

IDENTIFICATION DES PROBLÈMES MAJEURS

Fonctions asynchrones sans gestion d'erreur

Dans user.js côté serveur, getUser() utilise un fetch sans try catch. De même pour getAllUsers() dans liste_user.js

Duplication de code

Beaucoup de code se retrouve doublé comme getFavori ou ajouterFavori dans les fichiers accueil.js, produits.js et favori.js.

De la duplication de code est aussi présente dans le fichier changemdp.js

Fonctions inutiles

Certaines fonctions sont définies mais inutiles dans function.js

Manque d'efficacité

Certains morceaux de code manquent d'efficacité par exemple la succession de if else dans le fichier changemdp.js

Vérification null

Dans le fichier register.js, la vérification est null.

Erreur de redirection

Dans le fichier compte.js, la redirection suite au fetch est mauvaise et ne fait rien.

Redondance

L'appel de btn() dans le fichier favoris.js est redondant il pourrait uniquement être appelé dans afficherfavoris()

Pas de gestion des erreurs

Globalement, dans l'intégralité des fichiers il y a un manque cruel de gestion des erreurs, le code est simplement balayé et exécuté et l'identification d'erreur est donc bien plus complexe.

Mot de passe non sécurisés

Les mots de passe sont envoyés en clair au serveur depuis login.js

Problèmes fonctions asynchrones

Globalement, dans l'intégralité des fichiers, les fonctions asynchrones ralentissent sévèrement l'application, sans gestion d'erreur ou de time out.

Sécurité des inputs

Globalement, l'intégralité des inputs ne sont pas sécurisées et permettent à l'utilisateur d'envoyer des données impossibles (nombres négatifs...)

Manque d'optimisation

Globalement le code manque grandement d'optimisation, par exemple dans ajout_article, la fonction recupererInformationForm() est appelée deux fois.

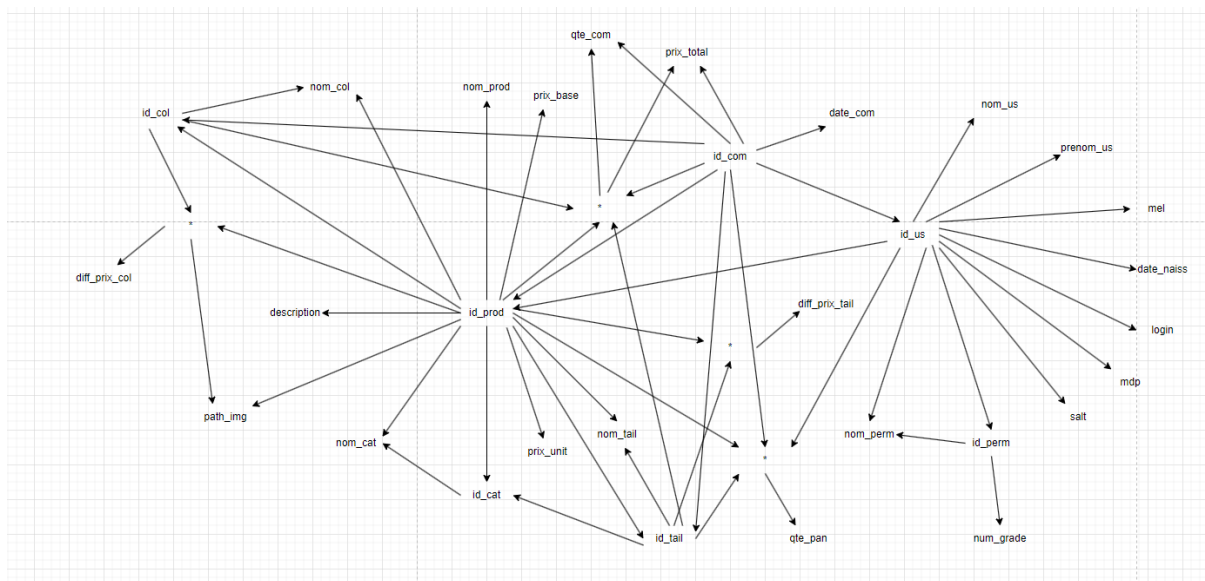
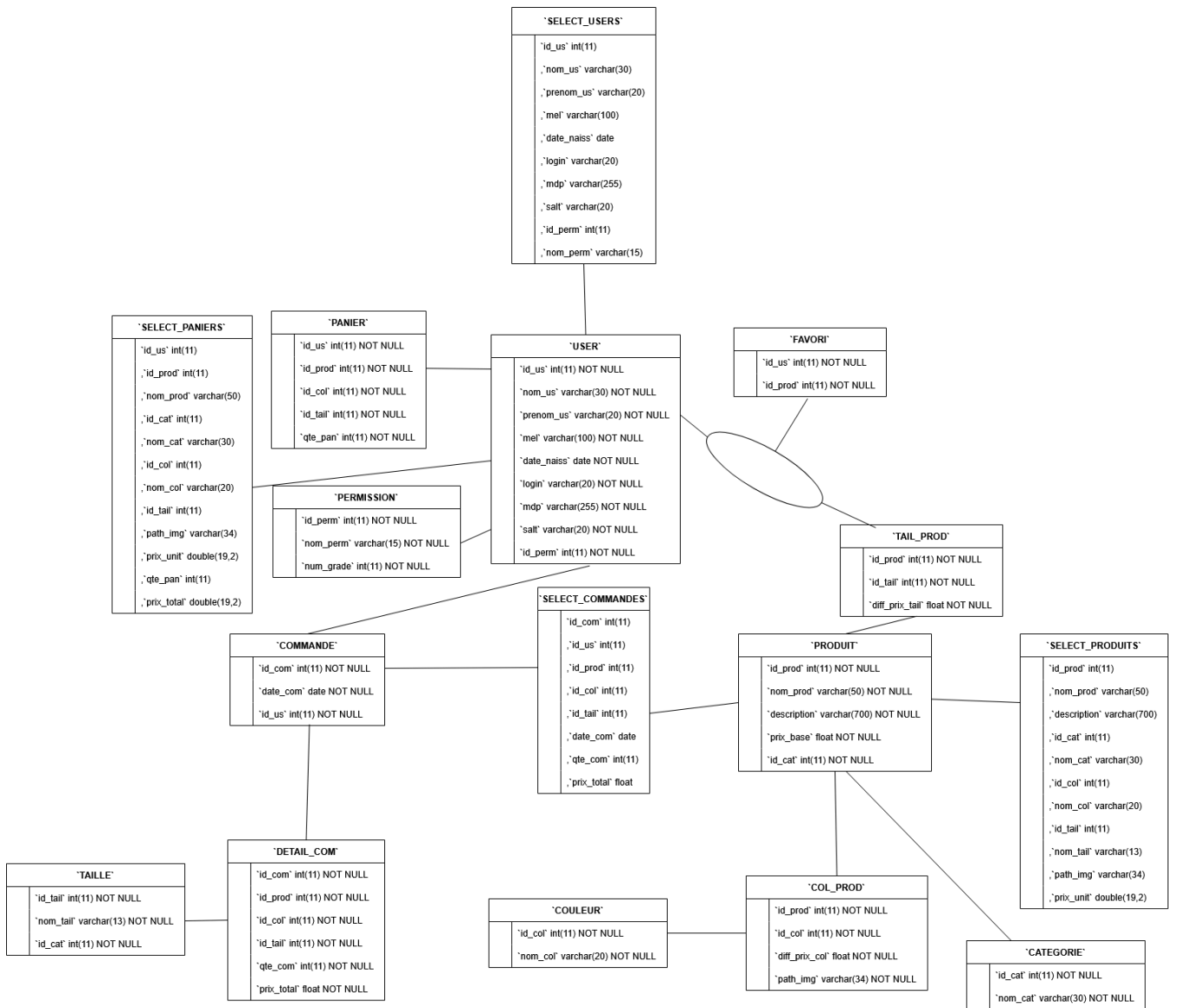
PHP:

Optimisation de la connexion à la BDD: création d'un singleton

Optimisation de l'api:

- Regroupement des requêtes
- Optimisation des requêtes en adéquation avec la BDD
- Classe pour le PDO

SCHÉMAS BASE DE DONNÉE



RÉPARTITION DES TÂCHES

Analyse de la base de donnée : Théo

Mise en place du schéma E/A : Théo

Analyse du code : Arthur / Nicolas