

Documentation Technique

Oracle-JEE

Radigoy frederic, Lloyd Dingald, Yann Cedric Moratinos,
Karima makhtas, Florent Ferrand



Historique des révisions

Révision	Date	Modification
A	Juin 2014	Création
B	Juin 2014	Modification

Table des matières

- 1) Présentation du projet
 - a. Contexte
 - b. Description de la demande
 - c. Analyse des besoins
 - i. Besoins fonctionnels
 - d. Solutions proposées
 - i. Principes
 - ii. Fonctionnalités
 - iii. Maquette de la solution – Cahier des charges
 - iv. Outils de développement utilisés
- 2) Dossiers d'analyse
- 3) Développement du projet
 - a. Normalisation
 - i. Normalisation de la charte graphique
 - ii. Normalisation de la base de données
 - b. Architecture logiciel
 - i. Design pattern DTO
 - ii. Constitution
- 4) Gestion de projet
 - a. Outils et moyens techniques, mise en œuvre
- 5) Problèmes rencontrés

1) Présentation du projet

a. Contexte

Au cours de ce projet Oracle-JEE, on nous a permis de traiter les différentes phases d'une étude logicielle :

- L'analyse de la demande et du besoin
- La conception en mettant en œuvre UML
- La conception et la réalisation de l'application
- Les tests
- La livraison et le déploiement

L'équipe est constituée de cinq personnes :

- Lloyd Dingalt : Développeur
- Karima Makhtas : Développeur
- Florent Ferrand : Développeur
- Yann-Cédric Moratinos : Développeur
- Frédéric Radigoy : Développeur

Le sujet de ce projet, consiste en l'élaboration d'une base de données permettant de faire vivre le front-office d'un site web e-commerce et une application JEE/Oracle exploitant cette base de données front-office.

L'application devra répondre aux différentes attentes fonctionnelles formulées pour la fiche projet Oracle / Objet Semestre 2.

b. Description de la demande

Objectifs :

- Créer une base de données permettant de gérer le back-office d'un site web e-commerce.
- Créer des procédures stockées **pertinentes** permettant d'obtenir ou transmettre des informations nécessaires au cycle de vie du site (gestion des stocks, réapprovisionnement, calcul du bénéfice moyen annuel)
- Créer une base de données permettant de faire vivre le front-office d'un site web e-commerce.
- Créer des procédures stockées **pertinentes** permettant au site de faire vivre une commande du début à la fin avec des informations indispensables (top des ventes, catégories les plus consultées, les articles les mieux notés)
- Créer l'application JEE/ORACLE exploitant votre base de données front-office et back-office. Ce peut être une seule base de donnée ou plusieurs. Créer un projet web java avec Maven/Hibernate-JPA/Servlet-JSP-JSF. Créer une interface graphique simple permettant de mettre en évidence quelques-unes de vos procédures. Pas plus de 4 ou 5 écrans sont attendus mais tout soin particulier apporté au design sera apprécié.

c. Analyse des besoins

i. Besoins fonctionnels

- Faire des choix fonctionnels judicieux et pertinents quant à la création de vos procédures
- Manipuler le langage PL-SQL au travers d'un exemple
- Faire des choix les plus optimisés possibles en termes de performance
- Remettre un travail propre, soigné et professionnel
- Être capable de programmer une application JEE minimaliste
- Être capable d'exploiter Oracle via JEE
- Savoir adapter sa BDD aux besoins applicatifs sans détériorer les performances
- Être capable de mener un projet du début à la fin en équipe
- Savoir vendre son produit

d. Solutions proposées

i. Principes

Afin de permettre une utilisation simple, intuitive, et rendre plus conviviale notre programme, nous avons mis en place une Interface Homme Machine (IHM) simple qui sert de point de contact entre l'utilisateur et la base de données. Cette IHM a été réalisée en respectant une charte graphique très précise.

ii. Fonctionnalités

1. Liste des fonctionnalités de l'application par acteurs

- Un **site marchand** qui vend ses produits sur Internet, assurant le suivi des commandes de ses clients et la livraison des produits commandés.
- Un **keymaster** qui sera le responsable de sécurité de la plateforme hébergeant le site du marchand.
- Un **correspondant** de la solution de paiement qui sera l'interlocuteur privilégié du marchand qu'il aidera et conseillera tout au long du partenariat établi entre le site et la solution.
- Une **solution de paiement** qui gère le service et qui recouvre la gestion et le paiement des commandes effectuées sur le site du marchand.
- Une **banque** qui assure le débit des clients et le crédit du marchand pour les commandes effectués.
- Un **client** qui effectue un achat de biens ou de service en utilisant la solution de paiement.
- Un **administrateur** chez le site marchand qui s'occupe du paramétrage de la plateforme et de la gestion des commandes clients passées sur son site.

iii. Outils de développement utilisés



Editeur de code source : Eclipse IDE



JavaEE : Technologie de programmation web



ORM : Hibernate : un framework open source gérant la persistance des objets en base de données relationnelle.



Un système de gestion de base de données relationnel (SGBDR).

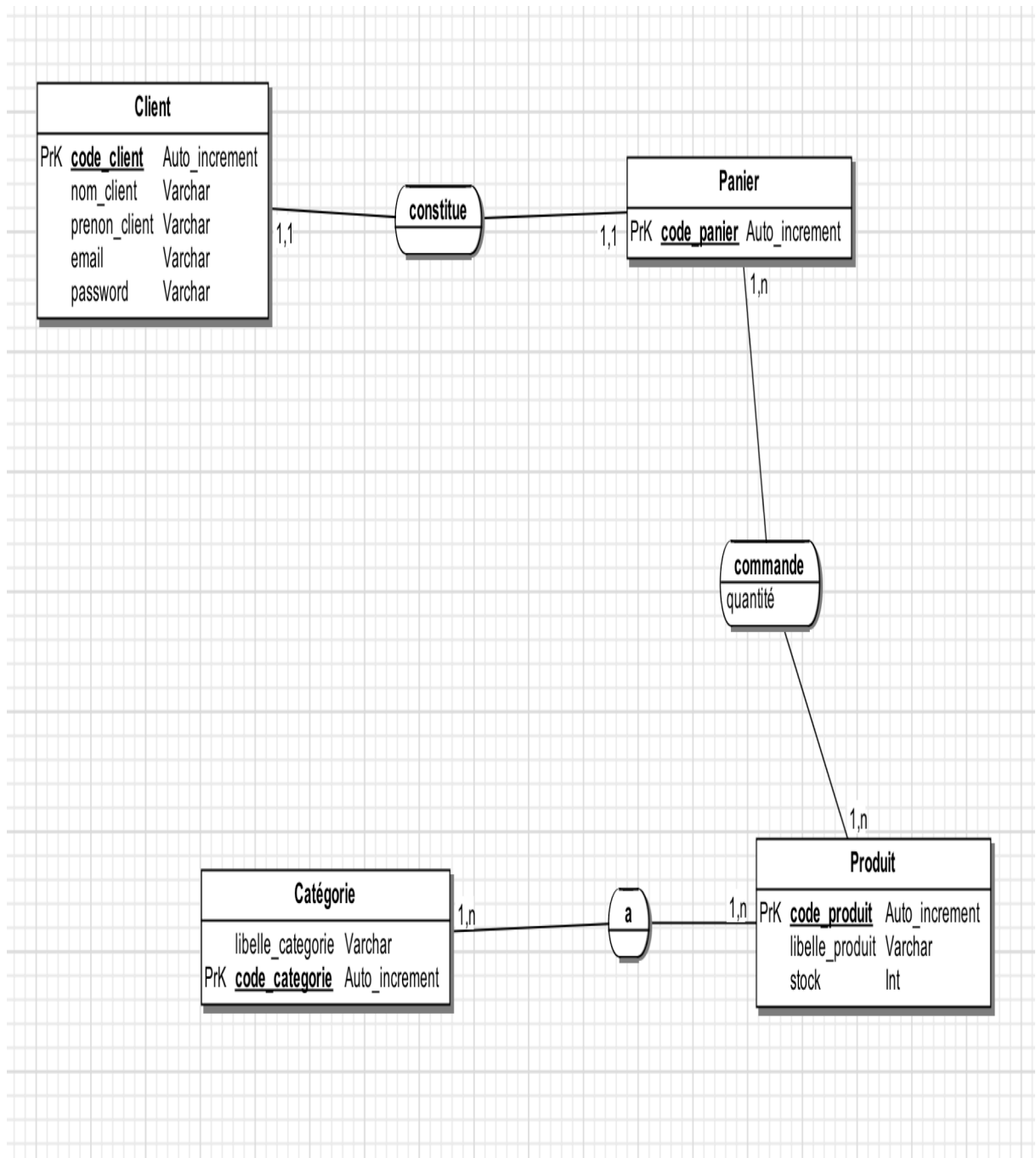


Un conteneur web libre de servlets et JSP Java EE



Spring est un framework libre pour construire et définir l'infrastructure d'une application java², dont il facilite le développement et les tests.

2) Dossier d'analyse



3) Développement du projet

a. Normalisation

i. Normalisation de la charte graphique

La première chose que nous avons faite durant la phase de développement a été de créer des feuilles de styles (**CSS***), où nous avons déterminés les couleurs à utiliser dans notre application, la forme des boutons, les différentes polices de caractères, les images d'arrière plan

ii. Normalisation de la base de données

Lors de la création de la base de données, nous avons pensé à plusieurs normes a respecté afin de faciliter le développement de notre logiciel :

- Non utilisation d'accents et de caractères spécifique dans le nom des tables et des champs qui composent celles-ci.
- Pas d'espaces dans les noms de champs.
- Première lettre en majuscule des mots qui composent les noms de champs

b. Architecture logicielle

i. Architecture logicielle n-tiers

L'architecture n-tiers est devenu un classique dans le milieu du logiciel. Il n'est plus rare de trouver de telles architectures, même pour des logiciels de petite taille. Avec l'avancé d'Internet et des navigateurs en tout genre, il sera de plus en plus fréquent de trouver de telles architectures. De plus en plus d'applications, en particulier développé en Java prene cette architecture afin de pouvoir donné une application disponible par Navigateur interposé, tout en étant relativement court à charger (petite taille de code).

L'application du modèle plus général qu'est le multi-tier. L'architecture logique du système est divisée en trois niveaux ou couches :

- couche présentation ;
- couche métier ;
- couche accès aux données.

ii. Design pattern

La structuration des applications se traduit par une décomposition logique de chaque application en 5 couches :

- Présentation
- Contrôleur
- Services
- Domaine
- Persistance

Chaque couche a ses propres responsabilités et utilise la couche située en dessous d'elle. En fonction du projet, les architectes enrichissent et élaguent le modèle. La struturation est alors guidée par les contraintes exprimées et existantes.

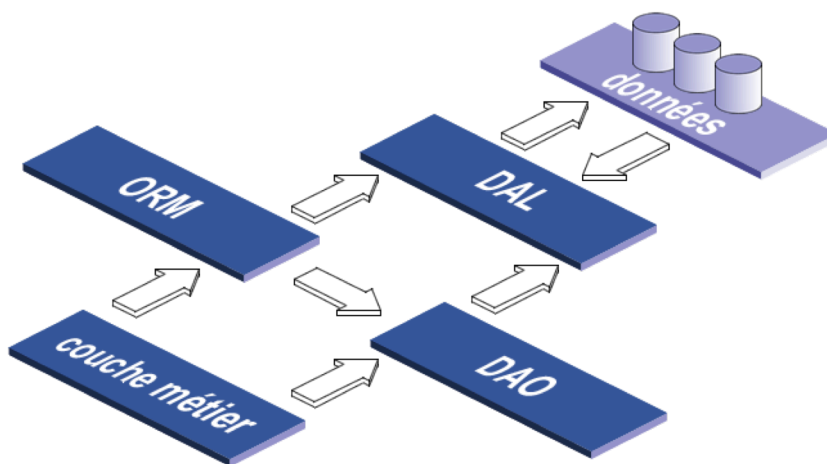
Design pattern DTO : Couche abstraction de données

Désign pattern DAL : Objet d'accès aux données

La DAL Permet de s'abstraire du support des données. Pour se faire elle met à disposition des méthodes génériques permettant d'accomplir des actions de maintenances sur les données.

L'ORM (Object / Relation Mapping) : Mapping objet / relationnel

L'ORM a pour but de transformer les relations entre les tables d'une base de données en relations entre objets et inversement. Elle va typiquement se préoccuper de matérialiser les clés étrangères par des dépendances entre objets.



4) Gestion de projet

a. Outils et moyens techniques, mise en oeuvre

i. Github

Afin de pouvoir stocker en ligne les différents documents, analyses, comptes rendus de réunions, programmes et les rendre accessibles depuis n'importe quel ordinateur interne ou externe. Cet environnement nous a permis l'accès permanent à notre planning et à la totalité des éléments partageables entre nous, constituant ainsi une source unique d'informations. Un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.

ii. Skype

Tout au long de ce projet, nous avons mise en place des réunions permettant de faire le point sur l'avancement grace à skype.

iii. Developpez.net

Ce sont des forums où j'ai posté des questions sur les problèmes que j'ai rencontré durant ce projet.

5) Problèmes rencontrés

- Configuration de Spring framework/
- Coordination au niveau du travail d'équipe
- Le principal problème technique que l'équipe a dut affronter a été d'apprendre le développement orienté Objet Java EE couplé à Oracle, cette étape a été dès le départ sous-estimée en temps d'apprentissage ce qui nous a fortement ralentis dans la création de notre application.