**DOSSIER PROJET ECF**

SOMMAIRE :

1. Description du projet
2. Description de ma tache
3. Méthodologie
4. Retour d’expériences
5. Axe d’amélioration

1/ Description du projet

Le projet a pour but une gestion de stock d’une famille avec conception d’une liste de course si des produits viennent à manquer et des propositions de recette en fonction des produits que vous avez en stock.

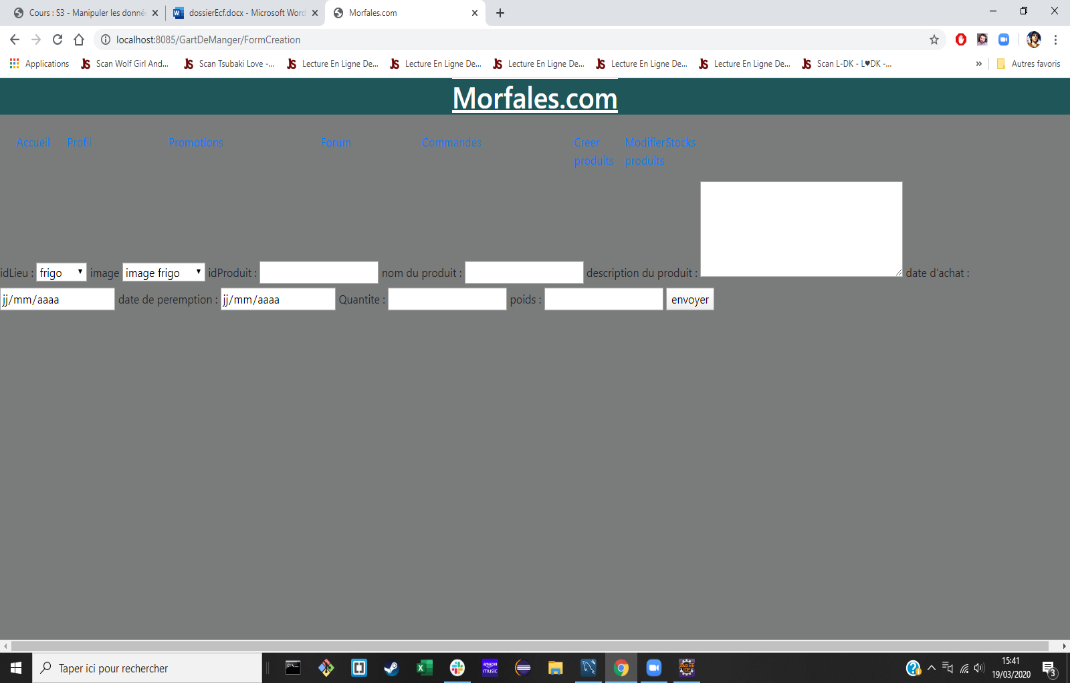
2/ Description de ma tache

Ma 1ère partie de tâche est de gérer le stock et de réaliser un affichage des produits total disponible, d’afficher une alerte si le produit est bientôt périmé ou s’il il n’y a presque plus de stock et de réaliser aussi un affichage des produits en fonction de l’endroit où ils sont ranger cad placard et frigo.

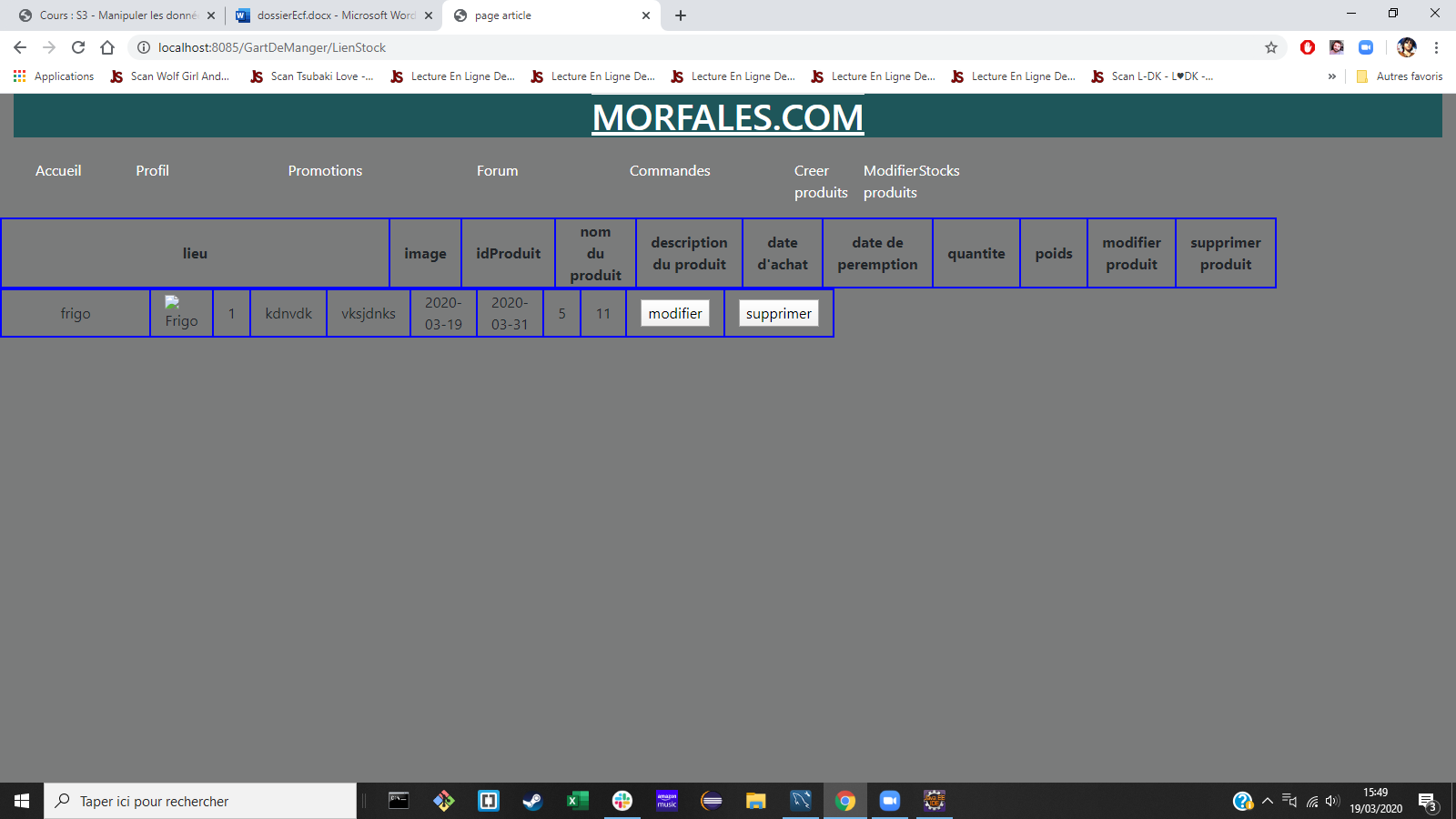
Ma 2nd partie de tâche est de gérer les modifications ou la suppression d’un produit.

3/Méthodologie

Pour réaliser la 1ere partie j’ai créé un formulaire qui contiendra des champs pour **le lieu où est stocker le produit, le lien vers l’image qui définit le lieu, un id de produit, un nom de produit, une description de produit, une date d’achat, une date de péremption, une quantité et un poids.**



Une fois valider ce formulaire vous enverra dans une servlet qui traitera les données vérifiera s'il y a des erreurs et après les stockera dans une arraylist de produit pour ensuite les afficher sur une page jsp



Les inscriptions et affichage sont gérer dynamiquement dès que l’utilisateur rentre les infos dans le formulaire.

Sur la barre de navigation vous avez la fonction supprimer qui vous amené sur un formulaire qui liste vos produits dans un menu déroulant ensuite vous choisissez quel produit à supprimer.

En appuyant sur le bouton de validation du formulaire de suppression cela vous emmène vers votre servlet qui aura le traitement pour la suppression du produit ma page suppression est faite totalement en CSS

Code ci-dessous :

Une image contenant capture d’écran, assis, moniteur

Description générée automatiquement

Suite à la 1ere version nous avons décidé avec mon groupe de repartir à zéro et de refaire tout le code.

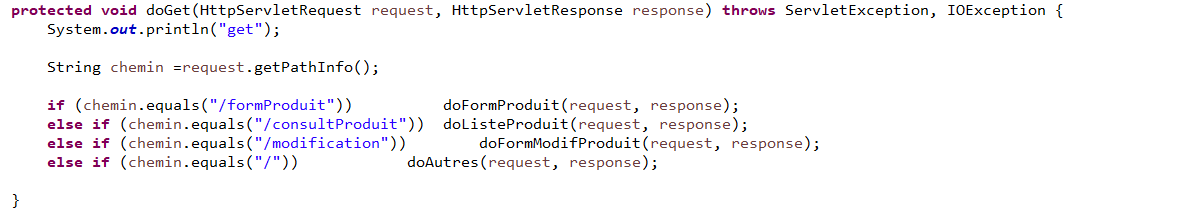
Nous sommes parties d’un Template que nous avons découper et adapter à nos besoins

J’ai créé un filtre qui me permettra de diriger vers mon contrôleur qui aura dans la méthode doGet des conditions qui enverront vers des méthodes qui permettront la redirection vers les bonnes pages

**Filtre :**

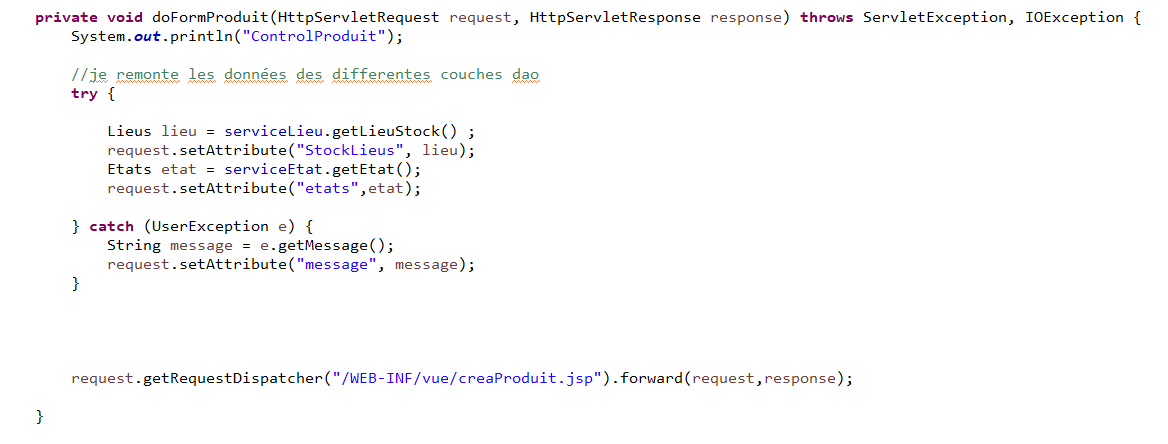


**Controleur :** méthode doGet



Dans la méthode doGet j’ais créé une chaine de caractère(String chemin) qui contiendra que le chemin qui passe par la servlet.

Les conditions permettent de dire que si le chemin est égale a ce qu’il y a de contenue dans les parenthèse alors il redirige vers la bonne page

Par exemple :

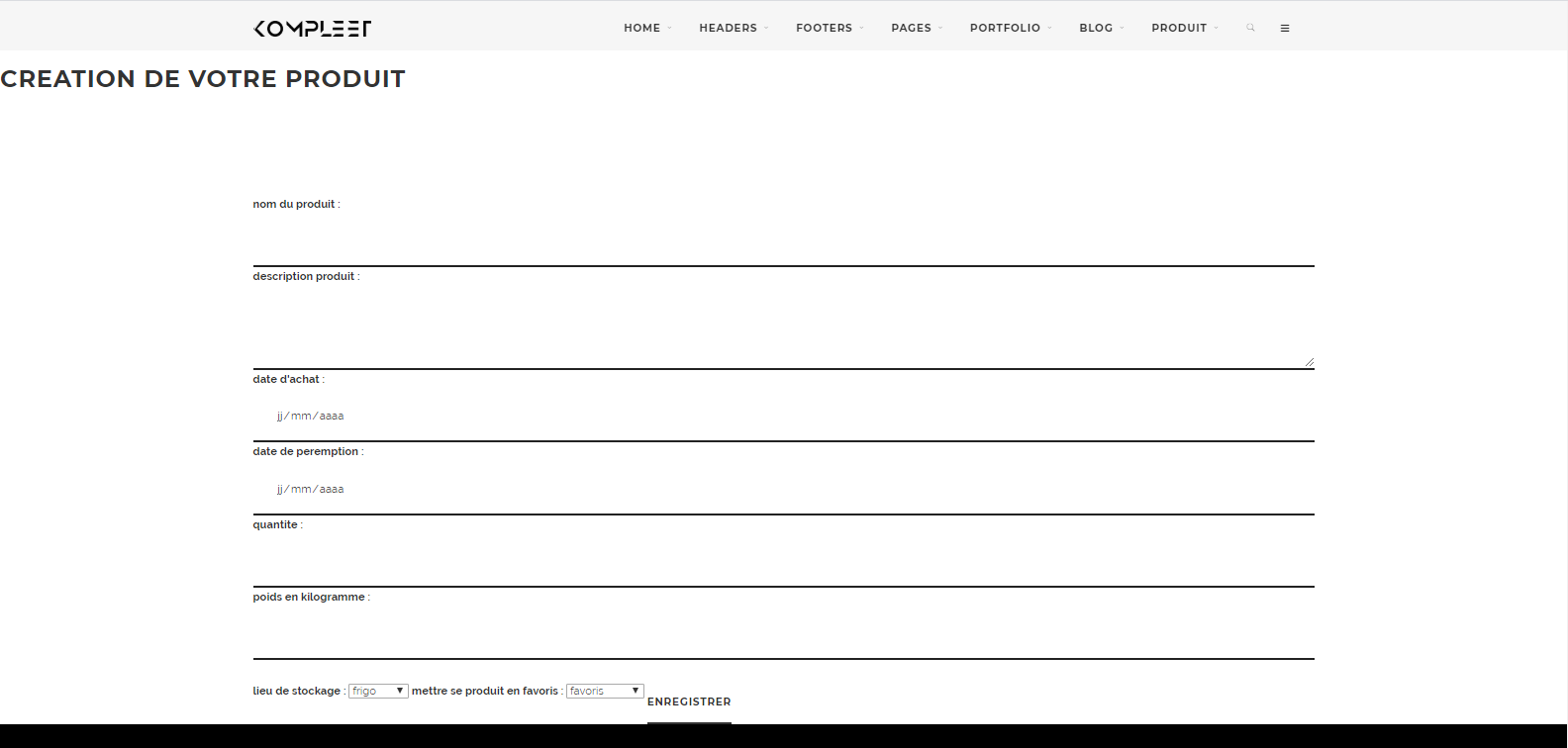
Cet méthode doFormProduit va chercher les données d’état et de lieu contenues dans ma couche dao pour pouvoir ensuite les afficher dans ma page creaProduit.jsp et à la fin je redirige vers ma page

Pour ma partie j’ai créé 4 JSP

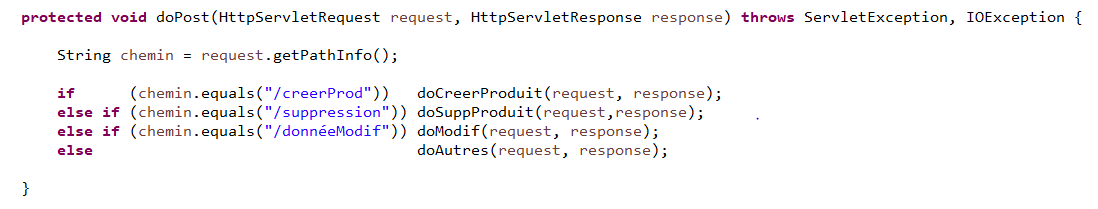
* L’accueil



* Un formulaire de création de produit pour enregistrer un produit avec son **nom**, sa **description**, sa **date d’achat**, sa **date de péremption**, sa **quantité**, son **poids**, son **lieu de stock** et son **état** (favoris , non favoris).

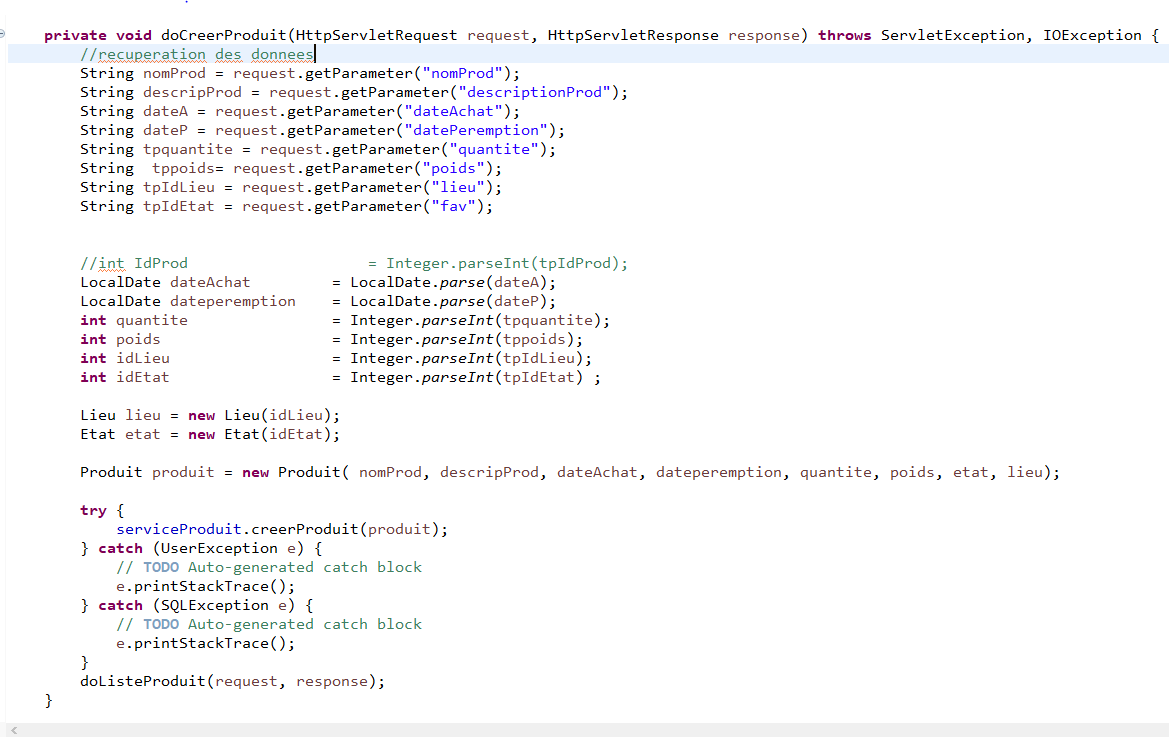


* Code création produit
* **Controleur :** méthode doPost



* Dans la méthode doPost je fais pareil que dans la méthode doGet mais cet fois les méthodes que j’appelle dans les conditions seront celle qui traiteront les données pour la création , la suppression et la modification d’un produit

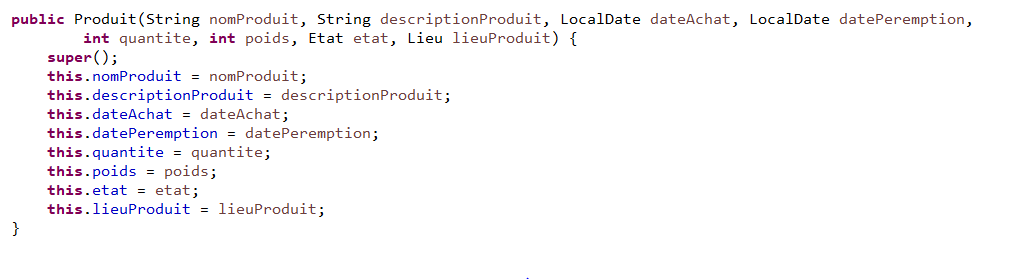
Pour la création d’un produit je passe par la méthode doCreerProduit(request, response)



Dans cette méthode je récupère tous les paramètres envoyer par le formulaire avec les request.getParameter(nom de l’attribut name ) je parse pour mettre les bons types de données.

J’instancie un lieu et un état auquel j’attribut les valeurs de leur id respective et qui a un constructeur qui ne prend en paramètre que l’id.

Pour le produit c’est différent son constructeur ne prend pas en compte son id car il est généré automatiquement par la base de données (auto-incrémente sur l’id du produit)

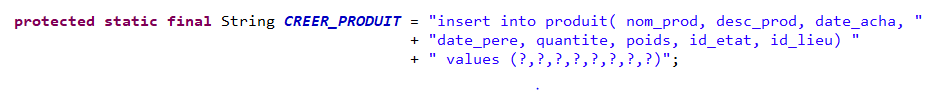


Les instances de lieu et d’état sont appeler dans le constructeur du produit.

Ensuite ma méthode appelle ma couche service puis ma couche dao pour insérer les données



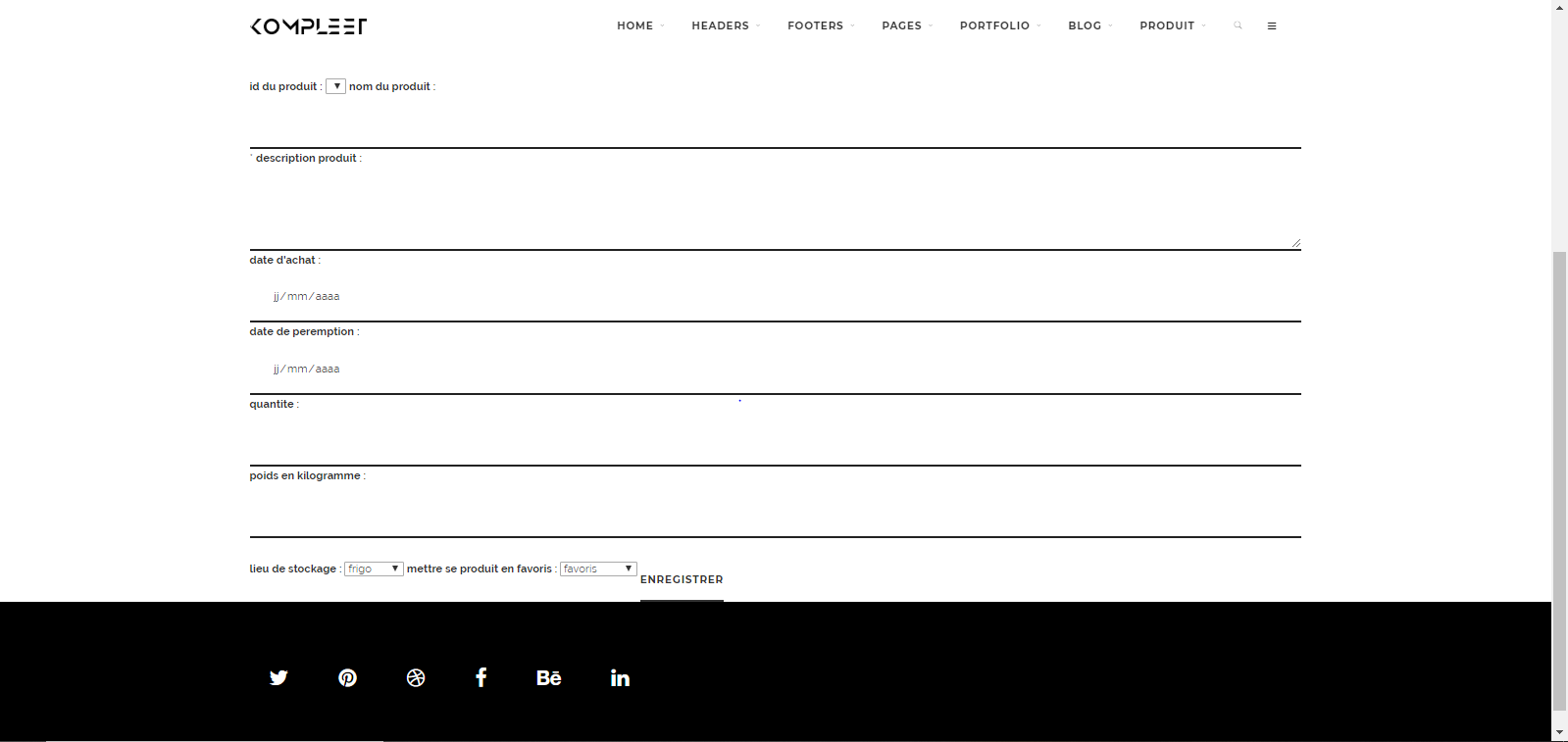
Je fais un preparedStatement après avoir ouvert la connexion vers ma base de données

j

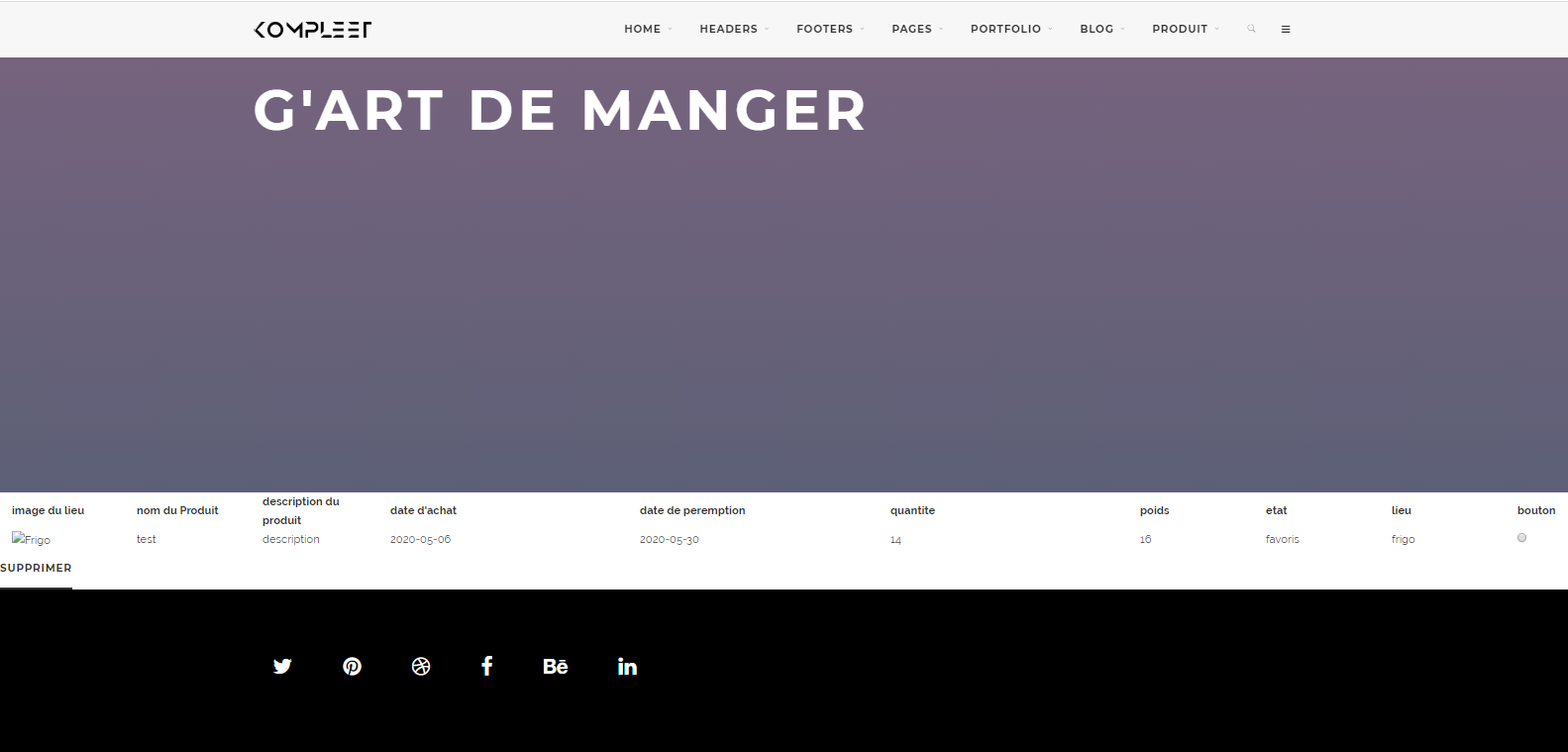
Et je set mes données qui vont aller remplacer les points d’interrogation de ma requête ci-dessus et je les numérote en fonction de l’ordre dont je les appelle dans ma requête

* Un formulaire de modification de produit pour modifier un produit qui gère **l’identifiant sous forme de select**  qui liste tous les produits du stock et les autres champs sont pareil que ceux de la page création

Je pouvais gérer la modification d’un produit sur la jsp qui affiche le stock mais j’ais décider de ne pas surcharger la page avec un formulaire et de le faire dans un fichier a part



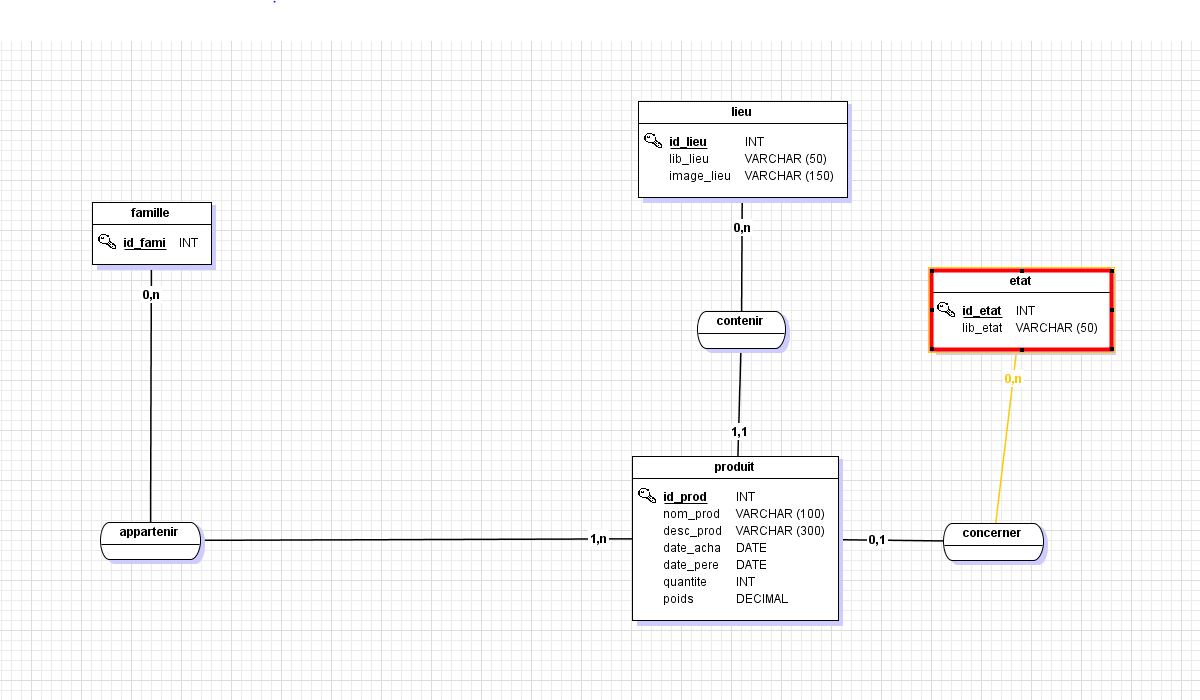
* Et une page qui liste tous les produits contenus dans le stock



Dictionnaire de données



Modèle Conceptuel des Données



Modèle Logique des Données

famille(id\_fami)

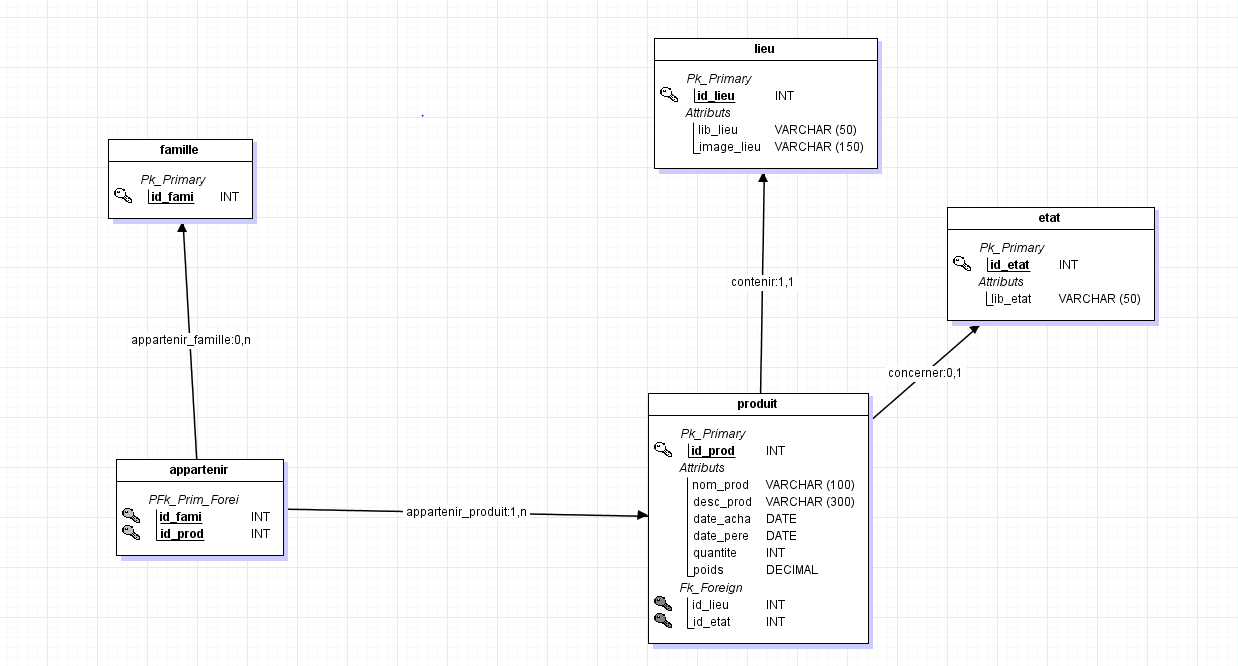
etat( id\_etat, lib\_etat)

produit(id\_prod, nom\_prod, desc\_prod, date\_acha, date\_pere, quantite, poids, id\_lieu\*, id\_etat\*)

lieu(id\_lieu, lib\_lieu, image\_lieu)

appartenir (id\_fami\*, id\_prod\*)

la table appartenir récupère en foreign key l’id de la table famille et l’id de la table produit



DDL

Une image contenant capture d’écran

Description générée automatiquement

Une image contenant capture d’écran

Description générée automatiquement

Une image contenant capture d’écran

Description générée automatiquement

Le **on delete cascade** de la table appartenir permet que quand l’on supprime un produit il n’y est pas de problème de foreign key et supprime aussi l’id produit et l’id famille qui correspondent dans la table appartenir

**JEU D’essai**

**-- lieu**

**insert into lieu values (1,'frigo','image');**

**insert into lieu values(2,'placard','image2');**

**select \* from lieu;**

**-- etat**

**insert into etat values(1);**

**insert into etat values(2);**

**select \* from etat;**

**-- famille**

**insert into famille values();**

**insert into famille values();**

**insert into famille values();**

**insert into famille values();**

**insert into famille values();**

**insert into famille values();**

**insert into famille values();**

**select \* from famille;**

**-- produit**

**insert into produit values (null,"pate","pesto","21-02-20","21-02-21",12,259,1,2);**

**insert into produit values (null,"riz","pesto","21-03-20","21-02-21",12,259,1,2);**

**insert into produit values (null,"poisson","pesto","21-04-20","21-02-21",12,259,1,2);**

**insert into produit values (null,"tomate","pesto","21-05-20","21-02-21",12,259,1,2);**

**insert into produit values (null,"banane","pesto","21-06-20","21-02-21",12,259,2,1);**

**insert into produit values (null,"comcombre","pesto","21-07-20","21-02-21",12,259,2,1);**

**insert into produit values (null,"steak","pesto","21-08-20","21-02-21",12,259,2,2);**

**insert into produit values (null,"patate","pesto","21-09-20","21-02-21",12,259,2,2);**

**select \* from produit;**

**-- appartenir**

**insert into appartenir values (3,2);**

**insert into appartenir values (6,4);**

**insert into appartenir values (2,5);**

**select \* from appartenir;**

**-- devait renvoyer une erreur**

**insert into lieu values(3,null,'');**

**insert into lieu values(3,'',null);**

22/04/20

Mise en place de l’architecture du site et du code

29/04/20

Affichage des produits dans la jsp et commencer la suppression des produits

04/05/20

Mise en place de la suppression d’un produit et de la modification d’un produit

06/05/20

Rédaction du dossier

4/Retour d’expérience

Suite a la première version du site je trouve que j’ai réussi a monter mon site avec une architecture cohérente .

5/Axe d’amélioration