prénom

Métier

Adresse

Code postal

Ville

Téléphone

Mail

Envoyer

Bouton de navigation de l'application

3.3 PRINCIPALES REQUETES DEVELOPPEES

3.3.1 CREER VISITE

Requete de creation d'une visite :

1. La fonction commence par vérifier si l'utilisateur connecté est un responsable de secteur en interrogeant la table `responsable_secteur` dans la base de données.
2. Ensuite, elle récupère toutes les régions à partir de la table `delegue_region`.
3. Si l'utilisateur est un responsable de secteur, elle récupère le nom du secteur du responsable et tous les démarcheurs associés à ce secteur en effectuant une jointure entre les tables `demarcheur` et `employe`.
Si l'utilisateur n'est pas un responsable de secteur, elle récupère les démarcheurs associés à l'utilisateur connecté ainsi que la région associée au délégué à partir des tables `demarcheur`, `employe`, et `delegue_region`.
4. La fonction récupère tous les secteurs, professionnels de santé et médicaments avec les noms de leurs catégories associées à partir des tables `secteur`, `professionnel_de_sante`, `medicament`, et `categorie`.
5. Enfin, la fonction retourne une vue `visites.create` en passant toutes les données récupérées sous forme de variables à cette vue, notamment les régions, la région déléguée (si l'utilisateur n'est pas un responsable de secteur), les professionnels de santé, les médicaments, les démarcheurs, et l'information sur le statut de responsable de secteur.

```

// Vérifier si l'utilisateur actuel est un responsable de secteur
$sis_responsable = DB::table( table: 'responsable_secteur' )
->where( column: 'identifiant_responsable', auth()->user()->identifiant_employe )
->first();

// Récupérer toutes les régions
$regions = DB::table( table: 'delegue_region' )
->get();

// Si l'utilisateur est un responsable de secteur
if ($sis_responsable) {
    // Récupérer tous les démarcheurs associés à ce secteur
    $demarcheurs = DB::table( table: 'demarcheur' )
        ->join( table: 'employe', first: 'demarcheur.identifiant_demarcheur', operator: '=', second: 'employe.identifiant_employe' )
        ->get();
} else {
    // Si l'utilisateur n'est pas un responsable de secteur
    // Récupérer les démarcheurs associés à l'utilisateur actuel
    $demarcheurs = DB::table( table: 'demarcheur' )
        ->where( column: 'demarcheur.identifiant_employe', auth()->user()->identifiant_employe )
        ->join( table: 'employe', first: 'demarcheur.identifiant_demarcheur', operator: '=', second: 'employe.identifiant_employe' )
        ->get();

    // Récupérer la région associée à l'utilisateur actuel
    $delegue_region = DB::table( table: 'delegue_region' )
        ->where( column: 'identifiant_delegue', auth()->user()->identifiant_employe )
        ->first()->nom_region;
}

// Récupérer tous les professionnels de santé
$pro_santes = DB::table( table: 'professionnel_de_sante' )->get();
// Récupérer tous les médicaments avec les noms de leurs catégories associées
$medicaments = DB::table( table: 'medicament' )
    ->join( table: 'categorie', first: 'medicament.identifiant_categorie', operator: '=', second: 'categorie.identifiant_categorie' )
    ->get();

```

3.3.2 LISTE DEMARCHEURS

```

// Récupérer tous les employés
$employees = DB::table( table: 'employe' )
->get();

// Récupérer tous les secteurs
$secteurs = DB::table( table: 'secteur' )
->get();

// Récupérer toutes les régions en joignant avec les secteurs
$regions = DB::table( table: 'region' )
    ->join( table: 'secteur', first: 'region.nom_secteur', operator: '=', second: 'secteur.nom_secteur' )
    ->get();

// Récupérer tous les délégués de région en joignant avec les employés
$delegue_regions = DB::table( table: 'delegue_region' )
    ->join( table: 'employe', first: 'delegue_region.identifiant_employe', operator: '=', second: 'employe.identifiant_employe' )
    ->get();

// Récupérer tous les responsables de secteur en joignant avec les employés
$responsable_secteurs = DB::table( table: 'responsable_secteur' )
    ->join( table: 'employe', first: 'responsable_secteur.identifiant_responsable', operator: '=', second: 'employe.identifiant_employe' )
    ->get();

// Récupérer tous les démarcheurs en joignant avec les employés
$demarcheurs = DB::table( table: 'demarcheur' )
    ->join( table: 'employe', first: 'demarcheur.identifiant_demarcheur', operator: '=', second: 'employe.identifiant_employe' )
    ->get();

// Retourner la vue avec toutes les données nécessaires
return view( view: 'employes_demarcheur', [
    // 'element(s)' est la variable qui se crée sur la vue avec pour valeurs => $variable(s)
    'secteurs' => $secteurs,
    'regions' => $regions,
    'employees' => $employees,
    'responsable_secteurs' => $responsable_secteurs,
    'delegue_regions' => $delegue_regions,
    'demarcheurs' => $demarcheurs,
] );

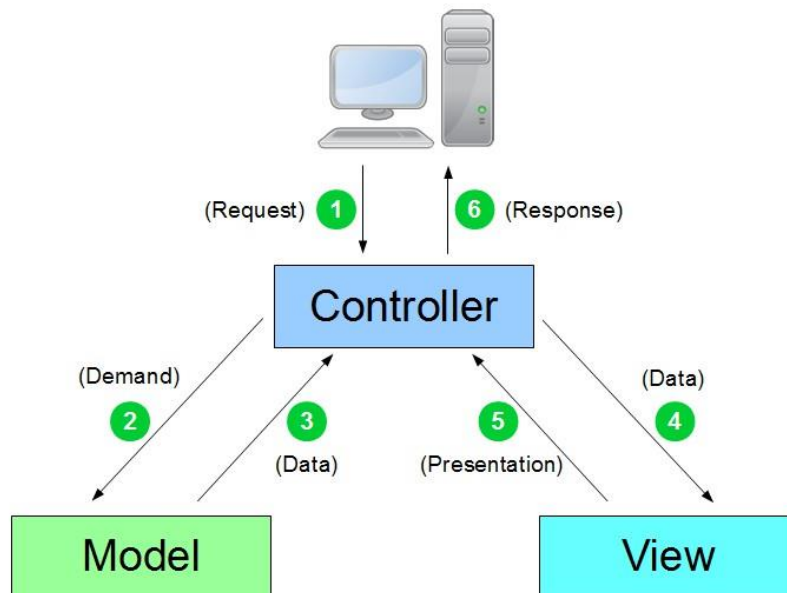
```

Requête pour récupérer toutes les informations de la table « demarcheur » puis les transmetts à la page qui affiche la liste des démarcheurs.

3.4 PRESENTATION DU CODE

Mon application est open source et vous trouverez tout le code sur [mon GitHub](#)

Pour ce projet j'ai utilisé le Framework laravel qui utilise le modèle MVC (Modèle-Vue-Contrôleur).



Une application conforme au motif MVC comporte trois types de modules : les modèles, les vues et les contrôleurs.

Modèle

Élément qui contient les données ainsi que de la logique en rapport avec les données : validation, lecture et enregistrement. Il peut, contenir uniquement une simple valeur, ou une structure de données plus complexe. Le modèle représente l'univers dans lequel s'inscrit l'application.

Par exemple pour une application de banque, le modèle représente des comptes, des clients, ainsi que les opérations telles que dépôt et retraits, et vérifie que les retraits ne dépassent pas la limite de crédit.

Vue

Partie visible d'une interface graphique. La vue se sert du modèle, et peut être un diagramme, un formulaire, des boutons, etc. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle. Dans une application de bureau classique, la vue obtient les données nécessaires à la présentation du modèle en posant des questions. Elle peut également mettre à jour le modèle en envoyant des messages appropriés. Dans une application web une vue contient des balises HTML.

Contrôleur

Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue.

La séparation claire de ces trois composants offre plusieurs avantages, notamment une meilleure maintenabilité, une réutilisation du code, et une facilité de collaboration entre les développeurs travaillant sur différentes parties de l'application. Le modèle MVC est largement adopté dans de nombreux Framework, dont Laravel pour PHP, pour structurer de manière efficace le code des applications web.

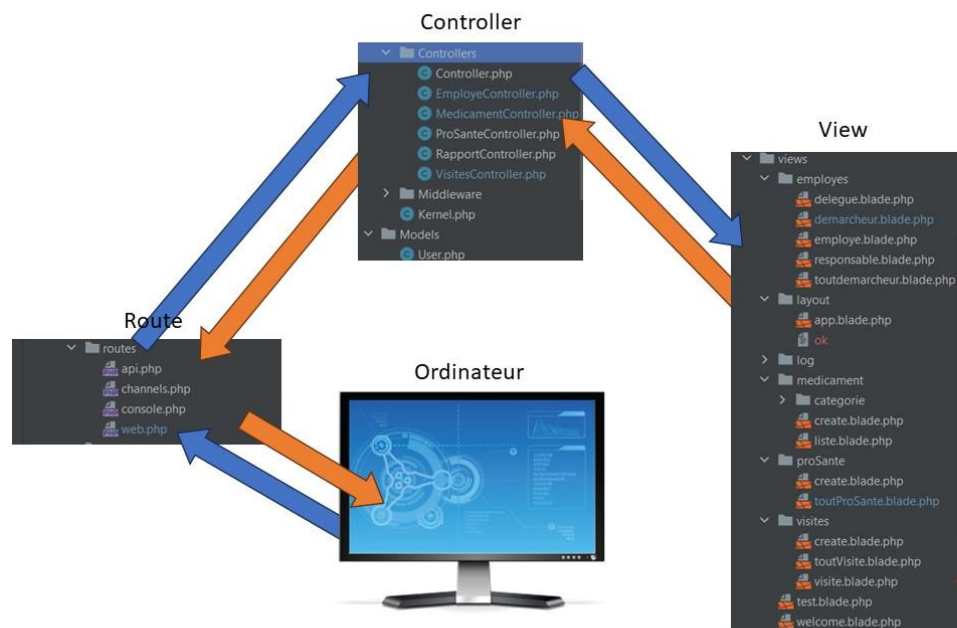


Schéma MVC de LARAVEL