

ROCK'N'CLOTHES

Projet tutoré

Dossier Technique

Elisa Lescarret, Corentin Berthelet, Cyril Keil et Racel Moshiur

14/06/2015

Table des matières

Introduction.....	2
I. Organisation et arborescence logique du site.....	2
II. Le schéma relationnel de la base de données	3
III. Différence entre Java et Oracle	4
A. Java	4
B. Oracle	4
IV. La base de données	4
V. Programme Java	4
Conclusion	4

Introduction

Nous avons réalisé un projet qui avait pour but de rassembler du web, du Java et de la base de données. Ce projet, nous a fait réaliser un site Web en relation avec une base de données pour la partie client et une partie administrateur en Java, toujours en relation avec la même base de données.

Nous allons vous présenter les solutions techniques que nous avons appliquées pour mener à bien ce projet.

I. Organisation et arborescence logique du site

Pour notre site, nous avons fait usage de Frameworks afin de faciliter l’implémentation : Bootstrap pour CSS et JQuery pour Javascript.

Bootstrap permet d’obtenir une mise en page simple et adaptable à tous les supports de navigation.

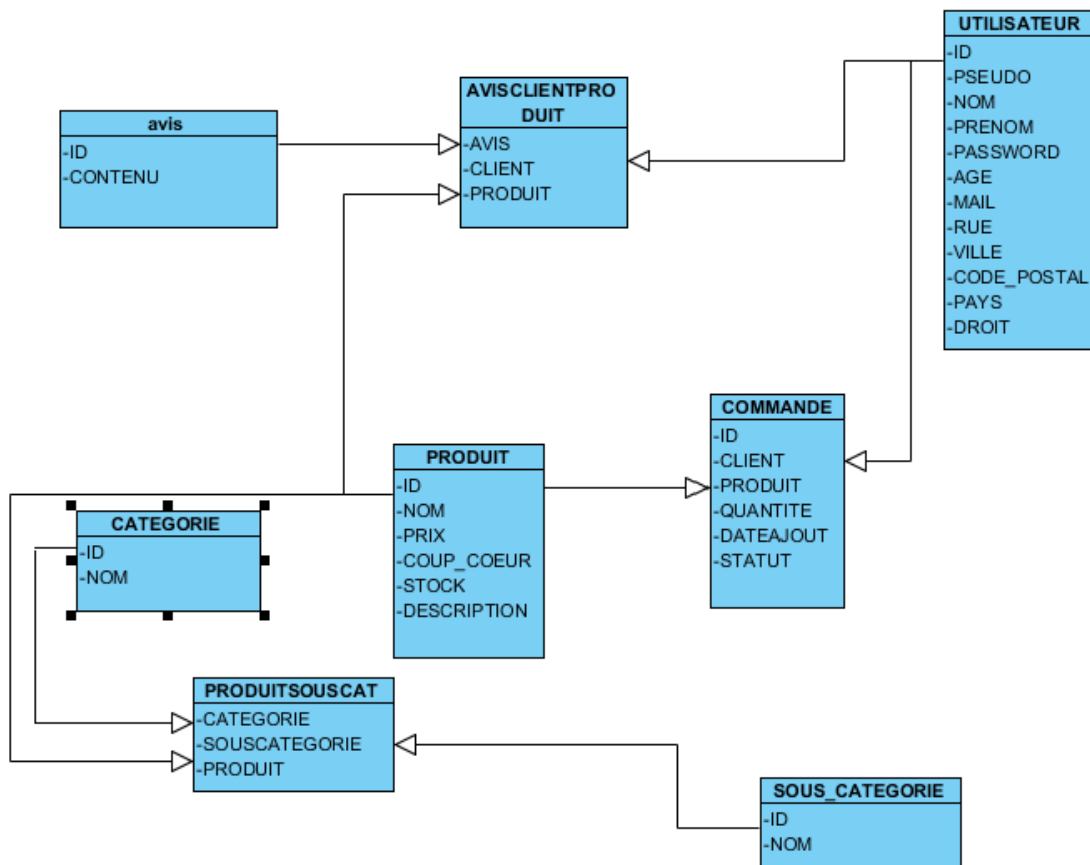
JQuery est un framework qui adopte une seule syntaxe Javascript simplifiée. Il est dès lors facilement adaptable à tous les navigateurs. Ainsi, l’usage du JQuery permet dans certains cas de passer de 50 lignes à 10 lignes de code, d’où un gain certain en lisibilité.

L’arborescence de notre site est organisée comme suit :

- Le répertoire **class** regroupe les classes telles que « catégorie », « produit », « client », « avis », « bdd » et « config ». Le fichier contenant la classe est standardisé ainsi : **nom_de_la_classe.class.php**.
- Le répertoire **controller** contient les contrôleurs utilisés pour les différentes fonctionnalités du site. Ceux-ci sont de la forme **controllerSpecification.php** (la partie bleu correspond au nom du contrôleur, elle diffère donc selon le contrôleur).
- Le répertoire **model** contient les fichiers .php décrivant les fonctions associées aux modèles.
Les fichiers du modèle sont standardisés ainsi : **nom_de_la_table.model.php**.
- Les différentes vues de site sont regroupés au sein du répertoire **view**.
- Le répertoire **style** contient les fichiers .css que nous avons créés (qui ne sont pas issus de bootstrap).
- Le répertoire **js** regroupe quant à lui les scripts JQuery que nous avons créés.
- Le dossier **lib** contient les librairies utilisées par notre site (Bootstrap et JQuery en l’occurrence).
- Enfin, le répertoire **img** regroupe les images du site. Voici un exemple de son arborescence :

Img -> image_produit -> categorie_1 -> souscategorie_1->produit_1

II. Le schéma relationnel de la base de données



Le schéma relationnel de notre base de données montre que nous avons deux tables principales qui sont :

-produit, défini par un ID, un nom, un prix, un coup de cœur, un stock et une description

-utilisateur, défini par un ID, un pseudo, un nom, un prénom, un password, un âge, une adresse mail, une rue, une ville, un code postal, un pays, un droit (si l'utilisateur est un client, dans ce cas « droit » aura la valeur 2, ou un administrateur, « droit » aura la valeur 1 dans ce cas).

Une commande est définie par un ID, une quantité, une date d'ajout qui est la date où la commande est passée, un statut. La commande est reliée aux tables "produit" et "utilisateur" car elle est passée par un utilisateur pour un produit donné.

Un produit possède des catégories et des sous catégories qui sont reliées grâce à la table "produitsouscat".

Un avis peut être donné par un utilisateur pour un produit donné grâce à la table "avisclientproduit".

III. Différence entre Java et Oracle

A. Java

Idéalement, chacune des données saisies par l'utilisateur par le biais de l'application doit être vérifiée. Par exemple, lors de l'ajout d'un produit, un utilisateur ne pourra pas valider un ajout comportant des champs vides (de type NULL). D'autre part, l'application a la possibilité de vérifier l'authenticité des identifiants de l'utilisateur (usage d'expression régulière).

B. Oracle

Au niveau de la base de données, Oracle doit s'assurer de la conformité des données (exemple : pas de valeurs négatives pour le stock d'un produit). De plus, le SGBD doit également s'assurer de la non redondance des informations au sein de la base de données (exemple : deux utilisateurs ou deux produits ayant le même ID).

IV. La base de données

Voir fichier script création de BD.sql

V. Programme Java

L'application Java est développée selon le design pattern MVC (Modèle, Vue, Contrôleur).

- **Modèle** : Ce package contient les différentes classes représentant les objets figurant dans la base de données tels que les Utilisateurs ou les Produits par exemple.
- **Vue** : Ce répertoire est constitué des différentes vues de l'application. Celles-ci sont développées à l'aide de la technologie Oracle **JavaFX**. Celle-ci permet de modéliser au mieux l'interface graphique de l'application, notamment grâce à l'outil **SceneBuilder**.
Ce dernier génère les vues au format (f)xml.
- **Contrôleur** : Enfin, on retrouve ici les contrôleurs permettant d'une part l'enchaînement logique des vues, mais également les fonctionnalités associées à celles-ci.

Conclusion

Nous avons utilisé plusieurs technologies pour mener à bien ce projet.

Pour le site web nous avons utilisé les Frameworks Bootstrap et JQuery. Pour l'application java, nous avons utilisé JavaFX.

Enfin, la base de données a été créée grâce à Oracle. Cette dernière partie ayant été moins conséquente que prévue, Corentin a participé au site en ce qui concerne la mise en page en CSS.