

Diplôme Universitaire de Technologie
Filière : Génie Informatique
Rapport de Projet Java

Sous le thème



**Gestion de commandes d'un
restaurant**

Réalisé par :

Omaima DRHOURHI

Nouhaila ELHALAOUI

Mariam BEN MBAREK

Encadré par :

M. BADDI Youssef

Remerciements

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet et à l'élaboration de ce rapport.

Nous remercions tout d'abord notre encadrant pédagogique, Monsieur BADDI Youssef, pour son accompagnement, ses orientations. Ses conseils et son expertise ont été essentiels à l'avancement du projet.

Enfin, nous exprimons notre reconnaissance à nos familles et à nos proches, dont le soutien moral, la patience et l'encouragement ont été d'un grand apport tout au long de notre parcours.

Résumé

Le projet consiste à développer une application Java de gestion des commandes pour un restaurant, fonctionnant en mode console. Elle permet aux clients de consulter le menu, passer leurs commandes et visualiser le total de la facture, tandis que l'administrateur peut gérer les plats, les catégories et suivre les commandes en cours. Le système vise à automatiser et simplifier la gestion des commandes, tout en mettant en pratique les concepts fondamentaux de la programmation orientée objet, tels que les classes, les objets, l'héritage et l'encapsulation. L'utilisation des collections Java facilite le stockage et la manipulation des données, comme les plats, les commandes et les catégories, et les fichiers texte assurent la persistance des informations.

Le projet répond à des besoins précis pour les clients, qui peuvent consulter le menu, choisir des plats, ajouter des articles à leur commande et générer la facture. Pour l'administrateur, il permet d'ajouter, modifier ou supprimer des plats et des catégories, de gérer les prix et de sauvegarder le menu. La conception repose sur une architecture orientée objet, illustrée par le diagramme de classes et le diagramme de cas d'utilisation, qui permettent de visualiser les entités du système et leurs interactions.

Table des matières

Remerciements	2
Résumé.....	3
Introduction	5
Chapitre 1 :Analyse du besoin Fonctionnalité	6
 Introduction :	6
 1.Analyse de besoin :	6
 1.1Objectifs du projet :	6
 1.2 Utilisateurs du système :	7
 2.Cahier des charges (Fonctionnalités) :	8
 2.1 Fonctionnalités client :	8
 2.2 Fonctionnalités administrateur :	8
 3.Environnement de développement :	9
 4.Exécution du programme :	10
Chapitre 2 : Conception	12
 Introduction :	12
 1. Diagramme de classe :	12
 2. Diagramme du cas d'utilisation :	13
 Conclusion :	14

Introduction

La gestion des commandes dans un restaurant constitue un aspect fondamental pour garantir le bon fonctionnement de l'établissement. Elle permet de suivre les commandes des clients de manière efficace, d'assurer la mise à jour constante du menu et de garantir une facturation exacte et rapide. Une organisation rigoureuse dans la gestion des commandes contribue à éviter les erreurs, à améliorer la satisfaction des clients et à faciliter le travail du personnel.

Dans ce contexte, le projet présenté vise le développement d'une application Java de gestion des commandes pour un restaurant, fonctionnant en mode console. Cette application offre aux clients la possibilité de consulter le menu, de passer leurs commandes et de visualiser le total de la facture. Parallèlement, elle permet à l'administrateur de gérer les plats disponibles, d'organiser les différentes catégories de plats et de suivre les commandes en cours, assurant ainsi une gestion complète et centralisée du restaurant.

L'objectif principal de ce projet est de mettre en pratique les concepts fondamentaux de la programmation orientée objet (POO) en Java. Il s'agit notamment de concevoir et d'utiliser des classes et des objets pour modéliser les éléments du système, de gérer les interactions entre ces objets et d'organiser le code de manière claire et modulaire. Ce projet combine ainsi un objectif pratique, en facilitant la gestion des commandes dans un restaurant, et un objectif pédagogique, en consolidant les compétences en Java orienté objet.

Chapitre 1 :Analyse du besoin Fonctionnalité

Introduction :

La gestion des commandes est essentielle au bon fonctionnement d'un restaurant, car elle permet de suivre les commandes, d'organiser le menu et d'assurer une facturation correcte. Le projet consiste à développer une application Java en mode console qui automatise ces tâches. Les clients peuvent consulter le menu, passer leurs commandes et obtenir la facture, tandis que l'administrateur gère les plats, les catégories et les commandes. Ce projet met en pratique les concepts de programmation orientée objet en Java et fournit une base solide pour la conception et le développement du système, depuis l'authentification jusqu'à l'affichage de la facture finale.

1.Analyse de besoin :

1.1Objectifs du projet :

Le projet Application de Gestion de Restauration vise à faciliter la gestion des commandes et du menu dans un restaurant, en automatisant certaines tâches et en offrant une interface simple pour l'utilisateur. Les principaux objectifs sont :

- **Gérer les catégories de plats :**
 - Permettre la création, la modification et la suppression de catégories telles que entrées, plats principaux, desserts, boissons, etc.
 - Assurer une organisation claire du menu pour faciliter la navigation du client.
- **Gérer les plats et leurs prix :**
 - Ajouter de nouveaux plats avec leur description et leur prix.
 - Modifier les informations d'un plat existant (nom, description, prix).
 - Supprimer des plats obsolètes ou indisponibles.
 - Garantir la cohérence des informations affichées aux clients.
- **Permettre aux clients de passer des commandes :**
 - Les clients peuvent sélectionner plusieurs plats et préciser les quantités désirées.
 - Possibilité de consulter le menu avant de commander.
 - Le système doit vérifier la disponibilité des plats et confirmer la commande.
- **Calculer automatiquement la facture :**

- Générer un total pour la commande passée, incluant éventuellement les taxes ou remises.
- Fournir un récapitulatif clair des plats commandés et des montants correspondants.
- **Offrir un mode administrateur pour gérer le menu :**
 - L'administrateur peut ajouter, modifier ou supprimer des plats et des catégories.
 - Gestion des prix et des détails des plats de manière sécurisée.
 - Permettre un contrôle complet du menu sans intervention du client.
- **Interagir avec l'utilisateur via le terminal :**
 - L'application fonctionne en mode console pour simplifier l'accès et éviter une interface graphique complexe.
 - Fournir des messages clairs, des menus interactifs et des instructions simples pour l'utilisateur.

1.2 Utilisateurs du système :

Le système cible deux types d'utilisateurs principaux :

- **Administrateur :**
 - Responsable de la gestion complète du menu et des plats.
 - Ses actions incluent :
 - ❖ Ajouter de nouvelles catégories ou plats.
 - ❖ Modifier les informations existantes (nom, description, prix).
 - ❖ Supprimer des plats ou des catégories obsolètes.
 - Garant de la cohérence et de l'exactitude des informations affichées aux clients.
- **Client :**
 - Consommateur final utilisant l'application pour passer une commande.
 - Ses actions incluent :
 - ❖ Consulter le menu complet avec les catégories et les prix.
 - ❖ Sélectionner les plats souhaités et préciser les quantités.
 - ❖ Recevoir un récapitulatif de sa commande et la facture correspondante.

2.Cahier des charges (Fonctionnalités) :

Le cahier des charges définit les fonctionnalités attendues du système pour répondre aux besoins des utilisateurs, qu'ils soient clients ou administrateurs.

2.1 Fonctionnalités client :

Les clients utilisent l'application principalement pour consulter le menu et passer des commandes. Les fonctionnalités qui leur sont destinées incluent :

- **Afficher les catégories de plats :**
 - Permettre au client de voir les différentes catégories disponibles (entrées, plats principaux, desserts, boissons, etc.).
 - Faciliter la navigation dans le menu pour choisir rapidement ce qu'il souhaite commander.
- **Afficher les plats et leurs prix :**
 - Pour chaque catégorie, afficher la liste des plats disponibles avec leur description et leur prix.
 - Aider le client à comparer les options et à prendre une décision éclairée.
- **Ajouter des plats à la commande :**
 - Permettre au client de sélectionner plusieurs plats et de spécifier la quantité souhaitée pour chaque plat.
 - Ajouter automatiquement les plats choisis à la commande en cours.
- **Afficher la facture finale :**
 - Calculer automatiquement le total de la commande.
 - Afficher un récapitulatif détaillé des plats commandés, des quantités et des prix.
 - Offrir une vue claire de la facture avant de quitter l'application ou de confirmer la commande.
- **Quitter l'application :**
 - Permettre au client de fermer l'application en toute sécurité après avoir consulté le menu ou passé une commande.
 - S'assurer que toutes les données de la commande en cours sont sauvegardées ou annulées selon le cas.

2.2 Fonctionnalités administrateur :

L'administrateur a un rôle **de gestion complète du menu**, avec des droits spéciaux pour modifier le contenu et les prix. Les fonctionnalités qui lui sont destinées incluent :

Connexion par login et mot de passe :

- Protéger l'accès aux fonctionnalités d'administration.
- Garantir que seules les personnes autorisées peuvent modifier le menu et les prix.

Ajouter un plat (fixe ou dynamique) :

- Permettre l'ajout de nouveaux plats dans une catégorie existante ou créer une nouvelle catégorie si nécessaire.
- Les plats peuvent avoir un prix fixe ou être modulables selon certaines règles (exemple : menu du jour ou promotion).

Supprimer un plat :

- Retirer les plats obsolètes ou indisponibles du menu.
- Mettre à jour automatiquement le menu affiché aux clients.

Modifier le prix d'un plat :

- Ajuster les prix des plats selon les besoins (promotion, variation du coût des ingrédients, etc.).
- Assurer que les modifications soient immédiatement reflétées pour le client.

Supprimer une catégorie :

- Supprimer une catégorie entière lorsqu'elle n'est plus nécessaire (exemple : catégorie temporaire).
- S'assurer que tous les plats associés à cette catégorie sont également supprimés ou transférés.

Nettoyer une catégorie :

- Supprimer uniquement les plats d'une catégorie, sans supprimer la catégorie elle-même.
- Permettre de renouveler rapidement le menu pour une catégorie spécifique.

Sauvegarder le menu dans un fichier texte :

- Conserver une copie du menu actuel pour consultation future ou pour restauration après une modification.
- Permettre de garder un historique des menus pour une meilleure gestion administrative.

3. Environnement de développement :

Pour réaliser ce projet, un environnement de développement adapté a été mis en place afin d'assurer le bon fonctionnement de l'application et la compatibilité avec les fonctionnalités Java orientées objet.

- **Langage de programmation : Java**
Le projet a été développé en **Java**, un langage orienté objet robuste et largement utilisé pour la création d'applications console, de gestion et d'interfaces graphiques. Java permet de gérer efficacement les objets, les collections et les fichiers, ce qui est essentiel pour ce type de projet.
- **IDE : Visual Studio Code**
L'IDE choisi est **Visual Studio Code**, un éditeur léger et polyvalent, qui offre des extensions Java pour faciliter la rédaction, la compilation et le débogage du code. Il permet également de gérer facilement les projets et les fichiers source.
- **Mode d'exécution : Terminal (Console) :**
L'application fonctionne en mode **console**, ce qui permet d'interagir directement avec l'utilisateur via le terminal. Ce mode est simple à mettre en œuvre et facilite la lecture des entrées et sorties standard.
- **JDK : Java JDK 8 :**
Le projet nécessite **Java Developpement Kit (JDK) version 8**. Le JDK fournit les outils nécessaires pour compiler et exécuter les programmes Java, ainsi que les bibliothèques standard pour la manipulation des collections, des fichiers et des entrées/sorties.
- **Bibliothèques utilisées :**
Plusieurs bibliothèques standard de Java ont été utilisées :
 - **java.util** : pour la gestion des collections dynamiques comme les ArrayList, permettant de stocker et manipuler les catégories de plats, les commandes et les éléments de menu.
 - **java.io** : pour la lecture et l'écriture des fichiers, ce qui permet de sauvegarder le menu et les commandes dans des fichiers texte, garantissant la persistance des données entre les exécutions de l'application.

4.Exécution du programme :

Lors de l'exécution du programme :

- ❖ L'utilisateur est invité à se connecter (admin ou client)

```
Username: client
Password: 1234

[Mode client]
```

- ❖ Le menu principal s'affiche avec les catégories disponibles :

```
===== MENU =====
1. PLATS MAROCAINS
2. PLATS ITALIENS
3. PLATS FRANÇAIS
4. PLATS LIBANAIS
5. BOISSONS
6. DESSERTS
0. Facture & Quitter
Choix: [
```

- ❖ Le client peut choisir des plats et les ajouter à sa commande :

```
Choix: 1

--- PLATS MAROCAINS ---
1. Couscous Royal - 70.0 DH
2. Tajine Poulet - 55.0 DH
3. Tajine Viande - 70.0 DH
4. Pastilla Poulet - 65.0 DH
5. Harira - 15.0 DH
6. Rfissa - 60.0 DH
0. Retour
Choix: [
```

- ❖ L'administrateur peut accéder au menu de gestion :

```
PS C:\xampp\htdocs\Gestion-Restaurant> & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.16.8-hotspot\bin\optionMessages' '-cp' 'C:\Users\pc\AppData\Roaming\Code\User\workspaceStorage\846f419e4f9efd895196b0cc455\aurant_449f160f\bin' 'AppConsole'
Username: admin
Password: 1234

[Admin connecté]
```

```
--- GESTION DU MENU ---
1. Ajouter plat (fixe)
2. Ajouter plat dynamique
3. Supprimer plat
4. Modifier prix
5. Sauvegarder menu
6. Supprimer catégorie
7. Nettoyer catégorie
0. Retour
Choix: [
```

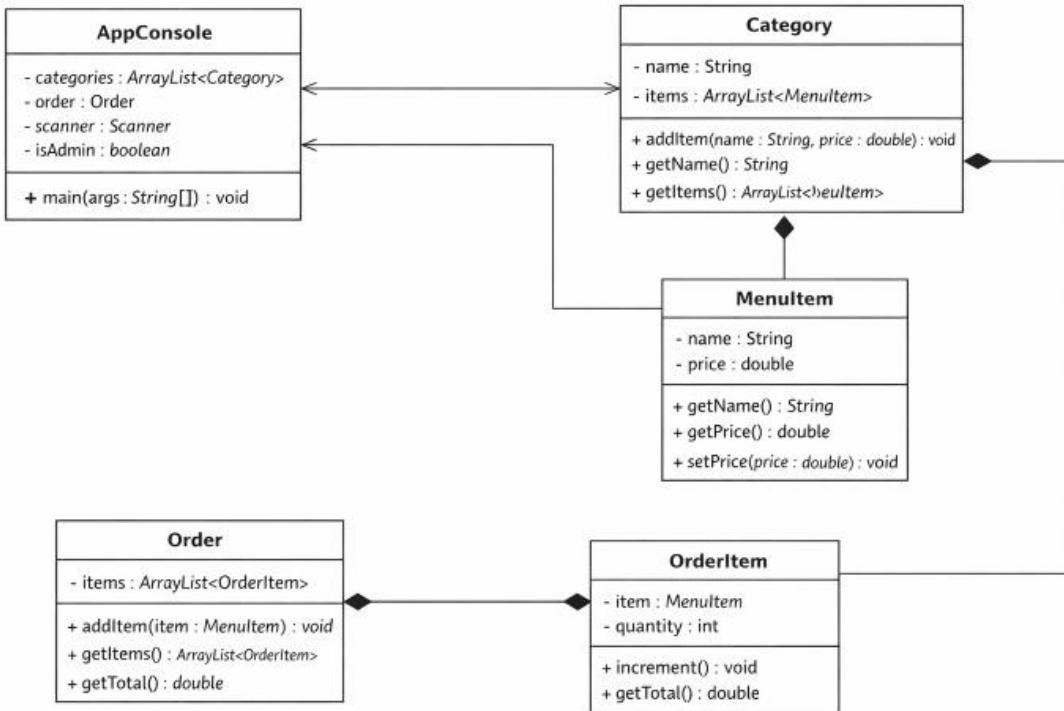
Chapitre 2 : Conception

Introduction :

Pour notre projet de gestion de commande d'un restaurant, la conception s'appuie sur une architecture orientée objet, permettant de modéliser les différents éléments du système sous forme de classes et de relations. Cette approche favorise la réutilisabilité du code, la maintenance et l'évolution future de l'application. Dans ce chapitre, nous présentons le diagramme de classes et le diagramme de cas d'utilisation

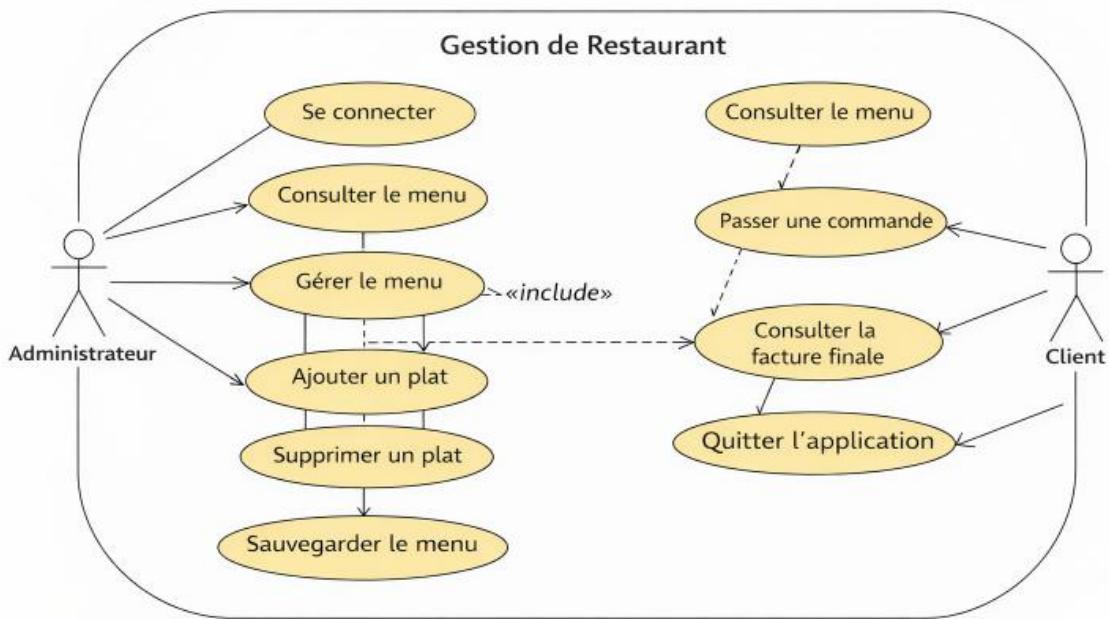
1. Diagramme de classe :

Le projet est basé sur une architecture orientée objet, composée de plusieurs classes principales :



2. Diagramme du cas d'utilisation :

Le diagramme de cas d'utilisation ci-dessous représente les principales interactions entre les utilisateurs (administrateur et client) et les fonctionnalités de l'application de gestion de commande d'un restaurant :



Conclusion :

Ce projet a permis de concevoir et de développer une application complète de gestion de restauration en Java, couvrant à la fois les aspects fonctionnels et techniques. Grâce à ce travail, nous avons pu modéliser un système réaliste où les clients peuvent passer des commandes et où l'administrateur peut gérer les plats, tout en assurant un suivi précis des factures et des opérations.

La réalisation de ce projet a également permis de renforcer la compréhension et la maîtrise des concepts fondamentaux de la programmation orientée objet, tels que les classes, les objets, l'héritage et l'encapsulation. L'utilisation des collections Java a facilité le stockage, la manipulation et la recherche des données, notamment pour la gestion des menus, des commandes et des clients.

En outre, le projet a mis en lumière l'importance de l'interaction avec l'utilisateur via le terminal, en mettant en place des menus clairs et des messages informatifs, afin d'assurer une expérience utilisateur intuitive et efficace.

Enfin, ce projet constitue une base solide pour de futurs développements, comme l'ajout d'une interface graphique ou la connexion à une base de données, et illustre comment une application bien structurée peut répondre aux besoins d'un restaurant de manière organisée et professionnelle.