

Compte Rendu — Application Catalogue de Produits

Compte Rendu — Application Catalogue de Produits.....	1
Présentation générale.....	3
Contexte et besoins.....	3
Architecture de l'application.....	3
Modules de l'application.....	4
Module 1 — Authentification (package login).....	4
Module 2 — Gestion du catalogue (package view).....	4
Module 3 — Couche métier (package Metier).....	6
Module 4 — Génération de rapports PDF.....	7
Base de données.....	8
Points forts de l'application.....	8
Pistes d'amélioration envisageables.....	9

Présentation générale

Cette application Java de bureau a été développée dans le cadre d'un projet de gestion commerciale. Elle permet à une entreprise de gérer l'ensemble de son catalogue de produits de manière centralisée, sécurisée et intuitive. L'application offre une interface graphique complète développée avec la bibliothèque Swing de Java, connectée à une base de données MySQL pour la persistance des données.

L'objectif principal est de fournir un outil opérationnel permettant à un utilisateur authentifié de consulter, rechercher, ajouter, modifier et supprimer des produits, tout en ayant la possibilité de générer des rapports PDF détaillés pour un usage professionnel.

Contexte et besoins

Dans un contexte commercial, la gestion d'un catalogue de produits nécessite un outil fiable qui centralise toutes les informations relatives aux articles disponibles. L'application répond aux besoins suivants :

- Sécuriser l'accès aux données par un système d'authentification
 - Permettre une consultation rapide et filtrée des produits
 - Offrir une gestion complète des données produits (création, lecture, mise à jour, suppression)
 - Organiser les produits par catégories pour une meilleure lisibilité
 - Produire des documents PDF exploitables pour des rapports ou des inventaires
 - Permettre la gestion des accès utilisateurs
-

Architecture de l'application

L'application suit une architecture en trois couches distinctes, ce qui permet de séparer clairement les responsabilités de chaque partie du code.

La couche Présentation (view) regroupe toutes les interfaces graphiques visibles par l'utilisateur. C'est la partie avec laquelle l'utilisateur interagit directement : les fenêtres, les boutons, les tableaux, les formulaires.

La couche Métier (Metier) contient la logique applicative. Elle traite les demandes provenant de la couche présentation, applique les règles de gestion et communique avec la base de données. C'est le cœur fonctionnel de l'application.

La couche Données assure la connexion et les échanges avec la base de données MySQL. Elle est matérialisée par la classe SingletonConnection qui garantit qu'une seule connexion est maintenue tout au long de l'utilisation de l'application.

Cette séparation en couches facilite la maintenance, les évolutions futures et la lisibilité du code.

Modules de l'application

Module 1 — Authentification (package login)

Ce module est le point d'entrée de l'application. Il regroupe trois classes distinctes qui gèrent l'ensemble du cycle d'authentification et de gestion des accès utilisateurs.

La fenêtre de connexion (LoginFrame)

Lorsque l'application est lancée, la première fenêtre affichée est l'écran de connexion. Cette interface demande à l'utilisateur de saisir son identifiant et son mot de passe dans deux champs dédiés. Un bouton "Valider" déclenche la vérification des informations saisies.

Concrètement, lorsque l'utilisateur clique sur "Valider", l'application vérifie que les deux champs ne sont pas vides. Si l'un des champs est vide, un message d'erreur s'affiche invitant l'utilisateur à remplir tous les champs. Si les deux champs sont remplis, l'application interroge la base de données pour vérifier si un utilisateur correspondant à l'identifiant et au mot de passe saisi existe bien dans la table users. Si la combinaison est correcte, l'utilisateur est redirigé vers l'interface principale du catalogue. Dans le cas contraire, un message d'erreur est affiché et le champ mot de passe est automatiquement vidé pour permettre une nouvelle tentative, le curseur revenant automatiquement sur le champ identifiant pour faciliter la nouvelle saisie.

La fenêtre d'accueil utilisateur (AccueilUtilisateur)

Une fois authentifié, l'utilisateur peut accéder à une page d'accueil intermédiaire qui lui propose deux actions principales. La première est la déconnexion, qui ferme la session en cours et redirige vers l'écran de connexion après confirmation. La seconde est le changement de mot de passe, qui ouvre une nouvelle fenêtre dédiée à cette opération.

La fenêtre de changement de mot de passe (ChangerMdp)

Cette interface permet à l'utilisateur connecté de modifier son mot de passe. Elle affiche un champ de saisie sécurisé pour entrer le nouveau mot de passe, ainsi qu'un bouton "Valider" pour confirmer la modification. Lorsque l'utilisateur valide, la mise à jour est effectuée directement dans la base de données via une requête SQL paramétrée qui cible l'enregistrement correspondant à l'identifiant de l'utilisateur actuellement connecté. Un message de confirmation s'affiche pour informer l'utilisateur que le changement a bien été pris en compte.

Module 2 — Gestion du catalogue (package view)

Ce module constitue le cœur fonctionnel de l'application du point de vue de l'utilisateur. Il regroupe les interfaces graphiques permettant de consulter et gérer l'ensemble des produits du catalogue.

L'interface principale du catalogue (CatalogueSwingJFrame)

Cette interface est la fenêtre principale de l'application, celle avec laquelle l'utilisateur passe le plus de temps. Elle est organisée en plusieurs zones fonctionnelles distinctes.

La zone supérieure de la fenêtre contient les outils de recherche et de filtrage. On y trouve un champ de texte permettant de saisir un mot-clé, un bouton de recherche qui déclenche la requête, une liste déroulante affichant toutes les catégories disponibles, un champ affichant l'identifiant de la catégorie sélectionnée, et un bouton de déconnexion.

La zone centrale est occupée par un tableau de données qui affiche les produits correspondant aux critères de recherche. Ce tableau comporte cinq colonnes : l'identifiant du produit, sa désignation, son prix, sa quantité en stock et sa catégorie.

La recherche par mot-clé fonctionne de la manière suivante : lorsque l'utilisateur saisit un mot ou une partie de mot dans le champ de recherche et clique sur le bouton OK, l'application interroge la base de données pour trouver tous les produits dont la désignation contient la chaîne de caractères saisie. La recherche est insensible à la position du mot-clé, ce qui signifie qu'une recherche sur "clav" renverra par exemple "Clavier mécanique" et "Mini-clavier". Les résultats s'affichent immédiatement dans le tableau.

La recherche par catégorie permet de filtrer les produits selon leur catégorie. Lorsque l'utilisateur sélectionne une catégorie dans la liste déroulante, l'application interroge d'abord la base de données pour récupérer l'identifiant numérique correspondant au nom de la catégorie sélectionnée, puis affiche dans le tableau tous les produits appartenant à cette catégorie. L'identifiant de la catégorie s'affiche automatiquement dans le champ prévu à cet effet, en lecture seule.

L'interface avancée du catalogue (CatalogueSwingV2)

Il s'agit d'une version enrichie de l'interface principale, qui intègre les mêmes fonctionnalités de recherche et de filtrage mais y ajoute une gestion complète des produits (CRUD) ainsi que des fonctionnalités de génération de rapports PDF.

La fenêtre est organisée en trois zones principales. En haut se trouve la barre de recherche avec tous les filtres disponibles ainsi que les boutons de génération PDF et de déconnexion. Au centre s'affiche le tableau des produits. En bas est positionné le formulaire de gestion des produits.

Le formulaire de gestion des produits permet d'effectuer toutes les opérations sur les produits. Il comprend des champs pour saisir l'identifiant du produit, son nom, son prix, sa quantité, ainsi qu'une liste déroulante pour choisir sa catégorie. Quatre boutons complètent ce formulaire : Ajouter, Modifier, Supprimer et Nouveau.

Lorsque l'utilisateur clique sur une ligne du tableau, les informations du produit sélectionné se remplissent automatiquement dans le formulaire, permettant une modification ou une

suppression rapide et intuitive. Le champ identifiant devient alors non modifiable pour éviter toute confusion ou erreur.

Module 3 — Couche métier (package Metier)

Ce module regroupe toutes les classes qui constituent la logique applicative de l'application. Il fait office d'intermédiaire entre les interfaces graphiques et la base de données.

L'interface IMetier

Cette interface définit le contrat que doit respecter toute implémentation de la couche métier. Elle liste les opérations disponibles : ajouter une catégorie, ajouter un produit, consulter les produits par mot-clé, récupérer toutes les catégories, consulter les produits d'une catégorie, et récupérer une catégorie avec ses produits. Le fait de passer par une interface garantit la cohérence et facilite d'éventuelles évolutions futures.

La classe MetierCatalogueImp

C'est l'implémentation concrète de l'interface IMetier. Elle contient toutes les méthodes qui interagissent avec la base de données via des requêtes SQL paramétrées, ce qui protège l'application contre les injections SQL.

La méthode d'ajout d'une catégorie exécute une requête d'insertion dans la table categorie avec le nom fourni. La méthode d'ajout d'un produit insère un nouvel enregistrement dans la table produit avec toutes ses caractéristiques et l'identifiant de la catégorie à laquelle il appartient.

La méthode de recherche par mot-clé utilise l'opérateur LIKE de SQL avec un joker de chaque côté du terme recherché, ce qui permet de trouver le mot-clé quelle que soit sa position dans la désignation du produit. Pour chaque produit trouvé, la méthode effectue une requête supplémentaire pour récupérer les informations de la catégorie correspondante et les associer au produit.

La méthode de récupération de tous les produits charge l'ensemble du catalogue depuis la base de données, en associant à chaque produit sa catégorie. Cette méthode est utilisée pour l'affichage initial et après chaque opération de modification.

La classe propose également des méthodes pour mettre à jour un produit, le supprimer et le récupérer par son identifiant, même si ces méthodes sont prévues pour être complétées dans une future version.

La classe SingletonConnection

Cette classe applique le patron de conception Singleton pour gérer la connexion à la base de données. L'idée est qu'une seule et unique connexion MySQL est établie au démarrage de l'application et réutilisée pour toutes les opérations. Cela évite d'ouvrir et fermer une connexion à chaque requête, ce qui serait coûteux en ressources et en temps. La connexion

est établie vers la base de données Catalogue sur le serveur local MySQL, en utilisant le pilote JDBC de MySQL.

La classe Produit

Cette classe représente un produit du catalogue. Elle définit les attributs d'un produit : son identifiant, sa désignation, son prix, sa quantité en stock et la catégorie à laquelle il appartient. Elle dispose de constructeurs, de getters et setters pour accéder et modifier ces attributs, ainsi qu'une méthode toString pour afficher les informations d'un produit sous forme de chaîne de caractères.

La classe Categorie

Cette classe représente une catégorie de produits. Elle contient l'identifiant de la catégorie, son nom, sa date de création et la liste des produits qui lui sont associés. Comme pour la classe Produit, elle dispose de tous les accesseurs nécessaires ainsi que d'une méthode toString.

La classe ProduitModel

Cette classe joue un rôle crucial dans l'affichage des données dans le tableau de l'interface graphique. Elle étend AbstractTableModel, ce qui lui permet d'être directement utilisée comme modèle de données pour le composant JTable de Swing. Elle définit les cinq colonnes du tableau (ID, Désignation, Prix, Quantité, Catégorie) et fournit une méthode loadData qui reçoit une liste de produits et met à jour l'affichage du tableau en conséquence. Chaque fois que de nouvelles données sont chargées, la méthode fireTableChanged est appelée pour notifier le tableau qu'il doit se rafraîchir.

Module 4 — Génération de rapports PDF

Ce module est intégré à l'interface principale du catalogue et permet de produire des documents PDF professionnels directement depuis l'application, sans nécessiter d'outil externe.

Rapport de la liste des produits

Ce rapport génère un document PDF contenant un tableau structuré avec toutes les informations des produits : identifiant, nom, catégorie, prix et quantité. En bas du tableau figure un résumé indiquant le nombre total de produits présents dans le catalogue. Le document mentionne également la date et l'heure de génération.

Rapport de la liste des catégories

Ce rapport produit un document PDF listant toutes les catégories disponibles avec leur identifiant, leur nom et le nombre de produits qui leur sont rattachés. Un total général est indiqué en bas du document.

Rapport complet du catalogue

Ce rapport combine les deux précédents en un seul document. Il présente d'abord la liste complète des produits, puis la liste des catégories, et termine par des statistiques globales comprenant le nombre total de produits, la valeur totale du stock calculée en multipliant le prix de chaque produit par sa quantité, et une identification du produit le plus cher du catalogue.

Visualisation et téléchargement

Lorsque l'utilisateur clique sur l'un des boutons PDF, une nouvelle fenêtre s'ouvre avec un aperçu du contenu du document sous forme de texte formaté. Cette fenêtre affiche également des informations sur le document généré (date, taille en kilo-octets) et propose trois actions : télécharger le fichier PDF sur le disque en choisissant l'emplacement via une boîte de dialogue, imprimer le document directement depuis l'application, ou simplement fermer la fenêtre d'aperçu.

La génération des PDF est réalisée en mémoire avant l'affichage, ce qui garantit des performances optimales et évite de créer des fichiers temporaires inutiles sur le disque de l'utilisateur.

Base de données

La base de données MySQL utilisée par l'application s'appelle Catalogue et contient trois tables principales.

La table **produit** stocke toutes les informations relatives aux articles du catalogue : un identifiant textuel (idProduit), une désignation (designationProduit), un prix numérique décimal (prix), une quantité entière (quantite) et une clé étrangère (idCategorie) qui fait le lien avec la table categorie.

La table **categorie** contient les informations sur les familles de produits : un identifiant entier auto-incrémenté (idCategorie), un nom (nomCategorie) et une date de création automatiquement renseignée (dateCreation).

La table **users** gère les comptes utilisateurs avec un identifiant de connexion (login) et un mot de passe (mdp). Cette table est consultée à chaque tentative de connexion pour vérifier les informations saisies par l'utilisateur.

Points forts de l'application

L'application présente plusieurs atouts notables. La sécurisation des accès par authentification garantit que seuls les utilisateurs autorisés peuvent consulter et modifier les données du catalogue. La séparation claire du code en packages distincts facilite la maintenance et les évolutions futures. L'utilisation de requêtes SQL paramétrées protège contre les tentatives d'injection SQL. L'interface intuitive avec remplissage automatique du

formulaire lors de la sélection d'un produit améliore l'ergonomie et réduit les risques d'erreur de saisie. La génération de rapports PDF directement depuis l'application évite toute ressaisie et garantit la cohérence des données exportées avec les données affichées.

Pistes d'amélioration envisageables

Quelques fonctionnalités pourraient être ajoutées dans une version future pour enrichir l'application. La gestion multi-utilisateurs avec des niveaux de droits différents (administrateur, lecteur seul) permettrait un meilleur contrôle des accès. L'implémentation complète des méthodes de mise à jour et de suppression dans la couche métier renforcerait les capacités CRUD. L'ajout d'une fonctionnalité de gestion des catégories (ajout, modification, suppression depuis l'interface graphique) offrirait plus de flexibilité. Enfin, une pagination du tableau pour les catalogues comportant un grand nombre de produits améliorerait les performances d'affichage.