

DESCRIPTION DES MISSIONS REALISEES

Application de gestion des essais cliniques

BTS SIO option SLAM

Bassette Chancereul Gwendoline

Du 10 décembre au 20 décembre 2020

Sommaire

I- CONTEXTE	3
1- INTERETS DES ESSAIS CLINIQUES	3
2- ACTEURS INTERVENANTS DANS UN ESSAI CLINIQUE.....	3
3- ENCADREMENT DES ESSAIS CLINIQUES.....	3
4- DIFFERENTS ESSAIS CLINIQUES	4
5- INDEMNISATION DES VOLONTAIRES AUX ESSAIS CLINIQUES	5
6- PUBLICATION DES RESULTATS AUX ESSAIS CLINIQUES.....	5
II- LE BESOIN	5
1- EXPRESSION DU BESOIN DE GSB	5
2- DESCRIPTION DE L'ENVIRONNEMENT TECHNIQUE.....	5
III- DESCRIPTION DU DOMAINE DE GESTION ET DES FONCTIONNALITES ATTENDUES	5
1- DOMAINE DE GESTION	5
2- SCHEMA PHYSIQUE DE LA BASE DE DONNEES.....	7
3- MODELE CONCEPTUEL DE LA BASE DE DONNEES	8
4- LISTE DES FONCTIONNALITES	9
IV- PRESENTATION DETAILLEE DES MISSIONS REALISEES	10
1- PAGE D'ACCUEIL	10
2- CREATION DE LA BASE DE DONNEES ET DES TABLES A PARTIR DE LA CONSOLE.....	12
3- CREATION DE JEUX D'ESSAIS AVEC FAKER.....	12
4- GESTION DES PRODUITS	13
A- <i>Consultation de la liste des produits</i>	13
B- <i>Consultation des informations d'un produit</i>	15
C- <i>Consultation d'un essai clinique</i>	18
D- <i>Ajout ou modification d'un produit</i>	20
E- <i>Suppression d'un produit de la liste</i>	23
F- <i>Consultation des produits d'une catégorie</i>	24
5- GESTION DES MEDECINS.....	27
A- <i>Consultation de la liste des médecins</i>	27
B- <i>Consultation des informations d'un médecin</i>	28
C- <i>Ajout ou modification des informations d'un médecin</i>	29

I- Contexte

Quand un nouveau médicament est mis au point par un laboratoire pharmaceutique, il est déjà testé sur l'animal (c'est la **recherche préclinique**). Mais cela ne suffit... car avant que les professionnels de santé puissent prescrire le nouveau médicament à leurs patients, il faut s'assurer que le médicament est efficace et bien toléré par des humains.

1- Intérêts des essais cliniques

Les essais cliniques ont un intérêt collectif et individuel :

- un intérêt collectif : ils permettent de progresser pour l'ensemble des malades
- un intérêt individuel : les patients qui sont volontaires aux essais cliniques peuvent accéder à de nouvelle molécule très précocement (bien avant leur commercialisation), ce qui peut être un avantage pour les patients atteints de maladies très graves. De plus, le patient qui participe aux essais cliniques bénéficie ainsi d'examens complémentaires plus réguliers et parfois plus poussés, susceptibles d'améliorer la qualité des soins.

Les études cliniques vont permettre de répondre à plusieurs questions essentielles concernant l'utilisation du nouveau médicament testé :

- des questions de pharmacocinétique sur le devenir du médicament dans le corps humain (absorption, métabolisme, distribution, élimination)
- des questions de pharmacodynamique du médicament sur l'effet du médicament sur l'organisme qui permettront d'établir ou de vérifier les données thérapeutiques (efficacité et effets indésirables).

2- Acteurs intervenants dans un essai clinique

- Le **promoteur** est la personne physique ou morale qui prend l'initiative de l'essai clinique. Dans notre cas le promoteur est **GSB**.
- Le promoteur confie la réalisation d'un essai clinique à un médecin qui dirige et surveille la réalisation de l'essai clinique. Le médecin travaille dans un centre hospitalier et il doit justifier d'une expérience appropriée à l'essai clinique.
- Un essai clinique peut être **uni centre** (réalisé dans un seul centre hospitalier) ou **multi centre** (réalisé dans plusieurs centres). Seuls les essais cliniques uni centre sont gérés par GSB
- Les personnes testant les médicaments lors de l'essai clinique, appelées également "sujets inclus", sont des personnes volontaires pour l'essai clinique : ces personnes peuvent provenir de pays différents, car un même essai clinique peut être réalisé dans plusieurs pays simultanément.

Obligations des investigateurs et du promoteur lors des essais cliniques :

- les investigateurs doivent notifier au promoteur tous les événements indésirables graves et les résultats d'analyse anormaux;
- le promoteur évalue en continu la sécurité des personnes qui se prêtent aux essais cliniques, il peut poursuivre l'essai, demander des modifications du protocole, arrêter l'essai prématurément

3- Encadrement des essais cliniques

Les essais cliniques sont encadrés par la loi et des autorités sanitaires pour assurer la sécurité des patients :

- Aucun essai clinique ne peut commencer sans un avis favorable de **l'organisme de santé** et du **comité d'éthique indépendant** situés dans les pays dans lesquels l'essai clinique est mené. En France, l'organisme de santé est l'**AFSSaPS** -Agence Française de Sécurité Sanitaire des Produits de Santé, et le comité d'éthique est le **CPP** – Comité de Protection des personnes -. Dans les pays dépourvus de comité d'éthique, un comité français composé de parties prenantes du pays concerné fait office de comité d'éthique et c'est à lui qu'incombe la tâche d'examiner le protocole et les procédures de l'essai clinique avant son démarrage.
- Le patient doit signer un formulaire de consentement contresigné par le médecin investigateur qui va l'accompagner pendant toute la durée de l'essai clinique
- L'investigateur (le médecin qui est responsable des essais) doit s'assurer de la sécurité du patient (dans le bénéfice thérapeutique et dans la surveillance des effets secondaires)

4- Différents essais cliniques

Il y a 4 catégories d'essais cliniques :

- Les essais **de phase I** qui évaluent la tolérance d'un nouveau médicament, et donc la tolérance de l'organisme au traitement. A ce stade, les essais sont menés principalement sur un nombre limité de sujets sains, sous strict contrôle médical. **Ces volontaires peuvent être indemnisés.** La molécule est testée sur une courte période. L'objectif est d'évaluer la sécurité d'emploi du produit, son devenir dans l'organisme, son seuil de tolérance, les effets indésirables et de déterminer la dose recommandée. Le traitement évalué est administré à un petit nombre de personnes (de 10 à 40). En oncologie, les essais de phase I sont menés chez des sujets malades.
- Les essais **de phase II** évaluent l'efficacité d'un traitement. Ils nécessitent en général l'inclusion de 40 à 80 patients (personnes malades). Leur objectif est de tester l'efficacité du produit et de déterminer la dose optimale (posologie). Ces études sont le plus souvent comparatives : l'un des 2 groupes de patients reçoit la molécule tandis que l'autre reçoit un placebo.
- Les essais **de phase III** sont des essais comparatifs. Menés sur de larges populations de patients, les essais permettent de comparer l'efficacité thérapeutique du médicament au traitement de référence (lorsque celui-ci existe) ou bien à un placebo (lorsque aucune thérapie n'existe). Ces essais sont très souvent multicentriques (menés dans de nombreux centres d'études). Généralement, ni le patient, ni l'équipe médicale ne savent quel traitement reçoit chacun des malades (essai en double aveugle) : cela permet d'écartier tout préjugé ou jugement faussé de l'une ou l'autre partie sur son efficacité ou ses effets indésirables.
Ces essais nécessitent l'inclusion d'un grand nombre de malades (plusieurs centaines ou milliers de malades) pour établir une différence entre les traitements sur des populations différentes (sexe, âge, pays...).

Si les données et résultats de ces essais sont en faveur du nouveau traitement, cela permet de constituer un dossier d'enregistrement qui sera soumis aux autorités de santé afin qu'elles délivrent l'Autorisation de Mise sur le marché (AMM) qui autorise la commercialisation du nouveau traitement.

- Les essais **de phase IV**. Les essais ne s'achèvent pas avec l'autorisation de mise sur le marché, mais se poursuivent tout au long de sa commercialisation. Des essais dits de Phase IV, sont réalisés dans des conditions proches de la prise en charge habituelle. Ces essais ont pour objectifs de repérer d'éventuels effets indésirables rares non détectés durant les phases précédentes (pharmacovigilance) et de préciser les conditions d'utilisation pour certains groupes de patients à risques. Cette phase permet d'analyser les interactions médicamenteuses et favorise la mise au point de nouvelles formes galéniques (forme physique du médicament tel qu'il sera utilisé chez un patient : comprimés, gélules, sachets, solutions buvables, suspensions injectables...) ainsi que des extensions d'indications thérapeutiques.

5- Indemnisation des volontaires aux essais cliniques

La loi française pose en principe général qu'il n'y a pas de contrepartie financière pour les participants à une recherche biomédicale, hormis le remboursement des frais exposés liés à la recherche (exemple : frais de transport). Toutefois, le promoteur peut verser aux participants à la recherche une indemnité de compensation des contraintes subies (durée, temps d'hospitalisation...).

En France, l'indemnisation maximale autorisée est actuellement de 4 500 euros par an et par personne. Le versement d'une telle indemnité est interdit pour les personnes vulnérables (enfants, personnes sous tutelle...).

La participation à un essai implique l'inscription de la personne sur le fichier national des volontaires se prêtant à une recherche biomédicale. Cette inscription est effectuée par l'investigateur et a comme objectif la sécurité des personnes. Elle permet notamment de s'assurer que la personne respecte bien la période d'exclusion (période pendant laquelle une personne ne peut participer à un autre essai) et que le plafond de l'éventuelle indemnisation est bien respecté.

6- Publication des résultats aux essais cliniques

Les résultats d'un essai clinique peuvent être publiés dans des revues médicales. A compter du 21 juillet 2014, les promoteurs d'essais cliniques sont obligés de saisir leurs résultats d'essais cliniques dans la base de données EudraCT.

II- Le besoin

1- Expression du besoin de GSB

Pour moderniser le suivi des essais cliniques, GSB souhaite disposer d'une application Web, ce qui lui permettra de faire un suivi plus précis des essais cliniques qu'elle gère en tant que promoteur.

Nous étions donc chargés de développer cette application Web au sein d'une équipe de 4 personnes.

2- Description de l'environnement technique

L'application a été développée avec le framework de développement **Symfony 4** qui utilise le langage **PHP**.

Un framework est un ensemble de composants logiciels, de librairies contenant des classes qui permettent de résoudre des problématiques (gérer les requêtes http, gérer des formulaires, abstraction de la base de données, gestion des templates).

L'utilisation d'un framework permet un gain de productivité et que le code soit bien organisé. Avec le framework Symfony, on utilise l'**architecture MVC** (modèle-vue-contrôleur) qui permet de répartir le travail entre le designer et le programmeur. De plus l'architecture et les conventions de codages sont connues.

L'utilisation de Symfony permet d'éviter le traitement de l'HTML, on n'a pas besoin de réaliser de requêtes avec du SQL. Aussi, on utilise des objets et le code est évolutif.

Avec l'**architecture MVC** l'application est composée de modèles, de vues et de contrôleurs. Les modèles permettent l'accès à la base de données et les vues sont les fichiers html permettant l'affichage à l'utilisateur. Les contrôleurs s'occupent du routing, du contrôle des données fournies, de créer des objets en utilisant les classes, et de demander au modèle de créer les enregistrements dans la base de données.

III- Description du domaine de gestion et des fonctionnalités attendues

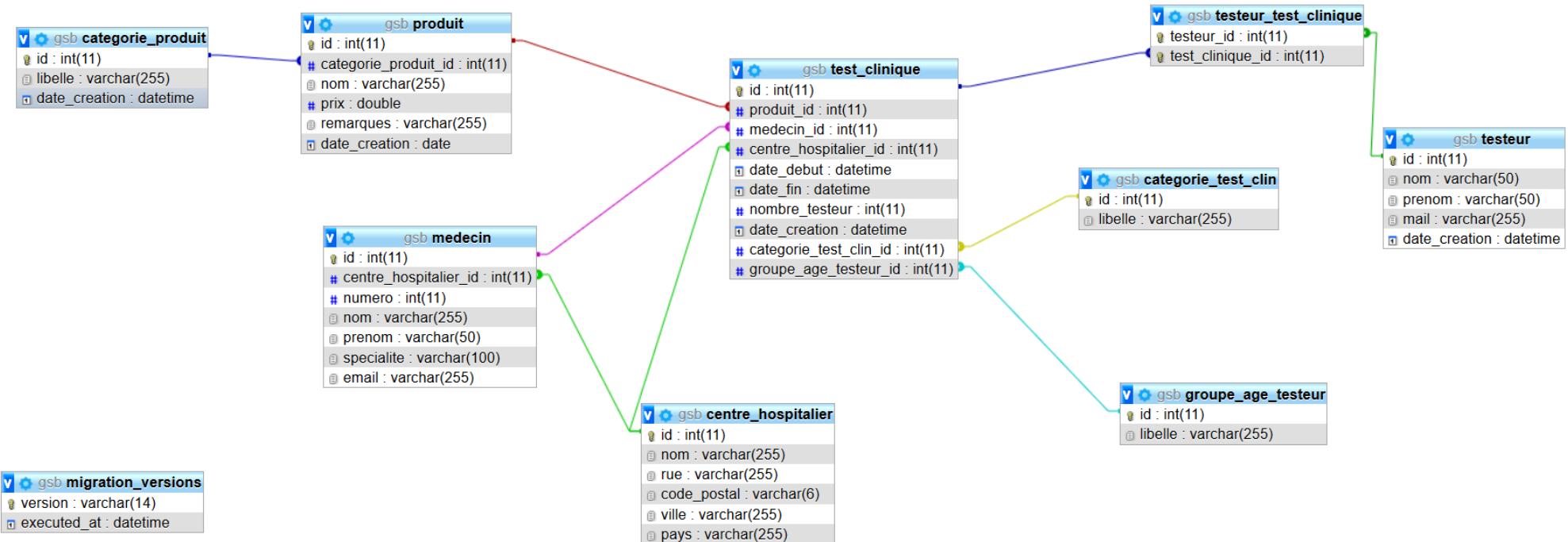
1- Domaine de gestion

L'application devra permettre de saisir les caractéristiques de chaque essai clinique à savoir :

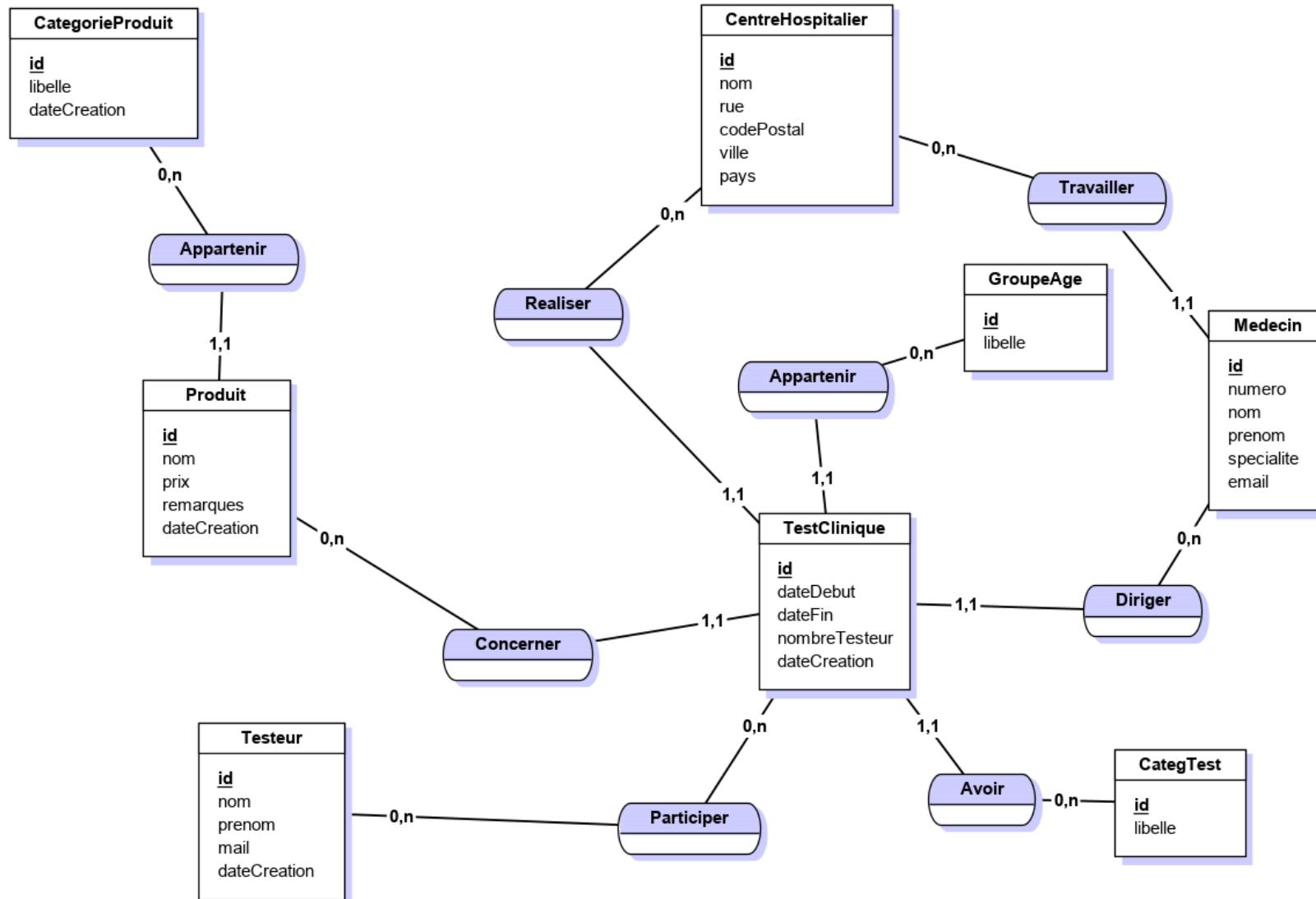
- **le numéro EudraCT** qui est un identifiant d'un essai clinique de médicament se déroulant dans au moins un centre d'investigation dans la Communauté Européenne. Ce numéro est obtenu par le promoteur et il est recensé dans la base de données européenne des essais cliniques de médicaments autorisés (appelée EudraCT)
 - **le numéro ISRCTN** : il s'agit de la Numérotation internationale standardisée des essais cliniques contrôlés randomisés (International Standard Randomised Controlled Trial Number : ISRCTN). Les promoteurs des essais cliniques de médicament peuvent utiliser une numérotation ISRCTN pour identifier leur essai clinique, en plus du numéro EudraCT (par exemple en cas d'essai multinational comprenant des centres d'investigations situés hors de la Communauté européenne).
(En savoir plus sur le numéro ISRCTN <http://www.controlled-trials.com/isrctn>)
- le médicament concerné. L'application doit proposer une fonctionnalité permettant de saisir les caractéristiques du médicament
 - l'objectif de l'essai clinique
 - la période prévue pour l'essai clinique
 - Le groupe d'âge (moins de 10 ans, moins de 18 ans, de 18 à 65 ans, plus de 65 ans) des personnes devant participer à l'essai clinique
 - Le sexe des personnes pouvant participer à l'essai clinique (hommes, femmes, hommes & femmes)
 - le médecin investigateur
 - la catégorie de l'essai clinique (phase I, phase II ...)
 - l'indemnité versée à chaque volontaire s'il y a lieu

La gestion des essais cliniques sera réalisée par les employés du service Recherche et Développement.

2- Schéma physique de la base de données



3- Modèle conceptuel de la base de données



4- Liste des fonctionnalités

L'application devra donc proposer :

- une fonctionnalité permettant de saisir, pour chaque essai clinique, la date d'autorisation de l'essai clinique auprès d'une autorité sanitaire, et ce dès que l'autorisation est reçue.
- Un CRUD permettant de gérer les caractéristiques de chaque autorité sanitaire (sigle, nom, pays)
- un CRUD permettant de gérer les caractéristiques des essais cliniques,
- un CRUD permettant de gérer les caractéristiques de chaque médecin investigator (son numéro (RPPS pour la France), son nom, son prénom, sa spécialité, son email et le centre hospitalier dans lequel il exerce)
- un CRUD permettant de saisir les caractéristiques des différents centres hospitaliers (nom, adresse, pays)
- un CRUD permettant de saisir les caractéristiques des personnes participant à un test
- une fonctionnalité permettant d'indiquer la date de fin de l'essai clinique et son motif (terminé, annulé, suspendu, arrêté pour problème grave) ainsi que la description du résultat de l'essai clinique.
- une fonctionnalité permettant d'enregistrer une publication sur le résultat d'un essai clinique (date de publication, revue médicale concernée).
- Un CRUD permettant de saisir les caractéristiques des revues médicales (nom de la revue, fréquence de parution, nationalité)

Le suivi des essais cliniques est nécessaire via l'application. Différentes consultations devront être proposées. Il est nécessaire de pouvoir consulter :

- pour un médicament, les différents essais cliniques réalisés (catégorie de l'essai, date de début, date de fin, motif de fin de l'essai, le centres hospitalier, le médecin responsable de l'essai et les testeurs enregistrés)
- les différents essais cliniques réalisés (catégorie de l'essai, date de début, date de fin, motif de fin de l'essai, le centre hospitalier, le médecin responsable de l'essai et les testeurs enregistrés) avec les critères de recherche suivants : catégorie de l'essai (plusieurs choix possibles), centre hospitalier
- la liste des centres hospitaliers (nom, adresse) d'un pays choisi par l'utilisateur. Pour chaque centre, on indiquera la liste des essais cliniques (catégorie, date début, date fin) qui y ont été effectués.
- pour chaque essai clinique terminé (catégorie de l'essai, date de début et date de fin de l'essai), la liste des différentes publications (date de publication, nom du support, type de support)
- la liste des centres hospitaliers (nom, adresse) d'un pays choisi par l'utilisateur

Le directeur du service Recherche et Développement aura accès aux mêmes fonctionnalités que ses employés, mais il disposera de deux fonctionnalités supplémentaires lui permettant :

- de saisir le coût financier d'un essai clinique
- de consulter la liste des essais cliniques par médicament (libelle du groupe d'âge, nom du centre hospitalier accueillant l'essai clinique, coût)
- de consulter la liste des essais cliniques par clinique (nom du médicament, libelle du groupe d'âge, coût)

Bien entendu, l'application Web de gestion des essais cliniques à développer doit être sécurisée : les fonctionnalités de saisie des données permettant de gérer les essais cliniques et les consultations décrites précédemment ne seront accessibles qu'après connexion avec identifiant et un mot de passe (la connexion sera accessible à partir de la page d'accueil du site).

A partir de la page d'accueil du site, tout internaute extérieur à GSB pourra consulter :

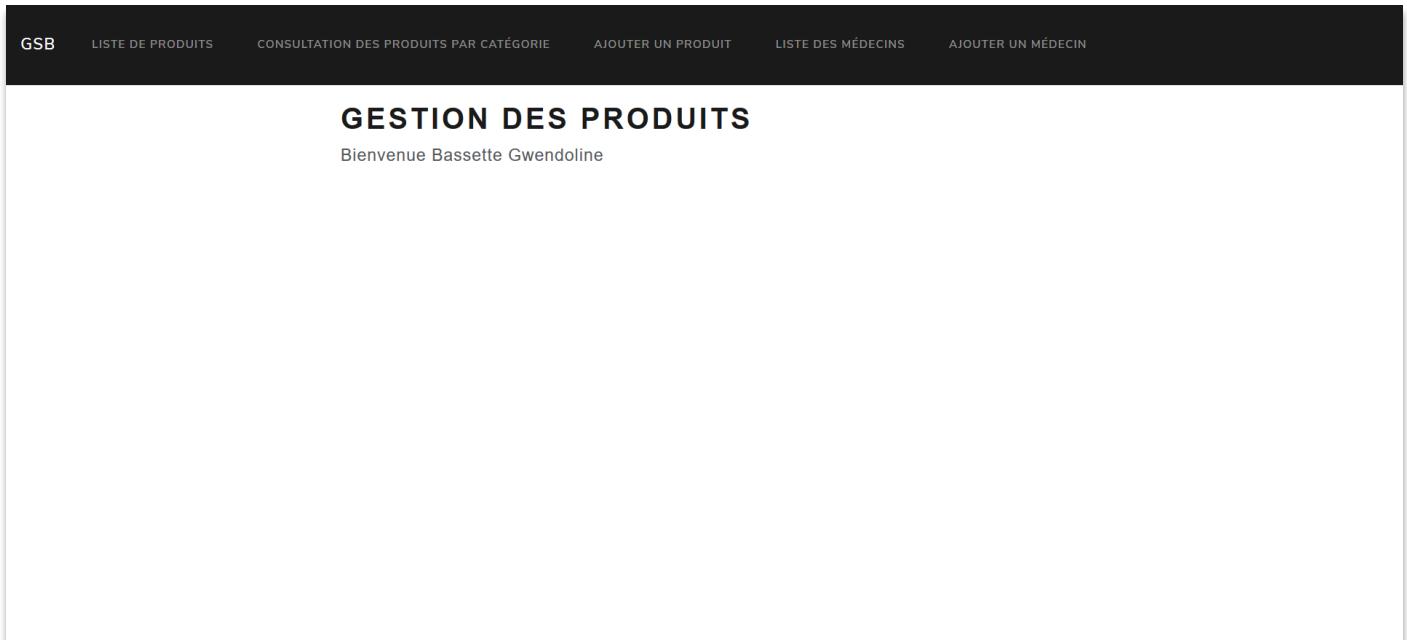
- les essais cliniques à venir proposés par GSB (numéro de l'essai, son objectif, son statut (en cours ou terminé)). Un lien cliquable permettant d'obtenir la description complète de l'essai clinique

- Les caractéristiques des médicaments proposés par GSB. Un lien cliquable permettant d'obtenir la liste des essais cliniques à venir sur le produit.

IV- Présentation détaillée des missions réalisées

1- Page d'accueil

Lorsque l'utilisateur lance l'application il obtient la page suivante :



Il y a sur cette page un message de bienvenu ainsi que la barre de navigation qui sera visible sur toutes les pages de l'application.

Voici le contenu du fichier index.html.twig qui est le template de cette page :

```

{% extends 'base.html.twig' %}

{% block title %}GSB - Gestion des essais cliniques{% endblock %}

{% block body %}
<style>
.example-wrapper { margin: 1em auto; max-width: 800px; width: 95%; font: 18px/1.5 sans-serif; }
.example-wrapper code { background: #F5F5F5; padding: 2px 6px; }

</style>

<div class="example-wrapper">
<h1>Gestion des produits</h1>
<p>Bienvenue {{ nom_prenom }}</p>
</div>
{% endblock %}

```

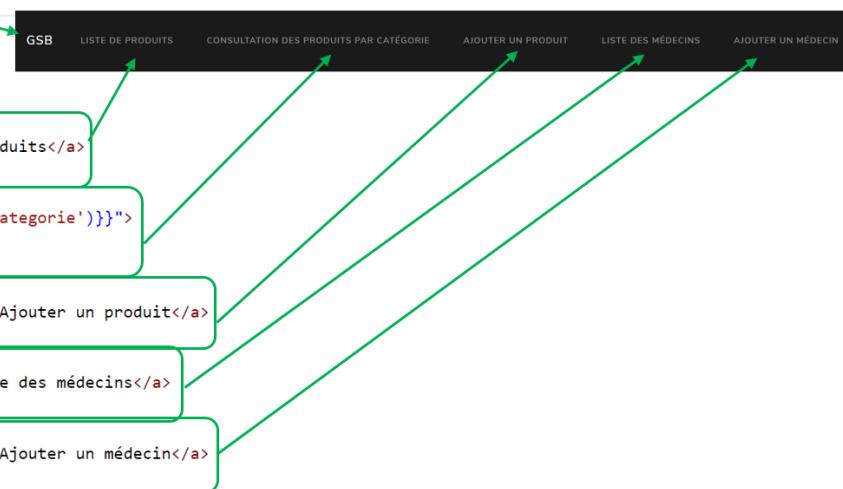
La page hérite de base.html.twig (cela correspond à la barre de navigation)

GSB - Gestion des essais cliniques +

Texte affiché dans l'onglet du navigateur internet

Les instructions de la barre de navigation sont dans le fichier base.html.twig dont voici les instructions :

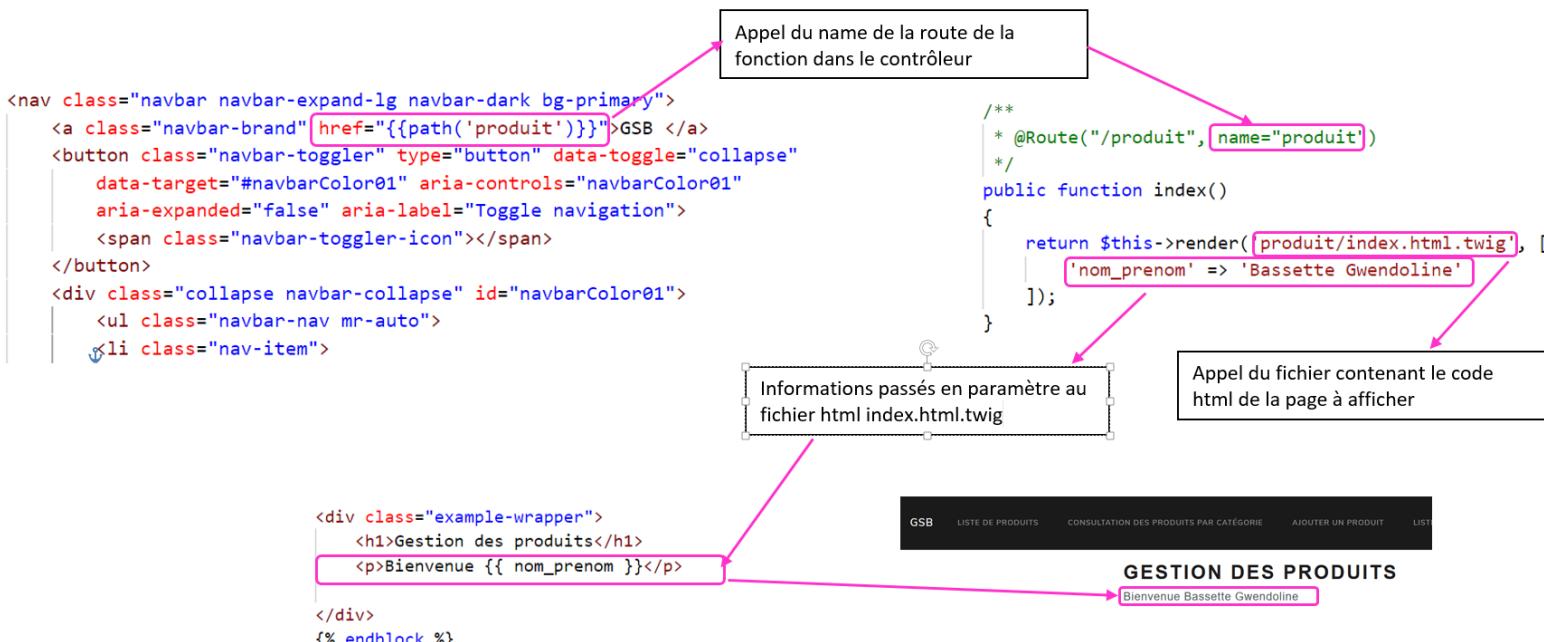
```
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">
<a class="navbar-brand" href="{{path('produit')}}">GSB </a>
<button class="navbar-toggler" type="button" data-toggle="collapse"
        data-target="#navbarColor01" aria-controls="navbarColor01"
        aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarColor01">
    <ul class="navbar-nav mr-auto">
        <li class="nav-item">
            <a class="nav-link" href="{{path('liste')}}">Liste de produits</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{{path('produit_recherche_par_categorie')}}">
                Consultation des produits par catégorie</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{{path('ajoute_un_produit')}}">Ajouter un produit</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{{path('liste_medecin')}}">Liste des médecins</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{{path('ajoute_un_medecin')}}">Ajouter un médecin</a>
        </li>
    </ul>
</div>
</nav>
```



L'application respectant l'architecture MVC (modèle-vue-contrôleur), les pages communiquent les unes avec les autres.

Dans le cas de la barre de navigation, si on clique sur « GSB » qui amène à la page d'accueil, la page base.html.twig appelle la route dont la propriété name est « produit » dans le contrôleur. Cette route correspond à une fonction, ici index, qui elle-même appelle la vue à afficher, dans ce cas il s'agit de la page index.html.twig. Des informations peuvent être fournies à la vue comme une liste par exemples.

Voici comment les pages communiquent les unes avec les autres :



2- Création de la base de données et des tables à partir de la console

L'application nécessite une base de données.

J'ai tout d'abord créé un compte utilisateur qui sera utilisé par l'application pour gérer la base de données en lui donnant le droit de création de base de données, de création, de modification et de suppression de tables.

J'ai ensuite créé la base de données avec la console en utilisant la commande suivante :

php bin/console doctrine:database:create

Puis, j'ai créé mes tables à l'aide de la console en utilisant la commande : **php bin/console make:entity**

Une fois mes tables créées j'ai générer un script de migration en utilisant la commande suivante : **php bin/console make:migration**

Enfin, j'ai appliqué la migration pour que les tables soient créées dans la base de données. J'ai utilisé la commande : **php bin/console doctrine:migration:migrate**

Les tables ont ainsi été créées dans la base de données.

3- Création de jeux d'essais avec Faker

Les fixtures sont un ensemble de données qui permet d'avoir un environnement de développement proche d'un environnement de production avec des fausses données. Cela permet d'avoir des jeux d'essais, de ne pas avoir à les saisir manuellement et d'avoir un certain nombre d'enregistrements.

J'ai utilisé le framework Faker pour réaliser mes fixtures. Ce framework nous permet de créer beaucoup de fausses données lisibles, qui ont un sens et différentes les unes des autres.

Voici un extrait des instructions qui m'ont permis de créer mes fixtures :

```
public function load(ObjectManager $manager)
{
    $faker = Factory::create('fr_FR');

    //Créer 15 centres hospitaliers
    for($i=1; $i<15; $i++){
        $hopital = new CentreHospitalier();
        $hopital->setNom($faker->company());
        ->setRue($faker->streetAddress());
        ->setCodePostal($faker->postcode());
        ->setVille($faker->city());
        ->setPays($faker->country());
        $manager->persist($hopital);

        //Création de 3 à 5 médecin par centre hospitalier
        for($j=1; $j<mt_rand(3, 5); $j++){
            $medecin = new Medecin();
            $medecin->setNumero($faker->randomNumber($nbDigits = NULL, $strict = false));
            ->setNom($faker->lastName());
            ->setPrenom($faker->firstName());
            ->setSpecialite($faker->jobTitle());
            ->setEmail($faker->email());
            ->setCentreHospitalier($hopital);
            $manager->persist($medecin);

            //Création d'une catégorie produit par médecin
            $categProd = new CategorieProduit();
            $categProd->setLibelle($faker->word());
            ->setDateCreation($faker->dateTimeThisDecade());
            $manager->persist($categProd);
        }
    }
}
```

Permet que les données créées soient françaises. Par exemple les villes, les codes postaux et les noms d'entreprise seront français.

Cela générera des noms d'entreprise

Cela générera un nombre aléatoire

Cela créera un métier. C'est pour cela que lors de l'affichage des informations d'un médecin il peut avoir pour spécialité chanteur, peintre ou bien danseur et non une spécialité médicale.

Objet de type hopital créé précédemment

Cela générera un mot aléatoire de type « lorem ipsum »

Cela créera une date aléatoire qui se situe dans la dernière décennie

4- Gestion des produits

GSB propose des médicaments, les utilisateurs doivent donc pouvoir consulter les caractéristiques des produits ainsi qu'ajouter, modifier ou supprimer une fiche produit.

A- Consultation de la liste des produits

L'utilisateur obtient la liste des produits en cliquant sur l'option « Liste des produits » du menu.

The screenshot shows the main dashboard of the GSB application. At the top, there is a navigation bar with links: 'GSB', 'LISTE DES PRODUITS' (which is highlighted with a red oval), 'CONSULTATION DES PRODUITS PAR CATÉGORIE', 'AJOUTER UN PRODUIT', 'LISTE DES MÉDECINS', and 'AJOUTER UN MÉDECIN'. Below the navigation bar, the title 'GESTION DES PRODUITS' is displayed in bold capital letters. Underneath the title, a welcome message 'Bienvenue Bassette Gwendoline' is shown. The background is white with a dark header bar.

L'utilisateur obtient alors la page suivante qui contient un tableau avec l'identifiant et le nom de chaque produit proposé par GSB. L'utilisateur peut ensuite consulter la fiche du produit, modifier ou supprimer la fiche produit.

The screenshot shows the 'LISTE DES PRODUITS' page. The table has two columns: 'IDENTIFIANT DU PRODUIT' and 'NOM DU PRODUIT'. The data rows are:

IDENTIFIANT DU PRODUIT	NOM DU PRODUIT	Action	Action	Action
1	molestiae	Consulter la fiche produit	Modifier la fiche produit	Supprimer la fiche produit
2	sunt	Consulter la fiche produit	Modifier la fiche produit	Supprimer la fiche produit
3	aut	Consulter la fiche produit	Modifier la fiche produit	Supprimer la fiche produit
4	et	Consulter la fiche produit	Modifier la fiche produit	Supprimer la fiche produit
5	repellat	Consulter la fiche produit	Modifier la fiche produit	Supprimer la fiche produit
6	ducimus	Consulter la fiche produit	Modifier la fiche produit	Supprimer la fiche produit

Lorsqu'on clique sur l'option du menu correspondant à cette page on appelle la route ayant pour name « liste », cela correspond à la méthode listeProduit du contrôleur produit :

```
/*
 * @Route("produit/liste", name="liste")
 */
public function listeProduit(ProduitRepository $repo)
{
    $lesProduits = $repo->findAll();
    return $this->render('produit/liste.html.twig', [
        'lesProduits' => $lesProduits
    ]);
}
```

Cette méthode permet de créer une liste avec les informations de tous les produits en faisant appel à la méthode findAll de ProduitRepository. La page html à afficher est ensuite appelée et elle peut utiliser la liste de produits créée précédemment.

Le fichier liste.html.twig permet l'affichage du tableau contenant le nom de chaque produit. Voici son contenu :

```

{% extends 'base.html.twig' %}                                Hérite de base.html.twig qui est
{%- block title %}GSB - Liste des produits{% endblock %}      le fichier contenant la barre de
{%- block body %}                                         navigation



<style>
        .example-wrapper { margin: 1em auto; max-width: 800px; width: 95%; font: 18px/1.5 sans-serif; }
        .example-wrapper code { background: #F5F5F5; padding: 2px 6px; }
    </style>

    <div class="example-wrapper">
        <h1>Liste des produits</h1>
        <table class="table table-striped">
            <thead class="thead-dark">
                <tr>
                    <th>Identifiant du produit</th>
                    <th>Nom du produit</th>
                    <th></th>
                    <th></th>
                    <th></th>
                </tr>
            </thead>
            <tbody>
                {% for produit in lesProduits %}
                    <tr>
                        <td>{{ produit.id }}</td>
                        <td>{{ produit.nom }}</td>
                        <td>
                            <a href="{{ path ('affiche_un_produit', {'id' :produit.id})}}>
                                i class="fas fa-info" title="Voir toutes les caractéristiques"
                            </a>
                        </td>
                        <td>
                            <a href="{{ path ('modifie_un_produit', {'id':produit.id})}}>
                                i class="fas fa-edit" title="Modifier le produit"
                            </a>
                        </td>
                        <td>
                            <a href="{{ path ('supprime_un_produit', {'id':produit.id})}}>
                                i class="fas fa-window-close" title="Supprimer le produit"
                            </a>
                        </td>
                    </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
{% endblock %}


```

Texte affiché dans l'onglet du navigateur

On réalise un foreach sur les produits de la liste lesProduits. On parcourt la liste lesProduits et chaque élément de la liste est placé à tour de rôle dans la variable produit

Affichage de l'identifiant du produit

Affichage d'une icône pour la modification, lorsqu'on passe le curseur une info-bulle s'affiche avec inscrit : « Voir toutes les caractéristiques »
Lorsqu'on clique sur l'icône, la route ayant pour name « affiche_un_produit/id » est appelée, on précise qu'id est l'identifiant du produit sur lequel on clique.

On obtient alors la page suivante :

LISTE DES PRODUITS				
IDENTIFIANT DU PRODUIT	NOM DU PRODUIT	i	edit	x
1	molestiae	i	edit	x
2	sunt	i	edit	x
3	aut	i	edit	x
4	et	i	edit	x
5	repellat	i	edit	x
6	ducimus	i	edit	x

B- Consultation des informations d'un produit

La liste des produits permet simplement d'avoir le nom de chaque produit. Or, il faut permettre à l'utilisateur de consulter toutes les informations concernant le médicament. Pour cela l'utilisateur doit cliquer sur l'icône représentant un « i » à droite du produit dont il souhaite les informations.

IDENTIFIANT DU PRODUIT	NOM DU PRODUIT	i	✎	✖
1	molestiae			
2	sunt			
3	aut			
4	et			
5	repellat			
6	ducimus			

Il obtiendra la page suivante :

CARACTÉRISTIQUES DU PRODUIT N°1	
Libellé : molestiae	
Catégorie du produit : pariatur	
Prix : 110	
Remarques : Praesentium qui voluptas est libero veniam qui.	
Date de création du produit : 07/05/2020	
IL N'Y A PAS DE TESTS POUR CE PRODUIT	

L'utilisateur a ainsi accès pour chaque produit au libellé du produit, à la catégorie du produit, à son prix, aux remarques éventuelles, et à la date de création de la fiche produit. Si des tests, aussi appelés essais cliniques, ont été réalisés sur le produit, alors ils sont listés, sinon on affiche qu'il n'y a aucun test pour ce produit.

Pour avoir cette page, lorsque l'utilisateur clique sur l'icône représentant un « i », la méthode avec la route ayant pour nom « affiche_un_produit » est appelée. On passe également l'identifiant du produit en paramètre pour pouvoir afficher les informations du produit, ci-dessus l'identifiant est 1.

La méthode affiche_un_produit appelle la page html qu'on affiche à l'utilisateur et lui fournit en paramètre un objet de type Produit. Voici ses instructions :

```
/**  
 * @Route("/produit/{id}", name="affiche_un_produit")  
 */  
public function affiche_un_produit (Produit $leProduit = null)  
{  
    return $this->render('produit/unProduit.html.twig',[  
        'leProduit' => $leProduit  
    ]);  
}
```

Voici ce que contient le fichier unProduit.html.twig :

```
{% extends 'base.html.twig' %}

{% block title %}GSB - informations produit{% endblock %}

{% block body %}
<style>
.example-wrapper { margin: 1em auto; max-width: 800px; width: 95%; font: 18px/1.5 sans-serif; }
.example-wrapper code { background: #F5F5F5; padding: 2px 6px; }
</style>

<div class="example-wrapper">
  {% if leProduit is defined %}
    {% if leProduit is not null %}
      <h1>Caractéristiques du produit n°{{leProduit.id}}</h1>
      <p>
        Libellé : {{leProduit.nom}}
        <br> Catégorie du produit : {{leProduit.categorieProduit.libelle}}
        <br>Prix : {{leProduit.prix}}
        <br>Remarques : {{leProduit.remarques}}
        <br>Date de création du produit : {{leProduit.dateCreation | date ('d/m/Y')}}
        {% if leProduit.testClinique is defined %}...
        {% endif %}
      </p>
      {% else %}
        <h1>Produit inexistant</h1>
      {% endif %}
      {% else %}
        <h1>Problème de transmission de données</h1>
      {% endif %}
    {% endif %}
  {% endif %}
</div>
{% endblock %}
```

Si le produit est inexistant la page suivante s'affiche :

The screenshot shows a dark-themed navigation bar with links: GSB, LISTE DES PRODUITS, CONSULTATION DES PRODUITS PAR CATÉGORIE, AJOUTER UN PRODUIT, LISTE DES MÉDECINS, and AJOUTER UN MÉDECIN. Below the bar, the main content area displays the text "PRODUIT INEXISTANT" in large, bold, uppercase letters.

Pour chaque produit on affiche les essais cliniques qui ont été réalisés. Voici la fiche d'un médicament pour lequel des tests ont été réalisés :

The screenshot shows a product detail page for item number 297. The page title is "CARACTÉRISTIQUES DU PRODUIT N°297". Below the title, product details are listed: Libellé : delectus, Catégorie du produit : magnam, Prix : 136, Remarques : Tenetur beatae voluptas iure molestiae voluptatatem quo, Date de création du produit : 12/05/2020. A table then displays two test records:

DÉBUT DU TEST	FIN DU TEST	NOM DE LA CLINIQUE	ADRESSE DE LA CLINIQUE	MÉDECIN RESPONSABLE
23/12/1994	01/12/2009	Collet	78 348 Roger	Poulain
09/11/1981	20/01/2017	Collet	78 348 Roger	Poulain

Pour réaliser cela, j'ai ajouté les instructions suivantes :

```
{% if leProduit.testClinique is defined %}
  {% if leProduit.testClinique is not empty %}
    <table class="table">
      <thead class="thead-dark">
        <tr>
          <th>Début du test</th>
          <th>Fin du test</th>
          <th>Nom de la clinique</th>
          <th>Adresse de la clinique</th>
          <th>Medecin responsable</th>
          <th></th>
        </tr>
      </thead>
      <tbody>
        {% for test in leProduit.testClinique %}
          <tr>
            <td>{{test.dateDebut | date ('d/m/Y')}}</td>
            <td>{{test.dateFin | date ('d/m/Y')}}</td>
            <td>{{test.centreHospitalier.nom}}</td>
            <td>{{test.centreHospitalier.codePostal}} {{test.centreHospitalier.ville}}</td>
            <td>{{test.medecin.nom}}</td>
            <td>
              <a href="{{path ('affiche_un_test', {'id' :test.id})}}">
                <i class="fas fa-info" title="Voir toutes les caractéristiques du test" ></i>
              </a>
            </td>
          </tr>
        {% endfor %}
      </tbody>
    </table>
  {% else %}
    <h4>Il n'y a pas de tests pour ce produit</h4>
  {% endif %}
  {% endif %}
```

Si leProduit.testClinique est défini et n'est pas null, on affiche un tableau dans lequel on retrouve pour chaque test la date de début du test, la date à laquelle s'est terminé le test, le nom et l'adresse du centre hospitalier et le médecin responsable de l'essai. Aussi, on donne à l'utilisateur la possibilité de consulter les informations concernant l'essai clinique en cliquant sur l'icône représentant un « i ».

Si leProduit.testClinique est null, alors on affiche un message à l'utilisateur l'informant qu'aucun test n'a été réalisé pour le produit sélectionné.

C- Consultation d'un essai clinique

En cliquant sur l'icône entourée en rouge ci-dessous, l'utilisateur peut consulter les informations concernant le test clinique.

GSB LISTE DES PRODUITS CONSULTATION DES PRODUITS PAR CATÉGORIE AJOUTER UN PRODUIT LISTE DES MÉDECINS AJOUTER UN MÉDECIN

CARACTÉRISTIQUES DU PRODUIT N°297

Libellé : delectus

Catégorie du produit : magnam

Prix : 136

Remarques : Tenetur beatae voluptas iure molestiae voluptatem quo.

Date de création du produit : 12/05/2020

DÉBUT DU TEST	FIN DU TEST	NOM DE LA CLINIQUE	ADRESSE DE LA CLINIQUE	MÉDECIN RESPONSABLE	
23/12/1994	01/12/2009	Collet	78 348 Roger	Poulain	
09/11/1981	20/01/2017	Collet	78 348 Roger	Poulain	

L'utilisateur obtient alors la page suivante sur laquelle on retrouve le nom du produit testé, la catégorie du produit, les dates de début et de fin du test, le nombre de testeurs prévu, le nom et l'adresse de la clinique responsable, le médecin responsable de l'essai clinique et la date de création de la fiche.

GSB LISTE DES PRODUITS CONSULTATION DES PRODUITS PAR CATÉGORIE AJOUTER UN PRODUIT LISTE DES MÉDECINS AJOUTER UN MÉDECIN

CARACTÉRISTIQUES DU TEST NUMÉRO 652

Produit testé : delectus

Catégorie du produit testé : magnam

Date de début du test : 23/12/1994

Date de fin du test : 01/12/2009

Nombre de testeurs prévus : 152

Clinique : Collet 67, rue Rémy Paris 78 348

Médecin responsable du suivi : Poulain Éric

Date de création du test : 06/02/2016

IDENTIFIANT	NOM
938	Launay Eugène
939	Perrot Laetitia

Pour consulter cette fiche, l'utilisateur clique sur le bouton représentant un « i », la route ayant pour nom « affiche_un_test » est appelée. La fonction correspondant à la route est affiche_un_test qui appelle le fichier de template et qui passe une instance de TestClinique contenant les informations de l'essai clinique.

Voici ces instructions :

```
/**  
 * @Route("/test/clinique/{id}", name="affiche_un_test")  
 */  
public function affiche_un_test (TestClinique $leTest = null)  
{  
    return $this->render('test_clinique/unTest.html.twig',[  
        'leTest' => $leTest  
    ]);  
}
```

Le fichier unTest.html.twig appelé permet l'affichage des informations du test si l'objet leTest est défini et s'il n'est pas null. Si l'objet n'est pas défini, alors on affiche un message à l'utilisateur, s'il est défini mais est null, on informe l'utilisateur que l'essai est inexistant. On affiche aussi un tableau avec l'identifiant, le nom et le prénom de chaque testeur, si les testeurs du test sont définis et non null. S'ils sont null on affiche un message à l'utilisateur pour l'informer qu'il n'y a aucun testeur pour le test sélectionné.

Voici les instructions que contient le fichier de template :

```
<div class="example-wrapper">
    {% if leTest is defined %}
        {% if leTest is not null %}
            <h1>Caractéristiques du test numéro {{leTest.id}}</h1>
            <p>
                <strong>Produit testé :</strong> {{leTest.produit.nom}}
                <br><strong>Catégorie du produit testé :</strong> {{leTest.produit.categorieProduit.libelle}}
                <br><strong>Date de début du test :</strong> {{leTest.dateDebut | date ('d/m/Y')}}
                <br><strong>Date de fin du test :</strong> {{leTest.dateFin | date ('d/m/Y')}}
                <br><strong>Nombre de testeurs prévus :</strong> {{leTest.nombreTesteur}}
                <br><strong>Clinique :</strong> {{leTest.centreHospitalier.nom}} {{leTest.centreHospitalier.rue}}
                {{leTest.centreHospitalier.codePostal}}
                <br><strong>Médecin responsable du suivi :</strong> {{leTest.medecin.nom}} {{leTest.medecin.prenom}}
                <br><strong>Date de création du test :</strong> {{leTest.dateCreation | date ('d/m/Y')}}
```

```
        {% if leTest.testeurs is defined %}
            {% if leTest.testeurs is not empty %}
                <table class="table table-striped">
                    <thead class="thead-dark">
                        <tr>
                            <th>Identifiant</th>
                            <th>Nom</th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for testeur in leTest.testeurs %}
                            <tr>
                                <td>{{testeur.id }}</td>
                                <td>{{testeur.nom }} {{testeur.prenom }}</td>
                            </tr>
                        {% endfor %}
                    </tbody>
                </table>
            {% else %}
                <h4>Il n'y a pas de testeurs pour ce test clinique</h4>
            {% endif %}
        {% endif %}
        </p>
    {% else %}
        <h1>Test inexistant</h1>
    {% endif %}
    {% else %}
        <h1>Problème de transmission de données</h1>
    {% endif %}
</div>
```

Si le test est null, l'utilisateur obtient la page suivante :



Si le test existe mais qu'il n'y a pas de testeurs, l'utilisateur obtient la page suivante :

The screenshot shows a header with links: GSB, LISTE DES PRODUITS, CONSULTATION DES PRODUITS PAR CATÉGORIE, AJOUTER UN PRODUIT, LISTE DES MÉDECINS, and AJOUTER UN MÉDECIN. Below the header is a section titled "CARACTÉRISTIQUES DU TEST NUMÉRO 647". It contains the following information:
Produit testé : reiciendis
Catégorie du produit testé : et
Date de début du test : 16/06/2003
Date de fin du test : 03/11/2014
Nombre de testeurs prévus : 127
Clinique : Collet 67, rue Rémy Paris 78 348
Médecin responsable du suivi : Pelletier Henriette
Date de création du test : 17/02/2017
A message at the bottom states: "IL N'Y A PAS DE TESTEURS POUR CE TEST CLINIQUE".

Si le test existe et qu'il y a des testeurs, l'utilisateur obtient la page suivante :

The screenshot shows a header with links: GSB, LISTE DES PRODUITS, CONSULTATION DES PRODUITS PAR CATÉGORIE, AJOUTER UN PRODUIT, LISTE DES MÉDECINS, and AJOUTER UN MÉDECIN. Below the header is a section titled "CARACTÉRISTIQUES DU TEST NUMÉRO 652". It contains the following information:
Produit testé : delectus
Catégorie du produit testé : magnam
Date de début du test : 23/12/1994
Date de fin du test : 01/12/2009
Nombre de testeurs prévus : 152
Clinique : Collet 67, rue Rémy Paris 78 348
Médecin responsable du suivi : Poulain Éric
Date de création du test : 06/02/2016
Below this is a table with two rows of data:

IDENTIFIANT	NOM
938	Launay Eugène
939	Perrot Laetitia

D- Ajout ou modification d'un produit

Il est possible d'ajouter ou de modifier un produit.

Pour ajouter un médicament, l'utilisateur doit cliquer sur l'option « Ajouter un produit » du menu :

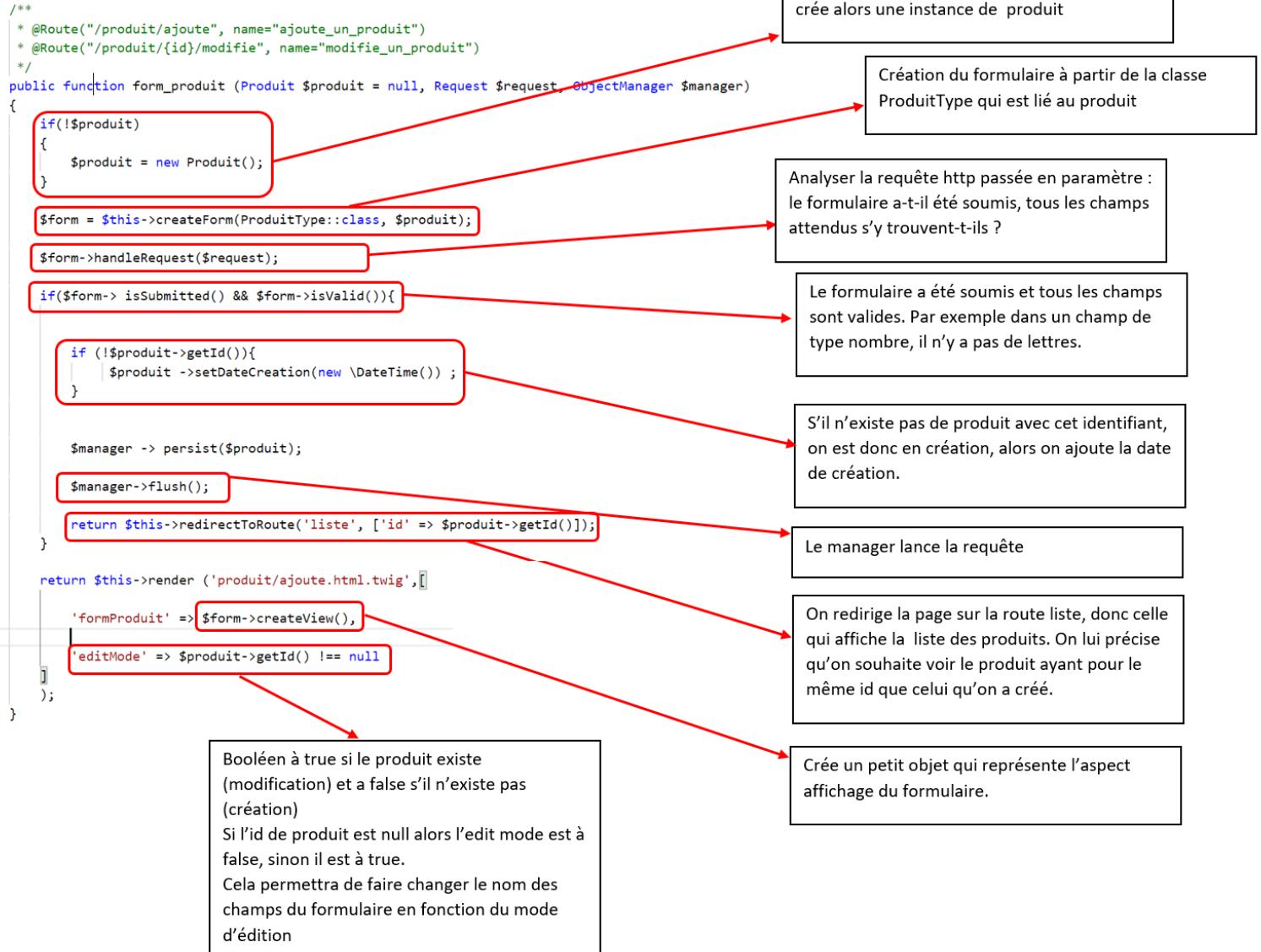
The screenshot shows a header with links: GSB, LISTE DES PRODUITS, CONSULTATION DES PRODUITS PAR CATÉGORIE, a button "AJOUTER UN PRODUIT" (which is highlighted with a red rectangle), LISTE DES MÉDECINS, and AJOUTER UN MÉDECIN. Below the header is a section titled "GESTION DES PRODUITS". It displays a welcome message: "Bienvenue Bassette Gwendoline".

Pour modifier un produit, l'utilisateur doit cliquer sur le bouton encadré en rouge dans la liste des produits :

LISTE DES PRODUITS

IDENTIFIANT DU PRODUIT	NOM DU PRODUIT	i	✎	x
292	laborum	i	✎	x
293	reiciendis	i	✎	x

Ce n'est pas la même route qui est appelée lorsqu'on ajoute ou modifie un produit mais les deux routes correspondent à la même fonction. La route ayant pour name « ajoute_un_produit » est appelée lorsqu'on clique sur « ajouter un produit » dans le menu, lorsqu'on clique sur l'icône de modification c'est la route dont le name est « modifie_un_produit » qui est appelée. Ces deux routes correspondent à la fonction form_produit dont voici les instructions :



Le fichier de template ajoute.html.twig contient les instructions suivantes :

```
{% extends 'base.html.twig' %}  
{% form_theme formProduit 'bootstrap_4_layout.html.twig' %}  
{% block body %}  
<br>  
    {% if editMode %}  
        <h1>Modification des caractéristiques d'un produit</h1>  
    {% else %}  
        <h1>Ajout des caractéristiques d'un produit</h1>  
    {% endif %}  
  
    {{ form_start(formProduit, {attr : {'novalidate':'novalidate'}}) }}  
  
    {{ form_row(formProduit.nom, {'attr' : {'placeholder' : "Dénomination du produit"}}) }}  
    {{ form_row(formProduit.prix ,{'attr' : {'placeholder' : "Prix du produit"}, 'label' : "Prix de vente"}) }}  
    {{ form_row(formProduit.remarques, {'attr' : {'placeholder' : "Remarques obligatoires"}}) }}  
    {{ form_row(formProduit.categorieProduit, {'label' : "Catégorie du produit"}) }}  
  
    {{ form_widget (formProduit) }}  
  
    <button type="submit" class="btn-primary btn">  
        {% if editMode %}  
            Enregistrer les modifications  
        {% else %}  
            Ajouter le produit  
        {% endif %}  
    </button>  
  
    {{ form_end (formProduit)}}  
{% endblock %}
```

Si editmode est à true, le titre de la page est « modification des caractéristiques d'un produit ». Sinon, on affiche « Ajout des caractéristiques d'un produit ».

Balise de début du formulaire, formProduit est le nom du formulaire à afficher.

Champs du formulaire. Le placeholder est ce qui s'affichera dans le composant du formulaire.

Si editmode est à true alors le bouton permettant de soumettre le formulaire aura pour texte « Enregistrer les modifications », sinon il aura pour texte « Ajouter le produit »

Balise de fin du formulaire

Voici le formulaire d'ajout :

The screenshot shows a web page titled 'AJOUT DES CARACTÉRISTIQUES D'UN PRODUIT'. The page has a dark header with navigation links: 'GSB', 'LISTE DE PRODUITS', 'CONSULTATION DES PRODUITS PAR CATÉGORIE', 'AJOUTER UN PRODUIT', 'LISTE DES MÉDECINS', and 'AJOUTER UN MÉDECIN'. The main content area contains four input fields: 'Nom' (placeholder: 'Dénomination du produit'), 'Prix de vente' (placeholder: 'Prix du produit'), 'Remarques' (placeholder: 'Remarques obligatoire'), and 'Catégorie du produit' (placeholder: 'pariatur'). Below these fields is a black button labeled 'AJOUTER LE PRODUIT'.

Voici le formulaire de modification d'un produit :

GSB LISTE DES PRODUITS CONSULTATION DES PRODUITS PAR CATÉGORIE AJOUTER UN PRODUIT LISTE DES MÉDECINS AJOUTER UN MÉDECIN

MODIFICATION DES CARACTÉRISTIQUES D'UN PRODUIT

Nom
laborum

Prix de vente
60

Remarques
Quos sunt aut quo id sequi expedita ut.

Catégorie du produit
et

ENREGISTRER LES MODIFICATIONS

E- Suppression d'un produit de la liste

Il est possible de supprimer un produit de la liste en cliquant sur l'icône encadrée en rouge ci-dessous :

IDENTIFIANT DU PRODUIT	NOM DU PRODUIT	i	✎	x
292	laborum	i	✎	x
293	reiciendis	i	✎	x
294	veniam	i	✎	x
295	ut	i	✎	x

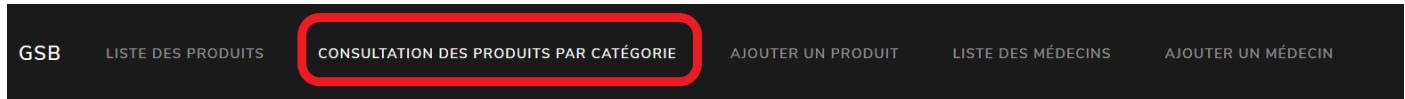
Lorsque l'utilisateur clique sur le bouton de suppression la route ayant pour name « `supprime_un_produit` » est appelée. Cette route correspond à la méthode « `supprime_un_produit` ». Voici son contenu :

```
/**  
 * @Route("/produit/supprime/{id}", name="supprime_un_produit")  
 */  
public function supprime_un_produit(Produit $produit = null, objectManager $manager, ProduitRepository $repo)  
{  
    $manager->remove($produit);  
    $manager->flush();  
    return $this->redirectToRoute('liste');  
}
```

Cette méthode supprime le produit passé en paramètre, effectue la requête sur la base de données et redirige l'utilisateur sur la route correspondant à la liste de produits.

F- Consultation des produits d'une catégorie

L'utilisateur a la possibilité d'afficher tous les produits d'une catégorie. Pour cela il doit cliquer sur l'option « consultation des produits par catégorie » du menu de navigation :



Il obtient alors la page suivante :

The screenshot shows a search interface titled 'PRODUITS D'UNE CATÉGORIE'. It includes a dropdown menu labeled 'Choisissez une catégorie :' containing the text 'pariatur'. Below the dropdown is a black button labeled 'LANCER LA RECHERCHE'.

L'utilisateur sélectionne la catégorie produit dans la liste déroulante puis clique sur le bouton « Lancer la recherche ». Il obtient ensuite le même tableau que pour la liste des produits dans lequel on retrouve l'identifiant du produit, son nom et trois boutons permettant la consultation du produit, sa modification et sa suppression.

Voici la page qu'obtient l'utilisateur après avoir sélectionné une catégorie :

The screenshot shows the search results for the category 'aut'. The results are displayed in a table:

IDENTIFIANT	NOM DU PRODUIT	i	✎	x
1	304	hic		
2	305	fugit		
3	306	cupiditate		

Lorsque l'utilisateur clique sur l'option dans le menu de navigation, la route ayant pour nom «produit_recherche_par_categorie» est appelée. Cela correspond à la fonction produit_recherche_par_categorie dont voici les instructions :

```

/**
 * @Route("/produit/categorie", name="produit_recherche_par_categorie")
 */
public function produit_recherche_par_categorie(ProduitRepository $repository, Request $request)
{
    //création de l'objet contenant les critères de recherche
    $recProdCateg = new RecProdCateg();

    //Création du formulaire
    $form = $this->createForm(RecProdCategType::class, $recProdCateg);

    //on demande au formulaire de gérer la requête HTTP
    $form ->handleRequest($request);

    //le formulaire a t-il été soumis et est-il valide ?
    if($form->isSubmitted() && $form->isValid()){
        //le formulaire a été soumis et est valide
        //on demande au repository de Produit les produits ayant la catégorie
        //sélectionnée par l'utilisateur
        $lesProduits=$repository->findByCategorieProduit($recProdCateg->categories);
    }else{
        //cas de la demande du formulaire suite au clic sur l'option du menu
        // "Consultation des produits par catégorie"
        // dans ce cas la collection lesProduits n'est pas renseignée
        $lesProduits = null;
    }

    //on appelle la vue : on lui transfert le formulaire et la collection des produits
    return $this->render('produit/rec_produit_categ.html.twig', [
        'lesProds' => $lesProduits,
        'formRec' => $form->createView()
    ]);
}

```

Voici les instructions de la vue rec_produit_categ.html.twig permettant l'affichage du formulaire pour le choix d'une catégorie de produit :

```

<div class="example-wrapper">
    <h1>Produits d'une catégorie</h1>

    {{ form_start(formRec) }}

    {{ form_row(formRec.categories, {'label': "Choisissez une catégorie :"} ) }}

    <button type="submit" class="btn btn-primary">Lancer la recherche</button>

    {{ form_end(formRec) }}

```

The diagram illustrates the structure of the rec_produit_categ.html.twig template with annotations:

- Balise de début du formulaire**: Points to the opening tag {{ form_start(formRec) }}.
- Liste déroulante avec les catégories de produits. Le label est le texte qui s'affiche au-dessus de la liste déroulante**: Points to the row tag {{ form_row(formRec.categories, {'label': "Choisissez une catégorie :"}) }}.
- Bouton de validation**: Points to the submit button <button type="submit" class="btn btn-primary">Lancer la recherche</button>.
- Balise de fermeture du formulaire**: Points to the closing tag {{ form_end(formRec) }}.

Lorsque l'utilisateur a choisi une catégorie de produit et validé, la liste des produits s'affiche grâce aux instructions suivantes :

```

{%
    if lesProds is defined %
        {% if lesProds is not null %}
            <div class="exemple-wrapper">
                <table class="table table-striped">
                    <thead class="thead-dark">
                        <tr>
                            <th scope="col"></th>
                            <th scope="col">Identifiant</th>
                            <th scope="col">Nom du produit</th>
                            <th scope="col"></th>
                            <th scope="col"></th>
                            <th scope="col"></th>
                        </tr>
                    </thead>
                    <tbody>
                        {% set ind = 0 %}
                        {% for prod in lesProds %}
                            {% set ind = ind + 1 %}
                            <tr>
                                <th scope="row">{{ind}}</th>
                                <td>{{ prod.id }}</td>
                                <td>{{ prod.nom }}</td>
                                <td>
                                    <a href="{{path ('affiche_un_produit', {'id' :prod.id})}}>
                                        <i class="fas fa-info" title="Voir toutes les caractéristiques" ></i></a>
                                    </td>
                                <td>
                                    <a href="{{path ('modifie_un_produit', {'id':prod.id})}}>
                                        <i class="fas fa-edit" title="Modifier le produit"></i></a>
                                    </td>
                                <td>
                                    <a href="{{path ('supprime_un_produit', {'id':prod.id})}}>
                                        <i class="fas fa-window-close" title="Supprimer le produit"></i>
                                    </a>
                                </td>
                            </tr>
                        {% endfor %}
                    </tbody>
                </table>
            </div>
        {% endif %}
    {% else %}
        <h1>Erreur grave de transfert de données</h1>
    {% endif %}
</div>

```

Initialisation de la variable ind qui permettra l'affichage du numéro de la ligne

On réalise un foreach sur les produits de la liste lesProds. On parcourt la liste lesProds et chaque élément de la liste est placé à tour de rôle dans la variable prod

On incrémente la variable ind à chaque fois qu'on passe dans la boucle foreach

Affichage du numéro de la ligne

Affichage des informations et des boutons de consultation, modification et de suppression

Si la liste de produits n'est pas définie on affiche un message d'erreur.

L'utilisateur obtient alors les pages suivantes :

PRODUITS D'UNE CATÉGORIE

Choisissez une catégorie :

pariatur

LANCER LA RECHERCHE

Choix d'une catégorie et validation

	IDENTIFIANT	NOM DU PRODUIT			
1	304	hic	<i>i</i>	<input checked="" type="checkbox"/>	<i>x</i>
2	305	fugit	<i>i</i>	<input checked="" type="checkbox"/>	<i>x</i>
3	306	cupiditate	<i>i</i>	<input checked="" type="checkbox"/>	<i>x</i>

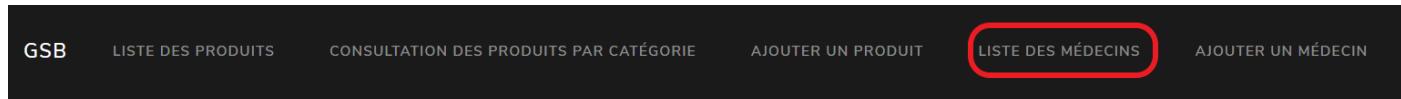
Page 26 sur 31

5- Gestion des médecins

Les essais cliniques sont suivis par des médecins. L'application permet la consultation de la liste des médecins, de la fiche d'un médecin et l'ajout ou la modification d'une fiche.

A- Consultation de la liste des médecins

L'utilisateur obtient la liste des médecins en cliquant sur l'option « liste des médecins » du menu de navigation :



La fonction `listeMedecin` correspondant à la route ayant pour name « `liste_medecin` » est alors appelée. Cette méthode réalise une liste contenant tous les médecins puis appelle la vue `liste.html.twig` et lui fournit la liste de médecins.

Voici ses instructions :

```
/*
 * @Route("medecin/liste", name="liste_medecin")
 */
public function listeMedecin(MedecinRepository $repo)
{
    $lesMedecins = $repo->findAll();
    return $this->render('medecin/liste.html.twig',[  

        'lesMedecins' => $lesMedecins  

    ]);
}
```

La vue `list.html.twig` permet l'affichage d'un tableau contenant le numéro du médecin, son nom, son prénom et sa spécialité. Voici les instructions permettant de réaliser cela :

```
<div class="example-wrapper">
<h1>Liste des médecins</h1>
<table class="table table-striped">
    <thead class="thead-dark">
        <tr>
            <th>Numéro du médecin</th>
            <th>Nom du médecin</th>
            <th>Prénom du médecin</th>
            <th>Spécialité</th>
            <th></th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        {% for medecin in lesMedecins %}
            <tr>
                <td>{{ medecin.numero }}</td>
                <td>{{ medecin.nom }}</td>
                <td>{{ medecin.prenom }}</td>
                <td>{{ medecin.specialite }}</td>
                <td>
                    <a href="{{ path ('affiche_un_medecin', {'id' :medecin.id})}}>  

                        <i class="fas fa-info" title="Voir toutes les informations" ></i></a></span>
                    </td>
                <td>
                    <a href="{{ path ('modifie_un_medecin', {'id':medecin.id})}}>  

                        <i class="fas fa-edit" title="Modifier les informations du médecin"></i></a>
                    </td>
                </tr>
            {% endfor %}
    </tbody>
</table>
{% endblock %}
```

On réalise un foreach sur les médecins de la liste lesMedecins.
On parcourt la liste lesMedecins et chaque élément de la liste est placé à tour de rôle dans la variable medecin.

Affichage des informations du médecin

Icône cliquable permettant d'être redirigé sur la route ayant pour name « `affiche_un_medecin` » en lui passant l'identifiant du médecin.
Le title permet l'affichage d'une info-bulle lorsqu'on survole l'icône.

Voici la page qu'obtient l'utilisateur :

LISTE DES MÉDECINS				
NUMÉRO DU MÉDECIN	NOM DU MÉDECIN	PRÉNOM DU MÉDECIN	SPÉCIALITÉ	
57506	Gay	Gabrielle	Danseur	 
112091	Lefebvre	Martin	Choréologue	 
9	Lacroix	Pierre	Radio chargeur	 
5736	Rousset	Louis	Sommelier	 
6752	Petitjean	Patrick	Appareilleur-gazier	 

B- Consultation des informations d'un médecin

L'utilisateur peut consulter la fiche du médecin en cliquant sur l'icône représentant un « i » :

LISTE DES MÉDECINS				
NUMÉRO DU MÉDECIN	NOM DU MÉDECIN	PRÉNOM DU MÉDECIN	SPÉCIALITÉ	
4357	Pelletier	Henriette	Chanteur	 
62709493	Poulain	Éric	Etainier	 
1168	Wagner	Eugène	Plaquiste enduiseur	 

Lorsque l'utilisateur clique sur l'icône, la fonction `affiche_un_medecin` dont la route a pour name « `affiche_un_medecin` » est appelée. Cette méthode appelle la vue et lui fournit un objet contenant les informations du médecin.

Voici ses instructions :

```
/**  
 * @Route("/medecin/{id}", name="affiche_un_medecin")  
 */  
public function affiche_un_medecin (Medecin $leMedecin = null)  
{  
    return $this->render('medecin/unMedecin.html.twig',[  
        'leMedecin' => $leMedecin  
    ]);  
}
```

La vue unMedecin.html.twig permet l'affichage de la fiche du médecin. Les informations affichées sont le nom et le prénom du médecin, son email, sa spécialité ainsi que le centre hospitalier auquel il est rattaché. Si l'objet leMedecin fournit par le contrôleur n'est pas défini, on affiche un message d'erreur, s'il est null on informe l'utilisateur que la fiche médecin n'existe pas

Voici les instructions permettant cet affichage :

```
<div class="example-wrapper">
    {% if leMedecin is defined %}
        {% if leMedecin is not null %}
            <h1>Informations du médecin numéro {{leMedecin.numero}}</h1>
            <p>
                <strong>Nom :</strong> {{leMedecin.nom}}
                <br> <strong>Prénom :</strong> {{leMedecin.prenom}}
                <br><strong>Email :</strong> {{leMedecin.email}}
                <br><strong>Spécialité :</strong> {{leMedecin.specialite}}
                <br><strong> Centre hospitalier :</strong> {{leMedecin.centreHospitalier.nom}}
            </p>
        {% else %}
            <h1>Medecin inexistant</h1>
        {% endif %}
        {% else %}
            <h1>Problème de transmission de données</h1>
        {% endif %}
    {% endif %}

</div>
{% endblock %}
```

Voici la page qu'obtient l'utilisateur :

The screenshot shows a web application interface. At the top, there is a navigation bar with links: GSB, LISTE DES PRODUITS, CONSULTATION DES PRODUITS PAR CATÉGORIE, AJOUTER UN PRODUIT, LISTE DES MÉDECINS, and AJOUTER UN MÉDECIN. Below the navigation bar, the main content area has a title "INFORMATIONS DU MÉDECIN NUMÉRO 4357". Underneath the title, there is a list of five items, each preceded by a bold label: "Nom : Pelletier", "Prénom : Henriette", "Email : henri12@sfr.fr", "Spécialité : Chanteur", and "Centre hospitalier : Collet".

C- Ajout ou modification des informations d'un médecin

Il est possible d'ajouter ou de modifier une fiche médecin.

Pour ajouter une fiche, l'utilisateur doit cliquer sur l'option « Ajouter un médecin » du menu :

The screenshot shows a web application interface with a navigation bar identical to the previous one. The "AJOUTER UN MÉDECIN" button is highlighted with a red oval.

Pour modifier une fiche, l'utilisateur doit cliquer sur l'icône de modification dans la liste des médecins :

The screenshot shows a "LISTE DES MÉDECINS" table. The table has columns: NUMÉRO DU MÉDECIN, NOM DU MÉDECIN, PRENOM DU MÉDECIN, and SPÉCIALITÉ. A row in the table contains the values: 4357, Pelletier, Henriette, Chanteur. To the right of the last column, there is an edit icon (pencil) and a delete icon (trash can), both of which are highlighted with red ovals.

En cas de modification c'est la route « modifie_un_medecin » qui est appelée en passant l'identifiant du médecin, en cas d'ajout la route ayant pour name « ajoute_un_medecin » est appelée. La fonction correspondant à ces deux routes est form_medecin, voici ses instructions :

```

public function form_medecin (Medecin $medecin = null, Request $request, ObjectManager $manager)
{
    //Si medecin est null alors on est en création, on crée donc une instance de médecin
    if(!$medecin)
    {
        $medecin = new Medecin();
    }
    //Création du formulaire à partir de la classe MedecinType qui est lié au medecin
    $form = $this->createForm(MedecinType::class, $medecin);

    //on demande au formulaire de gérer la requête HTTP
    $form->handleRequest($request);

    ////le formulaire a t-il été soumis et est-il valide ?
    if($form-> isSubmitted() && $form->isValid()){
        $manager -> persist($medecin);
        //Le manager lance la requête
        $manager->flush();

        //On redirige la page sur la route liste, donc celle qui affiche la liste des médecins.
        // On lui précise qu'on souhaite voir le médecin ayant pour le même id que celui qu'on a créé.
        return $this->redirectToRoute('liste_medecin', ['id' => $medecin->getId()]);
    }

    return $this->render ('medecin/ajouteMedecin.html.twig',[

        'formMedecin' => $form->createView(),
        'editMode' => $medecin->getId() != null
    ]);
}
}

```

La vue ajouteMedeicn.html.twig permet l'affichage du formulaire et des champs. Voici ses instructions :

The diagram illustrates the structure of the `ajouteMedecin.html.twig` template with various annotations:

- Header Logic:** A red box highlights the conditional logic at the top: `{% if editMode %}`, `<h1>Modification des informations d'un médecin</h1>`; `{% else %}`, `<h1>Ajout des informations d'un médecin</h1>`; `{% endif %}`. An arrow points from this block to a callout box stating: "Affichage du titre de la page en fonction du mode d'édition. Si editMode à pour valeur true alors on est en modification, sinon on est en création".
- Form Start:** A red box highlights the `form_start` block: `{{ form_start(formMedecin, {attr : {'novalidate':'novalidate'}}) }}`. An arrow points from this block to a callout box stating: "Balise de début du formulaire".
- Form Rows:** A red box highlights the `form_row` blocks for various fields: `numero`, `nom`, `prenom`, `specialite`, `email`, and `centreHospitalier`. An arrow points from this block to a callout box stating: "Eléments du formulaire. Le placeholder est le texte qui sera affiché dans les éléments du formulaire. Le label est le texte qui sera situé au-dessus du composant".
- Submit Button:** A red box highlights the `button` element: `<button type="submit" class="btn-primary btn">`. Inside the button, there is conditional logic: `{% if editMode %}`, `Enregistrer les modifications`; `{% else %}`, `Ajouter le médecin`; `{% endif %}`. An arrow points from this block to a callout box stating: "Bouton permettant de soumettre le formulaire. Le texte du bouton change en fonction de la valeur d'editMode".
- Form End:** A red box highlights the `form_end` block: `<{{ form_end (formMedecin)}}>`.

Voici le formulaire que l'utilisateur obtient en cas d'ajout :

GSB LISTE DE PRODUITS CONSULTATION DES PRODUITS PAR CATÉGORIE AJOUTER UN PRODUIT LISTE DES MÉDECINS AJOUTER UN MÉDECIN

AJOUT DES INFORMATIONS D'UN MÉDECIN

Numéro

Nom

Prenom

Spécialité

Email

Centre hospitalier où exerce le médecin

AJOUTER LE MÉDECIN

Voici ce qu'obtient l'utilisateur en cas de modification d'une fiche :

GSB LISTE DE PRODUITS CONSULTATION DES PRODUITS PAR CATÉGORIE AJOUTER UN PRODUIT LISTE DES MÉDECINS AJOUTER UN MÉDECIN

MODIFICATION DES INFORMATIONS D'UN MÉDECIN

Numéro

Nom

Prenom

Spécialité

Email

Centre hospitalier où exerce le médecin

ENREGISTRER LES MODIFICATIONS