

# PROJET SHOPPING- POO JAVA

Equipe 4

- Jérémy Populaire
- Angel Velasco
- Michael Adda
- Grégoire Marchal



**SHOPPING**  
**CLUB**

# Sommaire

- I. Présentation du Projet et de la répartition interne
- II. Diagramme de classes
- III. Modèle relationnel de la BDD
- IV. Design de l'interface graphique
- V. Bilan individuel et collectif

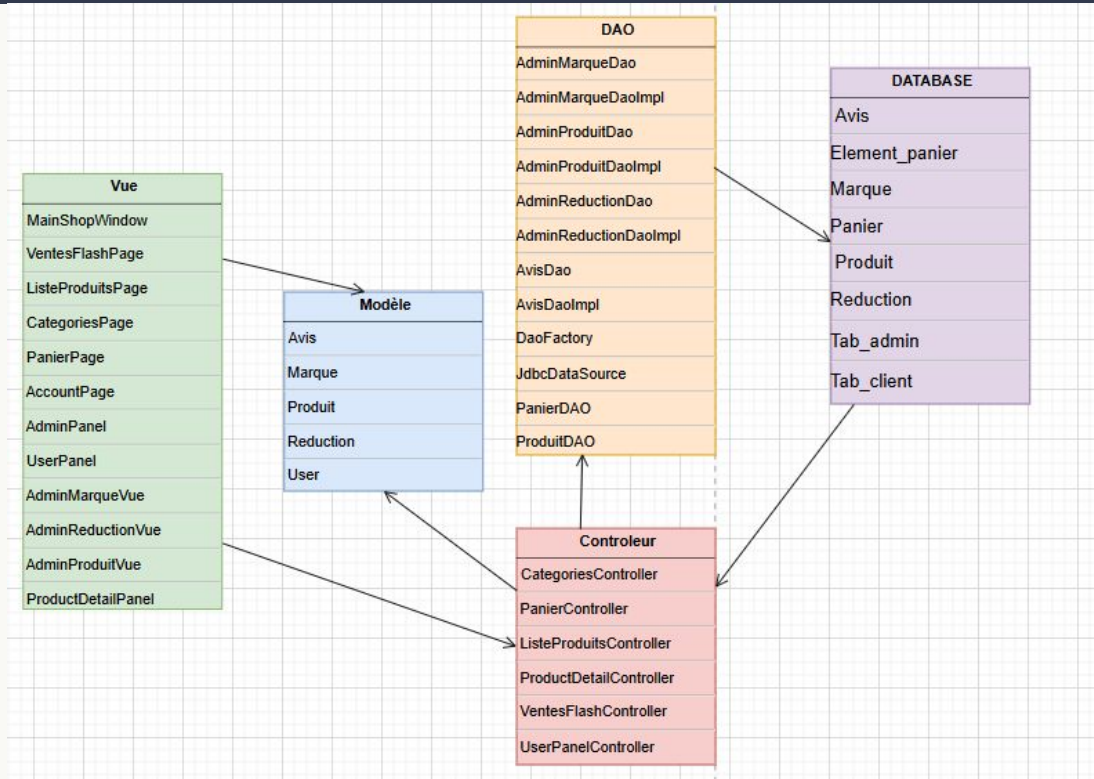
# Les Objectifs ?

<b>But du projet</b>	Concevoir une application Java simulant une interface d'achat permettant aux clients d'acheter des articles avec éventuelles réductions et facturation
<b>Fonction principale</b>	Gérer un catalogue d'articles à vendre, les commandes clients, les remises, les paiements, et conserver les historiques de ventes et clients
<b>Type d'Utilisateurs</b>	<ul style="list-style-type: none"><li>- Nouveaux clients : inscription avec identifiant et mot de passe.</li><li>- Anciens clients : connexion, accès à l'historique et aux notes de commande</li></ul>
<b>Navigation et recherche</b>	Parcours d'un catalogue par marques, recherche d'articles, possibilité d'acheter plusieurs unités d'un même article
<b>Gestion des prix</b>	Articles vendus à l'unité et en lot à tarif réduit. Calcul hybride selon la quantité commandée
<b>Rôle des clients</b>	Parcourir les articles, passer commande, consulter la facture et l'historique
<b>Rôle des administrateurs</b>	Gérer l'inventaire, les promotions, les profils clients et consulter des statistiques de vente

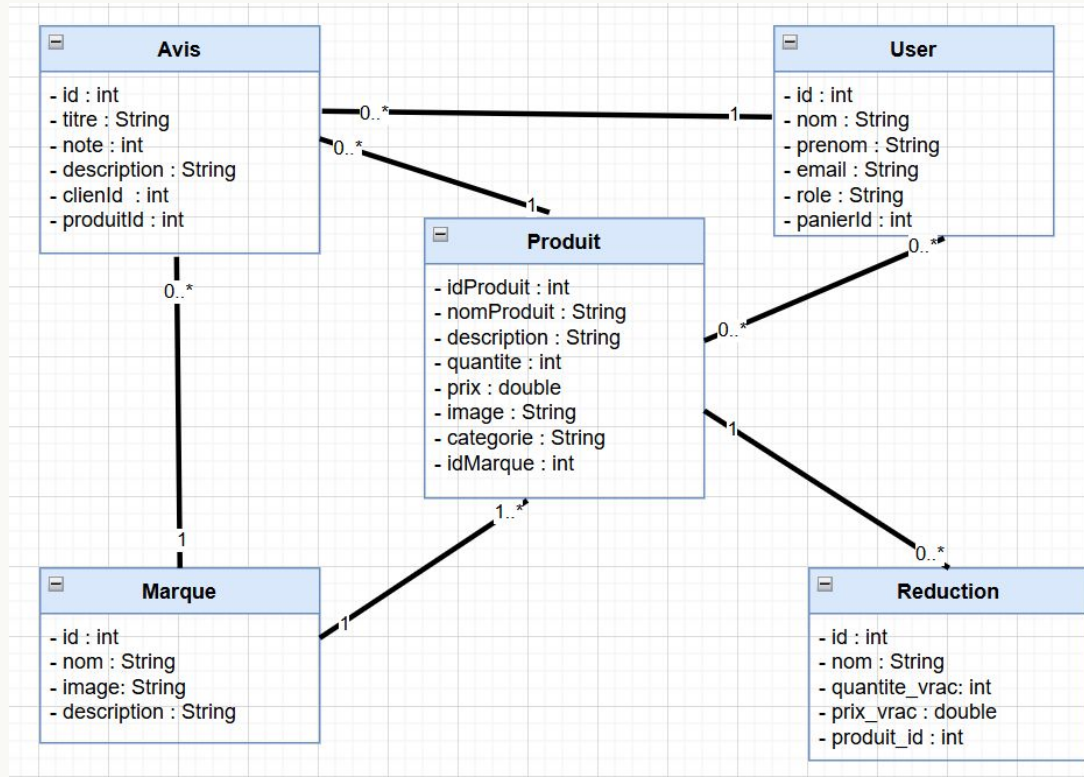
# Notre Organisation

	Conception BDD + Relation entre les entités (DAO)	Interface de connection et utilisateurs (admin/ client)	Marque et Réduction	Page Vente Produit + Ventes Flash + Catégorie	Page Panier	Avis + Historique
Angel	X	X			X	
Jérémy	X			X		X
Michael	X		X	X		X
Grégoire	X	X		X		

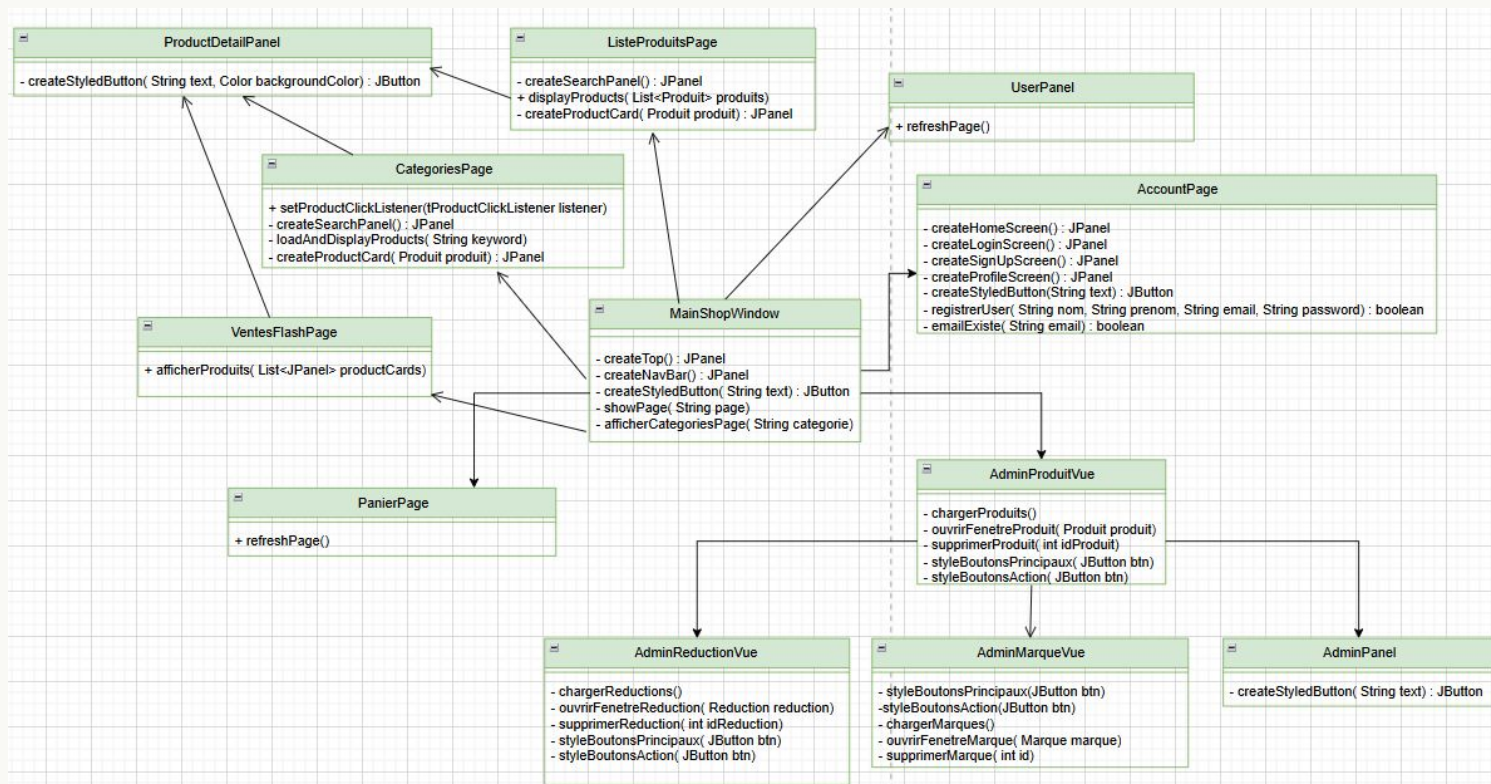
# Diagrammes de Classes : Architecture Générale



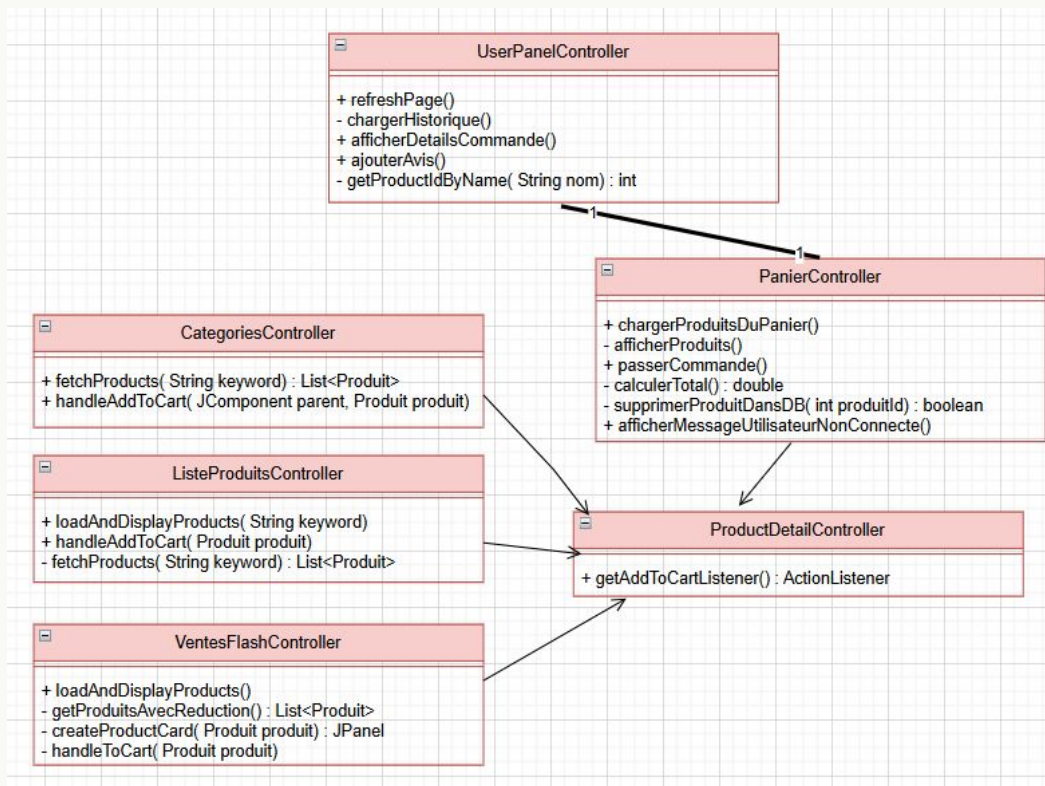
# Diagrammes de Classes : Package Modèle



# Diagrammes de Classes : Package Vue



# Diagrammes de Classes : Package Contrôleur





# Modèle Relationnel

Avis (id, titre, note, description, #produit\_id, #user\_id)

Element\_panier (id, quantite, #produit\_id, #user\_id)

Marque (id, nom, image, description)

Panier (id, taille, #utilisateur\_id)

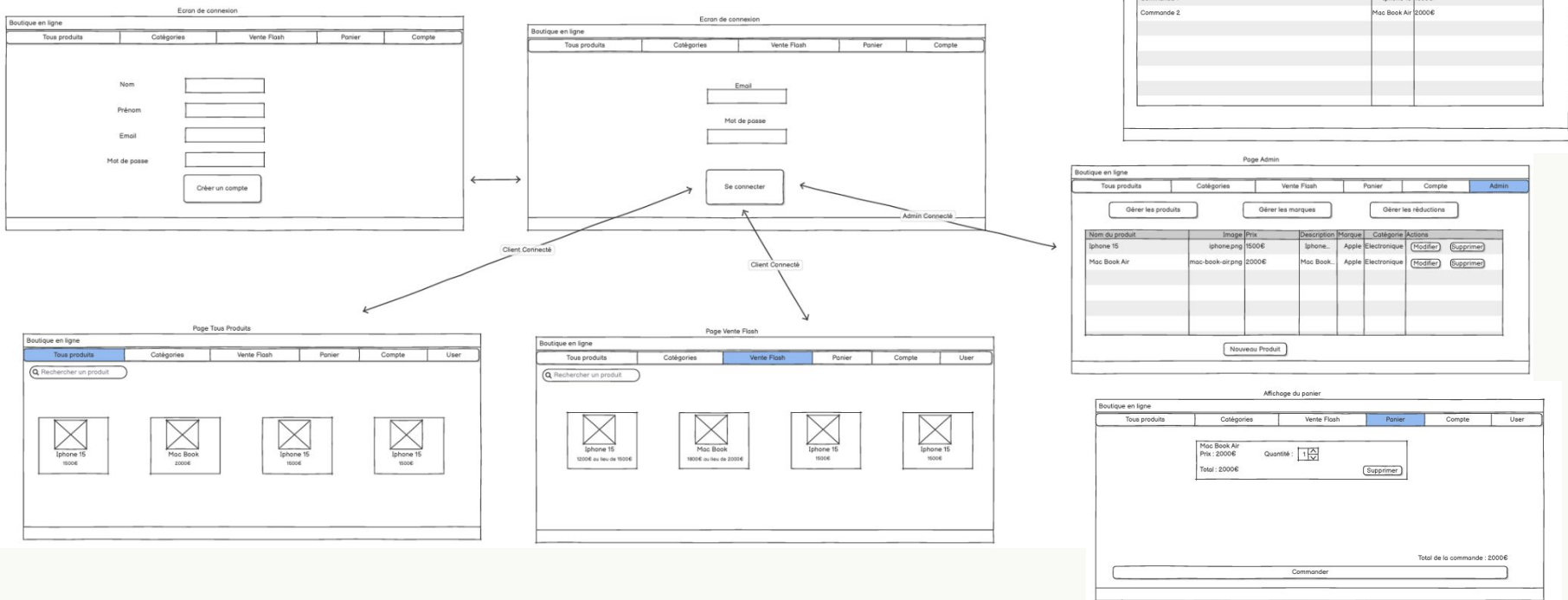
Produit (id, nom, image, prix, quantite, description, category, #marque\_id)

Reduction (id, nom, quantite\_vrac, prix\_vrac, #produit\_id)

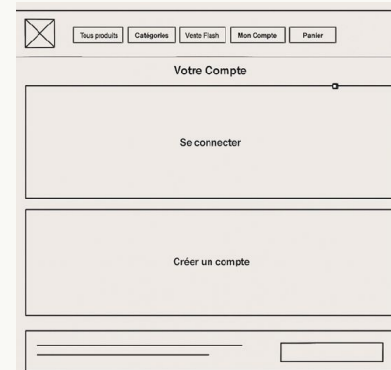
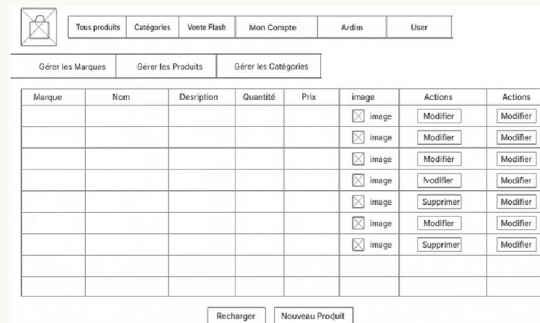
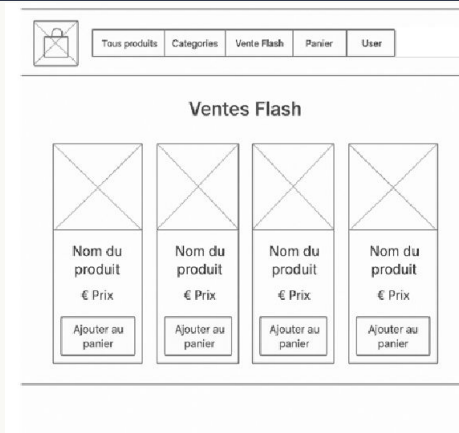
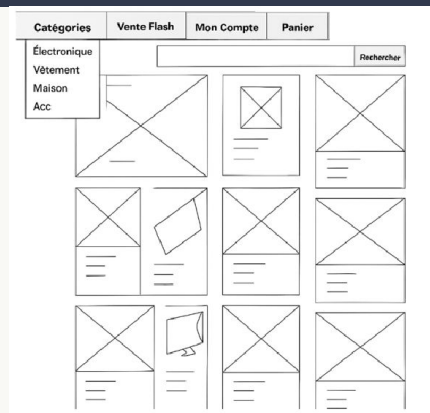
Tab\_admin (id, nom, prenom, MotDePasse, rôle, date\_inscription, #email)

Tab\_client (id, nom, prenom, MotDePasse, date\_inscription, #email)

# Storyboard



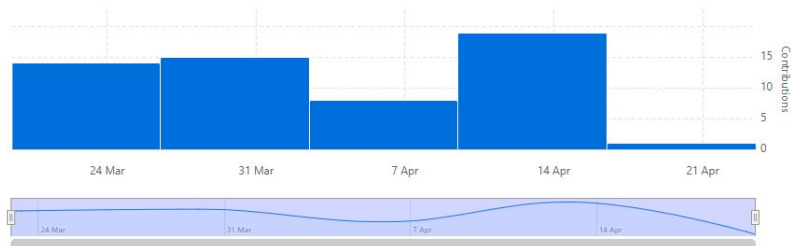
# Wireframes



# Versioning GIT

## Commits over time

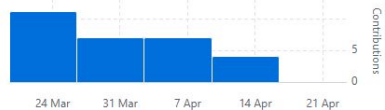
Weekly from 23 mar 2025 to 20 abr 2025



 **angv5845**

29 commits 330,090 ++ 1,767 --

#1 ...



 **GregECE**

13 commits 1,720 ++ 1,738 --

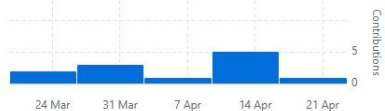
#2 ...



 **MichaelAdda7**

12 commits 2,784 ++ 412 --

#3 ...



 **Jeereemy**

3 commits 325 ++ 0 --

#4 ...



# Bilan individuel

	Compétences acquises	Points d'amélioration
Angel	<ul style="list-style-type: none"><li>- Aptitude à créer des flux de données</li><li>- Compréhension des principes MVC</li></ul>	<ul style="list-style-type: none"><li>- Améliorer le développement d'interfaces utilisateur</li></ul>
Jérémy	<ul style="list-style-type: none"><li>- Mise en place d'une architecture claire en Java</li><li>- Lien entre Java et la base de données</li></ul>	<ul style="list-style-type: none"><li>- Optimiser la gestion des données pour éviter les traitements répétitifs</li></ul>
Michael	<ul style="list-style-type: none"><li>- Maîtrise des différentes gestions de classe Java</li><li>- Relation base de données et Java</li></ul>	<ul style="list-style-type: none"><li>- Améliorer les visuels graphiques</li></ul>
Grégoire	<ul style="list-style-type: none"><li>- Maîtrise de l'interaction et de l'utilisation entre Modèle, Vue et Contrôleur</li><li>- Conception d'interfaces graphiques interactives avec Swing</li></ul>	<ul style="list-style-type: none"><li>- Mieux maîtriser l'optimisation des performances et réduire la duplication de code</li></ul>

# Bilan collectif

## **Méthodologie et Organisation de l'équipe**

- Répartition des tâches selon les compétences de chacun, avec des points réguliers pour synchroniser l'avancement
- On a avancé étape par étape : d'abord la base de données, puis la logique du programme, l'interface, et plusieurs tests intermédiaires et finaux
- Utilisation efficace de Git pour centraliser le code et faciliter les intégrations
- Bonne cohésion de groupe, grâce à une entraide sur les aspects techniques les plus difficile

# Bibliographie sources

## **Sources:**

- Boostcamp (Cours/ TP/ TD POO Java )
- Geeksforgeeks
- Stack Overflow : <https://stackoverflow.com/questions>