

Concevez la solution technique d'un système de gestion de pizzeria



Document des spécifications fonctionnelles

PANCARTE Marjorie

Sommaire

I. Contenu

II. Contexte	3
III. Diagramme de classe	4
IV. Descriptif des classes	5
V. Diagramme de composant	8
VI. Diagramme déploiement	9
VII. Justification des choix	9
VIII. MPD	10

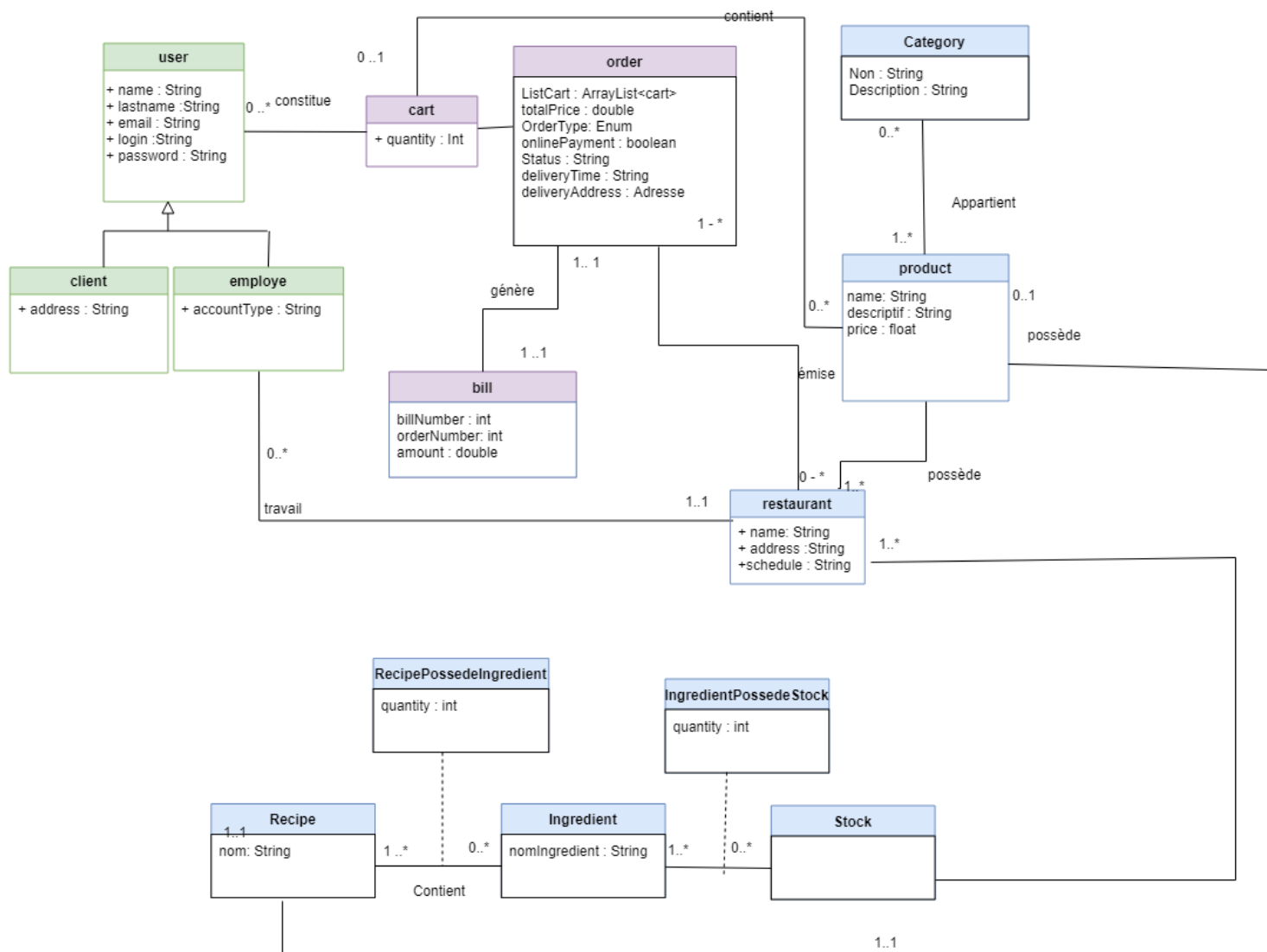
II. Contexte

Le groupe OC Pizza souhaite optimiser la gestion de son entreprise qui compte actuellement 5 pizzerias avec une prévision d'ouverture de 3 points de vente supplémentaires.

Suite à la prise de contact avec un responsable, il en ressort que le système devra permettre :

- d'être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation
- de suivre en temps réel les commandes passées et en préparation
- de suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas sont encore réalisables
- de proposer un site Internet pour que les clients puissent :
 - passer leurs commandes, en plus de la prise de commande par téléphone ou sur place,
 - payer en ligne leur commande s'ils le souhaitent – sinon, ils paieront directement à la livraison
 - modifier ou annuler leur commande tant que celle-ci n'a pas été préparée
- de proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza

III. Diagramme de classe



IV. Descriptif des classes

Classe *User* : La classe « User » représente un utilisateur qu'il soit client ou employée avec ses informations dans la base .

Nom des attributs	Type des attributs	Détails
name	String	Prenom de l'utilisateur
lastName	String	Nom de l'utilisateur
Email	String	
login	String	
password	String	

Classe *client* : La classe « Client » représente un client ,cette classe hérite des informations de la classe « User » avec ses informations dans la base .

Nom des attributs	Type des attributs	Détails
address	String	

Classe *employe* : La classe « Employe » représente un employé ,cette classe hérite des informations de la classe « User » avec ses informations dans la base .

Nom des attributs	Type des attributs	Détails
AccountType	String	Type de compte

Classe *cart* : La classe « Cart » représente le panier qui sera par l'utilisateur.

Nom des attributs	Type des attributs	Détails
quantity	Integer	

Classe *order* : La classe « *Order* » représente la commande qui sera passé l'utilisateur avec toutes les informations relatives à la commande.

Nom des attributs	Type des attributs	Détails
listCart	String	
totalPrice	Integer	
orderType	String	
onlinePaiement	boolean	
Status	String	
deliveryTime	String	
deliveryAddress	String	

Classe *bill* : La classe « *Bill* » représente la facture qui sera généré avec toutes les informations relatives à la commande.

Nom des attributs	Type des attributs	Détails
oderNumber	Integer	
amount	Integer	

Classe *restaurant* : La classe « *Restaurant* » représente le restaurant ou la commande sera passé par l'utilisateur avec toutes les informations relatives à la commande.

Nom des attributs	Type des attributs	Détails
name	String	
address	String	
schedule	String	horraires

Classe *product* : La classe « *Product* » représente les produits vendu par le restaurant avec toutes les informations relatives à la commande.

Nom des attributs	Type des attributs	Détails
name	String	
description	String	
price	Integer	

Classe *Category* : La classe « *Category* » représente la catégorie des produits vendu par le restaurant avec toutes les informations relatives à la commande.

Nom des attributs	Type des attributs	Détails
name	<i>String</i>	
description	<i>String</i>	

Classe *Recipe* : La classe « *Recipe* » représente les recettes des produits vendu par le restaurant avec toutes les informations relatives à la commande.

Nom des attributs	Type des attributs	Détails
name	<i>String</i>	

Classe *RecipePossedeIngredient* : La classe « *RecipePossedeIngredient* » représente les ingrédients possédé pour chaque recettes des produits vendu par le restaurant avec toutes les informations relatives à la commande.

Nom des attributs	Type des attributs	Détails
quantity	<i>Integer</i>	

Classe *Ingredient* : La classe « *Ingredient* » représente les ingrédients que propose le restaurant avec toutes les informations relatives à la commande.

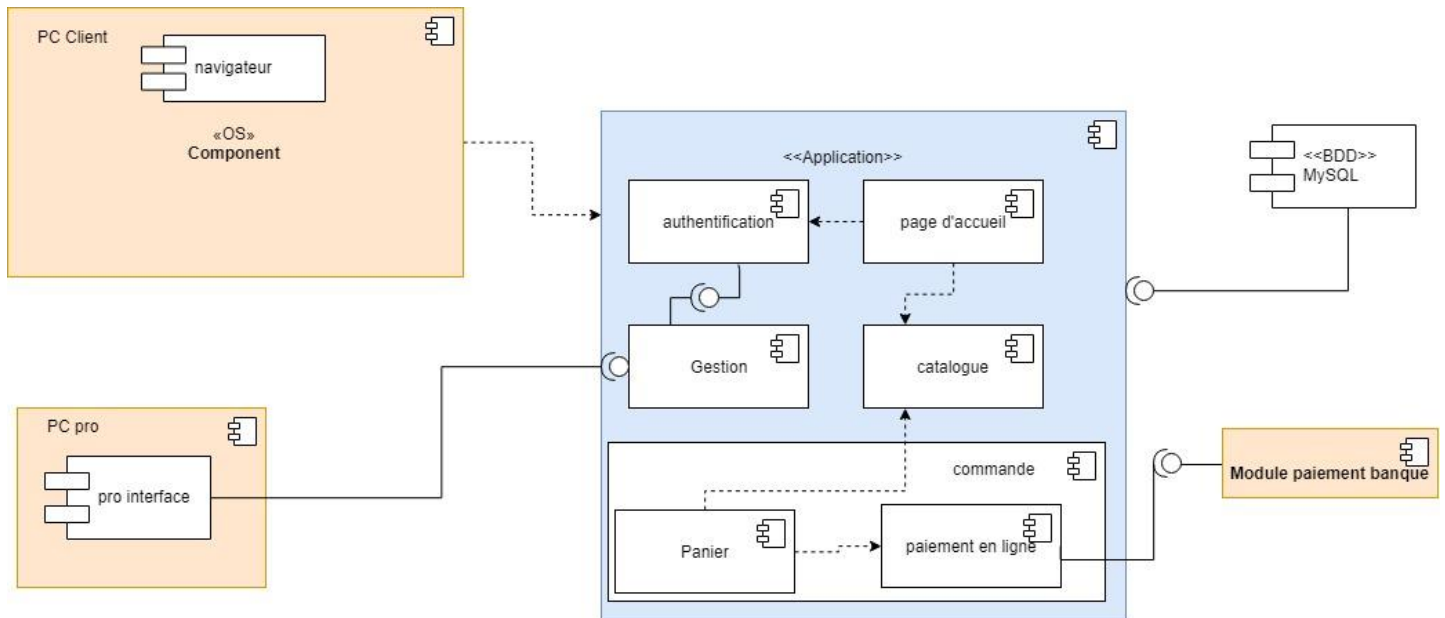
Nom des attributs	Type des attributs	Détails
nomIngredient	<i>String</i>	

Classe *IngredientPossedeStock*: La classe « *IngredientPossedeStock* » représente les ingrédient stocker t avec toutes les informations relatives à la commande.

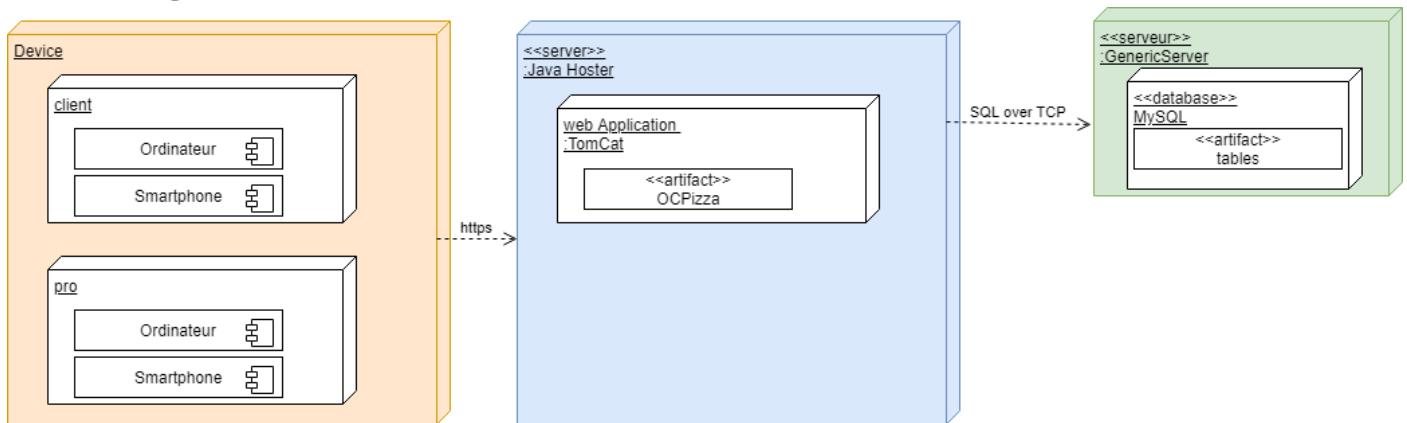
Nom des attributs	Type des attributs	Détails
quantity	<i>Integer</i>	

Classe *stock*: La classe « *Stock* » représente le stock du restaurant avec toutes les informations relatives à la commande.

V. Diagramme de composant



VI. Diagramme déploiement



VII. Justification des choix

Après l'étude des demandes émises, on en ressort les diagrammes précédents, le diagramme de composant et le diagramme de composant.

le diagramme de composant se compose de :

- Périphériques pour les utilisateurs (portable, Pc) et aussi pour l'équipe du restaurant qui seront accessibles depuis un navigateur.
- Un module d'authentification : qui sera le composant qui va permettre aux utilisateurs de créer leur compte et de se connecter
- Un module de gestion : qui sera le composant qui va permettre aux utilisateurs professionnels de gérer les paramètres du restaurant
- Un module page d'accueil pour accueillir les utilisateurs
- Un module catalogue pour consulter les produits
- Un module commande : pour commander qui sera relié au module de paiement
- un package pour la base de données qui sera elle en MySQL

le diagramme de composant se compose de :

- Une partie « device » pour les object connecté utilisé par les clients ou les employés nécessaire pour la connection
- Un serveur heberger pas java hoster
- Un serveur generique pour la base de donnée MySQL qui contiendra les table nécessaire pour le bon fonctionnement de l'applicaition web

Le choix d'un serveur à part pour le base de donnée est fait pour des raisons de sécurité

VIII. MPD

