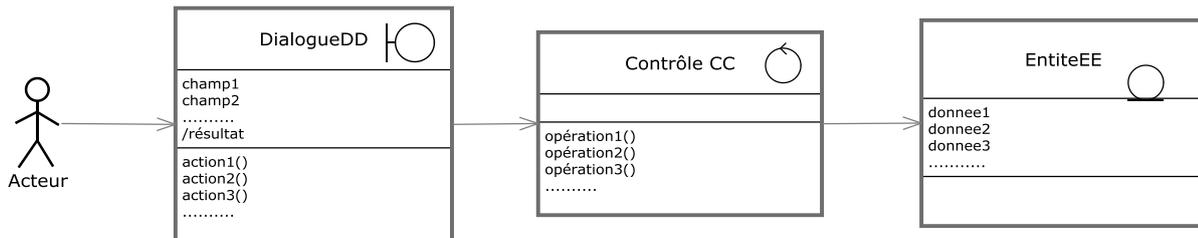


Diagramme des classes participantes

Le diagramme de classes participantes est important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, les modèles de la couche métiers et l'interface avec l'utilisateur. Il semble particulièrement important pour guider la phase de production du livrable final.

C'est cette importance qui nous a poussés à concevoir un tel diagramme dans le souci d'une phase de développement claire et efficace.

On utilisera alors une implémentation de l'architecture 3-tiers, le pattern Modèle-Vue-Contrôleur (MVC).



On découpera par la suite le diagramme des classes participantes par cas d'utilisation pour une meilleure visibilité et une meilleure compréhension.

■ Diagramme 1 :

On a regroupé les différents cas d'utilisation qui effectuent **une lecture sur la table Produit et/ou ProduitStock** :

- Effectuer une recherche
- Effectuer une recherche avancée
- Voir la fiche produit
- Consulter la liste des produits par catégorie de produits
- Voir la fiche de réapprovisionnement

Les classes de dialogue :

- **Accueil** : c'est la classe qui représente la classe de dialogue de la page d'accueil. A partir de cette page sont accessibles toutes les autres pages web du site. On remarquera néanmoins la méthode rechercheRapide(keyword : String) prenant une chaîne de caractères en paramètre et retournant un booléen : vrai si la requête s'est bien passée et retourne un résultat, faux en cas d'erreur. Cette méthode est contenue dans la classe Accueil car la page d'accueil de l'application contient une zone de texte pour saisir des mots-clés pour une recherche rapide sur la table Produit.
- **RechercheAvancee** : cette classe représente une page web accessible à partir du menu de la page d'accueil et contient un formulaire permettant de sélectionner des critères afin d'affiner la recherche dans la table Produit. Elle contient les méthodes : rechercheParReference(r: String), rechercheParDesignation(d: String), rechercheParCategorie(c: Integer) qui prennent en paramètres ces critères. Cependant, ces méthodes peuvent être combinées pour permettre un maillage encore plus fin lors des recherches.
- **ResultatRecherche** : présente dans une page web les résultats des appels de méthode de rechercheRapide() ou des méthodes de RechercheAvancee. Sa méthode afficherRResultat() permet de présenter à travers les éléments d'affichage de la page les données.

- FicheProduit : présente dans une page web les données propres à un produit. Sa méthode afficherProduit(r : String) affiche les informations sur la page du produit dont la référence à été passée en paramètre.
- FicheDeReapp : représente une page web qui affiche les produits dont les quantités sont en deçà d'un certain seuil d'alerte propre à chaque produit. Sa méthode afficherProduitsEnAlerte() permet d'afficher dans la page les données retournées par un appel à la méthode produitsEnAlerte()(cf les classes contrôle).
- ProduitParCategorie : affiche la liste des produits regroupés par catégorie de produit. La méthode listeProduitsParCategorie(c: Integer) prend en paramètre le code de la catégorie concernée et retourne un booléen pour indiquer l'échec ou la réussite de l'appel.

Classes de contrôle :

- ProduitLecture : comme la table Produit est au centre du package de la gestion de stock, elle contient relativement beaucoup de méthodes par rapport aux autres classes, d'où notre motivation à la diviser en 2 classes de contrôle : ici, ProduitLecture regroupe toutes les méthodes d'accès en lecture à la table Produit. Ainsi, les méthodes des classes dialogue décrites précédemment font appel, dans leur exécution, aux méthodes de ProduitLecture.
- ProduitStock : utile ici pour les jointures d'où la relation avec la classe de contrôle ProduitLecture afin de retourner les différentes quantités des produits.

Classes du modèle :

- Produit : c'est l'implémentation dans la base de données, sous forme de table relationnelle, de la classe métier Produit. Elle ne contient donc que des attributs qui correspondent aux champs de la table. Ce sont les méthodes de la classe de contrôle ProduitLecture qui sont les seules à avoir accès aux données persistantes stockées dans Produit.
- ProduitStock

La navigabilité à sens unique des classes dialogue vers les classes contrôle puis des classes contrôle vers les classes du modèle illustrent l'architecture MVC adoptée ici.

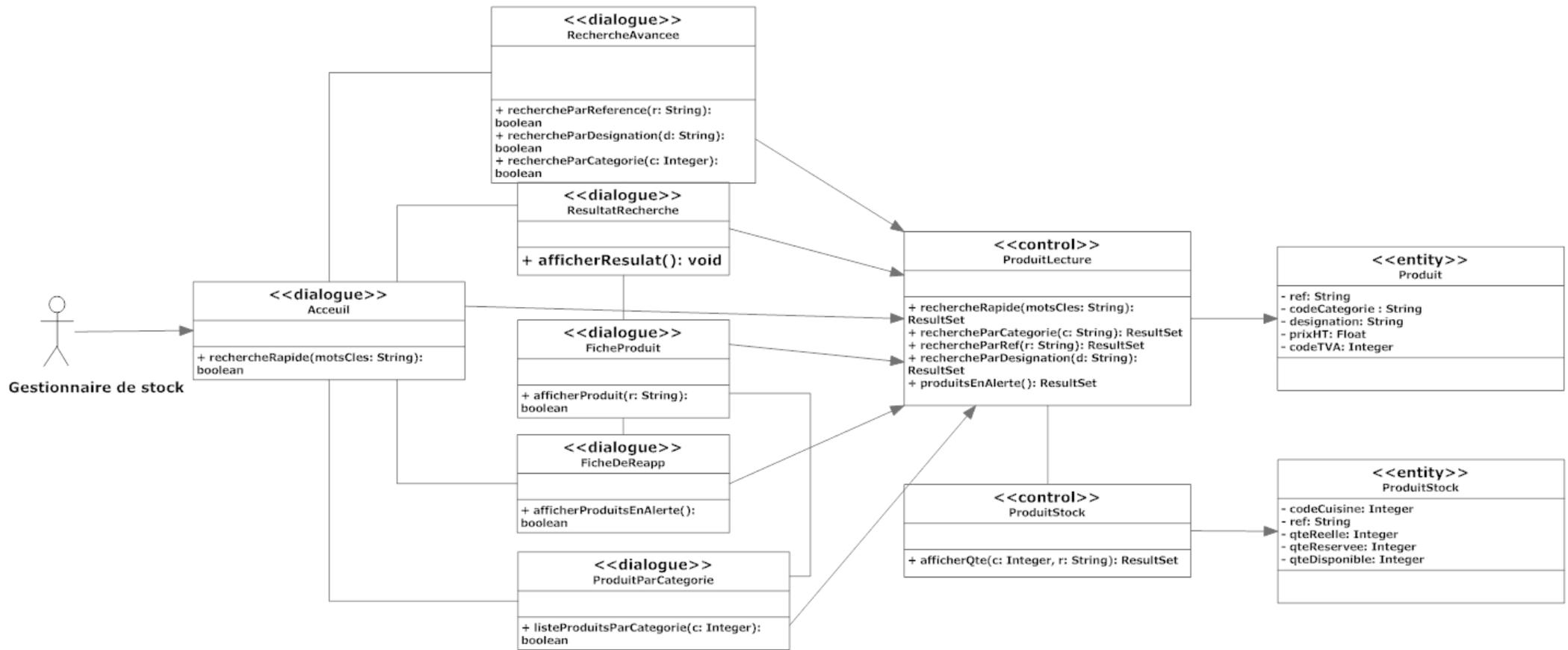


Diagramme de classes participantes 1

▪ Diagramme 2 :

Cas d'utilisation concerné : Consulter historique des prix

Classes de dialogue :

- HistoriquePrix : affiche dans une page l'historique des évolutions du prix des différents articles du catalogue. La méthode afficher() présente les données sur la page et filtrer() permet de filtrer l'affichage selon les produits, leur catégorie.

Classes de contrôle :

- HistoPrix : classe qui effectue les accès à la table HistoriquePrix. La méthode afficherHistorique() fait un opération en lecture tandis que HistoriquePrix() une opération en écriture.

Classes du modèle :

- HistoriquePrix : c'est la table HistoriquePrix.

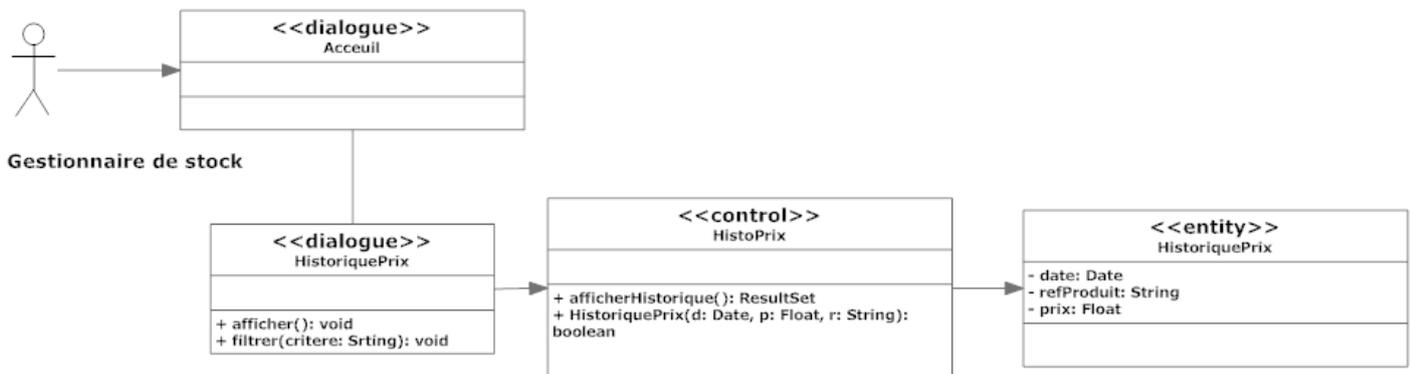


Diagramme de classes participantes 2

▪ Diagramme 3 :

Cas d'utilisation concerné : afficher réservations

Classes de dialogue :

- Reservations : sa méthode affiche dans la page le résultat des requêtes

Classes de contrôle :

- LigneDeCommande : la méthode afficherReservations effectue une opération en lecture dans la table des lignes de commande pour retourner les produits en réservations par date et par commande. Cette classe de contrôle est implémentée par l'équipe gérant la prise de commande.
- Commande : ce contrôle est en relation avec le contrôle LigneDeCommande pour effectuer les différentes jointures. Il est implémenté aussi par l'équipe qui gère la prise de commande.

Classes du modèle :

- Commande : on y récupère le statut de la commande
- LigneDeCommande : on y récupère les réservations des commandes dont le statut est « en cours ».

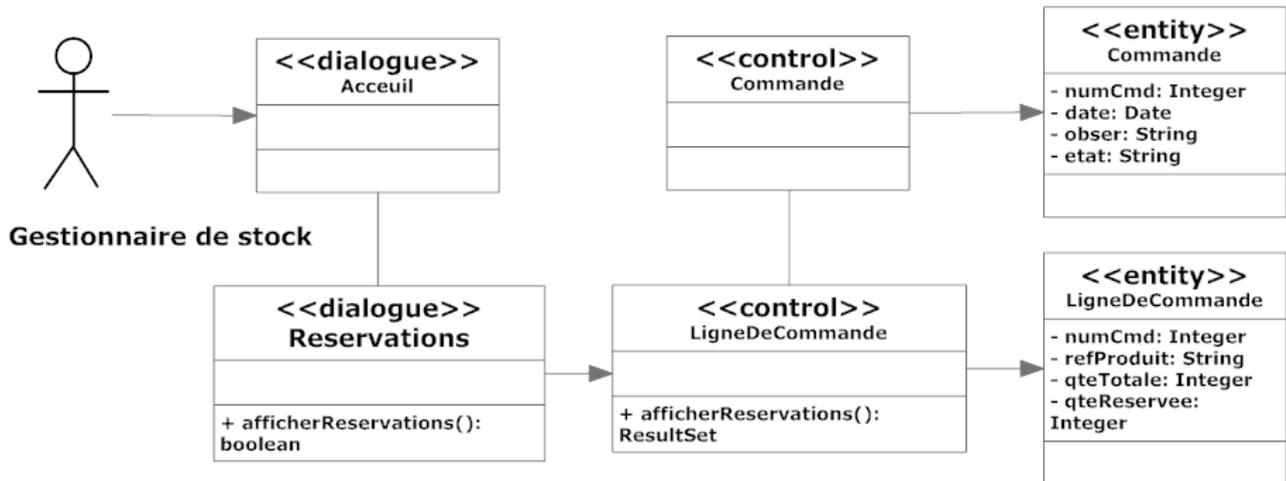


Diagramme de classes participantes 3

■ Diagramme 4 :

Cas d'utilisation concernés :

- approvisionner produit
- consulter mouvements
- ajouter/modifier produit

Classes de dialogue :

- Approvisionnement : la méthode approvisionnement() fait appel à la méthode approvisionner() de la classe de contrôle et lui donne en paramètres les informations entrées dans le formulaire de la page pour augmenter la quantité d'un produit donné.
- HistoriqueStock : la méthode consulterMouvements() permet de voir les opérations du stock à une date donnée en paramètre. afficherBilan() permet de retourner l'état du stock à la date entrée en paramètre par rapport à l'année précédente pour une cuisine donnée.
- EditionProduit : rechercherProduit() affiche les informations d'un produit dans le formulaire présenté par cette classe de dialogue en faisant appel de la méthode de recherche rapide de la classe de contrôle ProduitLecture, de plus la méthode n'est disponible que dans le cas de la modification d'un produit. modifierProduit() enregistre les modifications et ajouterProduit() enregistre un nouveau produit dans la base.

Classes de contrôle :

- ProduitStock : approvisionner() et livrer() augmente et diminue respectivement la quantité d'un produit donné dans une cuisine donnée.
- Mouvements : effectue les opérations de lecture et d'écriture sur les tables Approvisionnement et Livraisons selon qu'on a augmenté ou diminué le stock afin de garder une trace de ces opérations.
- ProduitEcriture : regroupe les méthodes en écriture sur la table Produit
- ProduitLecture : sa méthode rechercheRapide() sera utilisée pour l'édition d'un produit

Classes du modèle :

- ProduitStock
- Approvisionnements
- Livraisons
- Produit

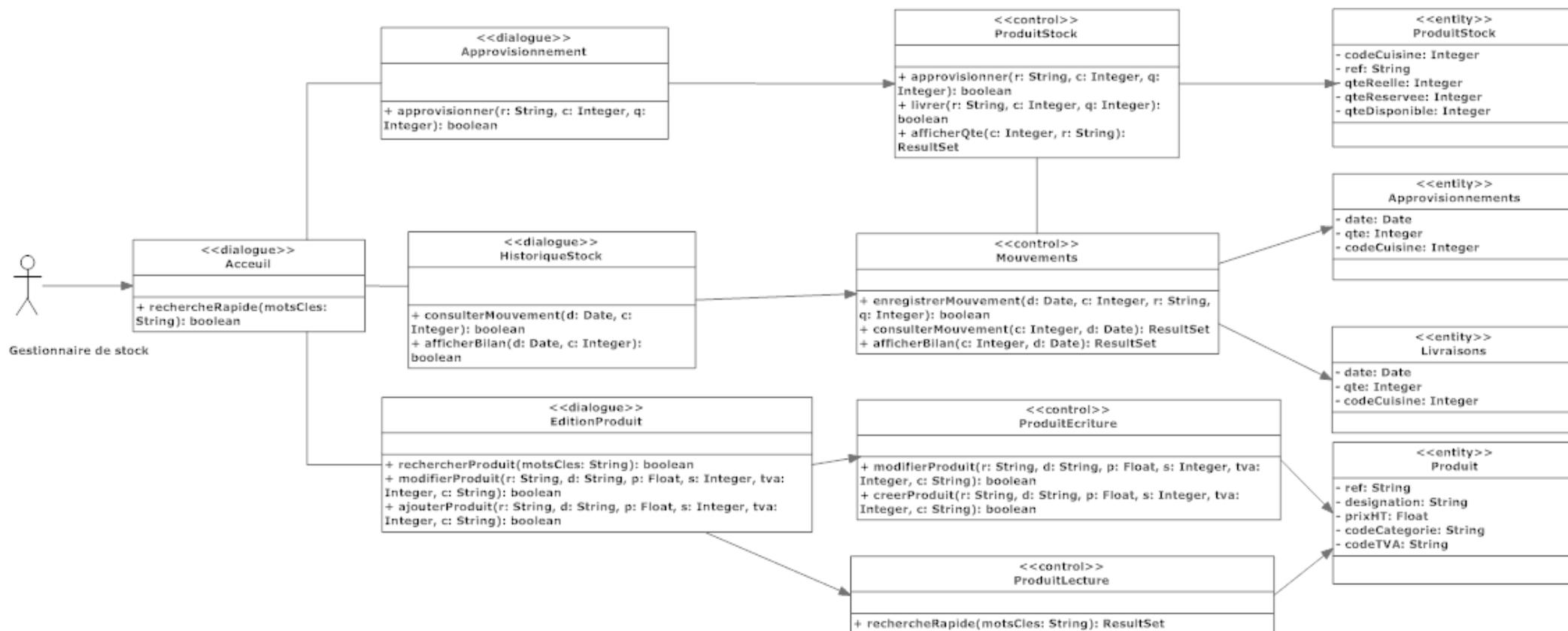


Diagramme de classes participantes 4

▪ Diagramme 5 :

Cas d'utilisation concernés : gérer les catégories (ajouter/éditer une catégorie)

Classes dialogues :

- FenetreCategorie : affiche la liste des catégories d'articles et leur description. A partir de cette page, on peut ajouter ou éditer les catégories.

Classes de contrôle :

- Categorie : regroupe les méthodes de lecture et écriture sur la table Categorie.

Classes du modèle :

- Categorie.