Université Euro Méditerranéenne Fès
Ecole d'Ingénierie Digitale et d'Intelligence Artificielle (EIDIA)

Projet de fin de semestre

Module : Application Répartie

Application RPC de gestion des commandes d'un restaurant

2023/2024

Réalisé par :
Wissal BOUTAYEB

Encadré Par :
Mr. AMAMOU Ahmed

Table des Matières

Introduction et Présentation du Projet

Contexte

Le projet de gestion des commandes d'un restaurant a été réalisé dans le cadre de notre cursus universitaire, au sein du module de programmation avancée. Le but de ce projet est de mettre en pratique les concepts de développement d'applications distribuées et d'interfaces graphiques, tout en répondant à une problématique concrète et courante dans l'industrie de la restauration. Le projet a été entrepris dans un environnement de développement Java, utilisant Java RMI (**Remote Method Invocation**) pour la communication entre le serveur et les clients, et Swing pour la création de l'interface utilisateur graphique.

Objectifs

Développer une application distribuée :Créer un système client-serveur où le serveur gère le menu du restaurant et les commandes des clients, et où les clients peuvent consulter le menu, passer des commandes, et obtenir le total de leur facture.

Implémenter une interface utilisateur intuitive : Concevoir une interface graphique conviviale et facile à utiliser pour les clients, permettant une interaction fluide avec le système.

Assurer la fiabilité et la sécurité des transactions : Garantir que les commandes sont correctement enregistrées et que les calculs de factures sont précis, tout en assurant une communication sécurisée entre le client et le serveur.

Portée

La portée du projet inclut les éléments suivants :

Consultation du Menu : Les clients peuvent voir la liste des plats disponibles avec leurs prix.

Passation de Commandes : Les clients peuvent sélectionner des plats ajouter au liste des plats et passer des commandes au serveur.

Calcul de la Facture: Le système (le serveur) calcule le total de la facture pour chaque client en fonction des plats commandés.

Gestion des Clients : Le serveur enregistre les commandes de chaque client de manière distincte dans une interface.

Problématique

Prise de Commande Inefficace : Dans de nombreux restaurants, les serveurs prennent les commandes à la main ou via des méthodes non centralisées, ce qui peut entraîner des erreurs de transcription, des pertes de tickets de commande, et des retards dans la transmission des informations à la cuisine

Erreurs dans les Commandes : Les erreurs humaines sont fréquentes lorsqu'il s'agit de noter les commandes des clients. Cela peut inclure des erreurs de transcription des plats, des oublis de certains éléments, ou des confusions entre différentes commandes.

Calcul des Factures : Le calcul manuel des factures peut entraîner des erreurs de calcul, des incohérences, et une perte de temps. Une gestion inefficace des factures peut également mener à des pertes financières pour le restaurant.

Manque de Centralisation : L'absence d'un système centralisé rend difficile la gestion des données des clients et des commandes. Il est compliqué d'analyser les tendances des commandes, d'optimiser les menus, ou de fidéliser les clients sans une base de données centralisée.

Cahier des Charges

- **Connexion des Clients :** Les clients doivent pouvoir se connecter à l'application en entrant leur nom.
  L'application doit valider l'entrée du nom du client avant de permettre l'accès aux autres fonctionnalités.
  Une fois connecté, le client peut voir les dernières nouveautés du restaurant.
  consulter le menu et passer des commandes.
  Choisir son plat désiré depuis le menu
- **Consultation du Menu :** Les clients doivent pouvoir voir une liste de tous les plats disponibles avec leurs descriptions et prix.
  L'interface doit permettre un défilement facile et une lecture claire du menu.
- **Prise de Commande:** Les clients doivent pouvoir ajouter des plats à leur commande en entrant le nom du plat.
  L'application doit vérifier que le plat existe dans le menu avant de l'ajouter à la commande.
- **Calcul de la Facture** : Une fois la commande passée, l'application doit permettre de calculer le total de la facture
  La facture doit inclure le coût de tous les plats commandés et afficher le total de manière claire.
  L'application doit récupérer et afficher le montant total de la facture depuis le serveur

Conception et Architecture

L'architecture de l'application de gestion des commandes d'un restaurant est basée sur un modèle **client-serveur** utilisant Java RMI (Remote Method Invocation)

Qu'est ce que Java RMI (Remote Method Invocation) :

Java RMI (Remote Method Invocation) est une API fournie par Java qui permet à un programme Java d'invoquer des méthodes sur un objet distant (situé sur une autre machine virtuelle Java, potentiellement sur un autre ordinateur). Cette technologie facilite la communication entre programmes Java répartis sur plusieurs systèmes en réseau.

Client :

Interface graphique utilisateur (GUI) développée avec **Java Swing** pour permettre aux clients de se connecter, consulter le menu, passer des commandes et calculer leur facture.

Communique avec le serveur via des appels RMI pour obtenir le menu, passer des commandes et calculer la facture.

Serveur :

Fournit des services RMI pour la consultation du menu, la prise de commande et le calcul de la facture.

Gère les données des plats disponibles (menu) et les commandes des clients.

Met à jour l'interface serveur pour afficher les commandes reçues en temps réel.

Interface :

Interface définissant les méthodes RMI pour les services du restaurant. Cette interface est implémentée par le serveur et utilisée par le client pour invoquer des méthodes distantes.

Diagrammes UML :

Diagramme de séquence :

Le Client entre dans le restaurant et consulte le Menu :Le client lance l'application et demande à consulter le menu disponible. Cette action envoie une requête au serveur pour obtenir la liste des plats disponibles.

Le client décide de commander un plat et envoie sa demande au Serveur :(Passer_une_commande) :Une fois que le client a choisi un plat, il passe une commande en envoyant les détails de cette commande au serveur via une communication RMI (**Remote Method Invocation**).

Le Serveur reçoit la demande et vérifie la disponibilité de la commande dans le stock :Le serveur vérifie si les plats commandés sont disponibles dans le stock du restaurant.

Préparation de la commande :Si les plats sont disponibles, le serveur commence à préparer la commande.

Attente de la préparation de la commande :

Il y a une période d'attente pendant laquelle la commande est en cours de préparation.

Le Serveur envoie la commande au client :Une fois la préparation terminée, le serveur notifie le client que sa commande est prête et la lui envoie.

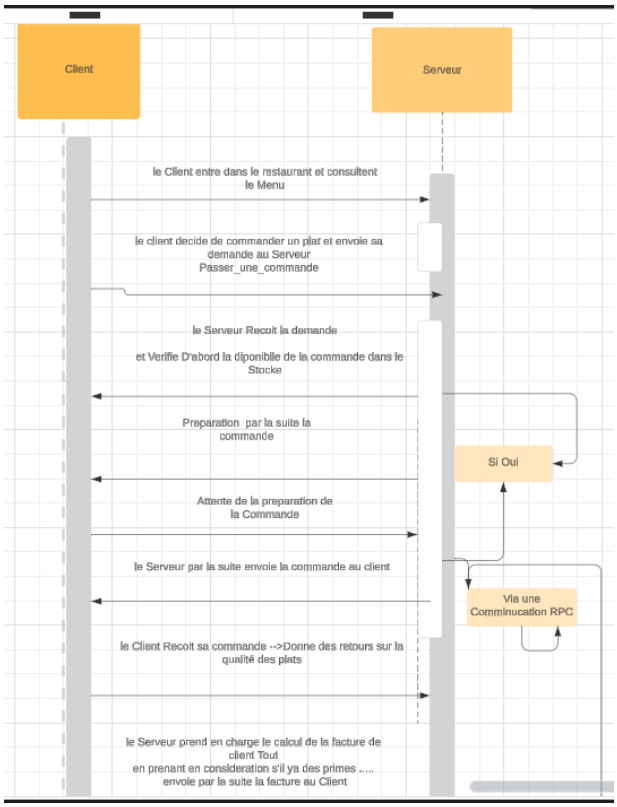
Le Client reçoit sa commande et donne des retours sur la qualité des plats :Le client reçoit la commande et peut donner des retours ou des avis sur la qualité des plats reçus.

Le Serveur prend en charge le calcul de la facture du client :Le serveur calcule la facture totale en fonction des plats commandés et des éventuelles primes ou réductions applicables.

Envoi de la facture au Client :La facture est envoyée au client via l'application.

Le Serveur enregistre la livraison de la commande dans le système :Le serveur enregistre la livraison de la commande comme complétée dans son système.

Si la commande n'est pas disponible :Si le serveur constate que certains plats ne sont pas disponibles, il envoie un message au client pour l'informer de cette indisponibilité et lui propose éventuellement d'autres plats de qualité similaire.



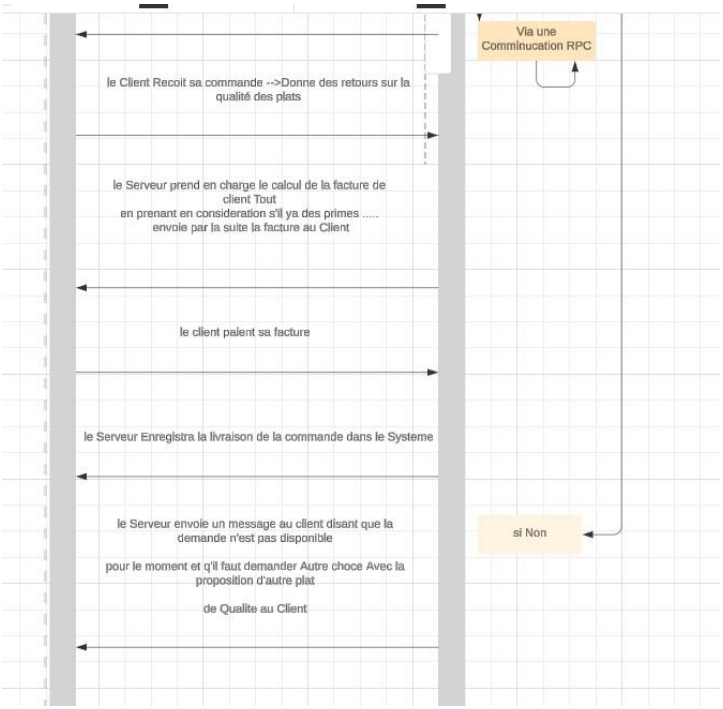
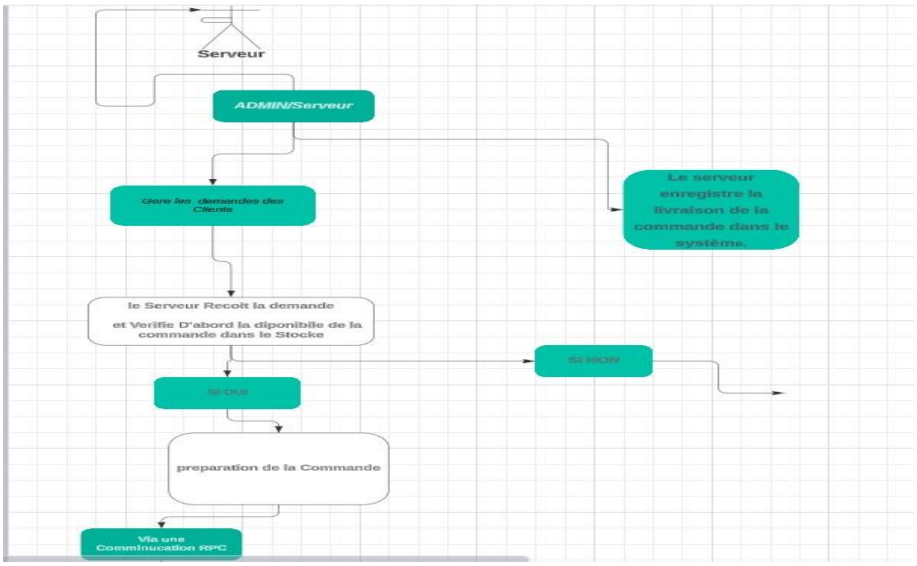
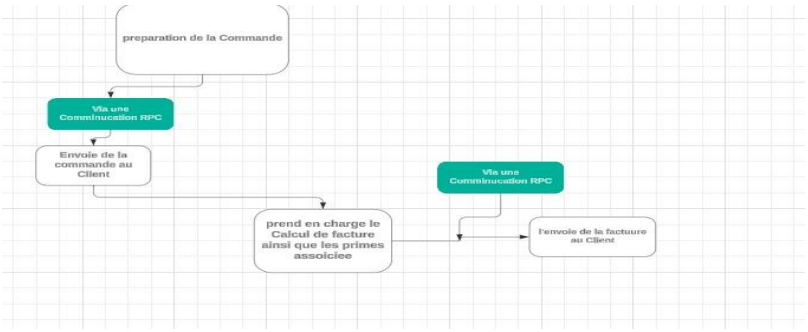


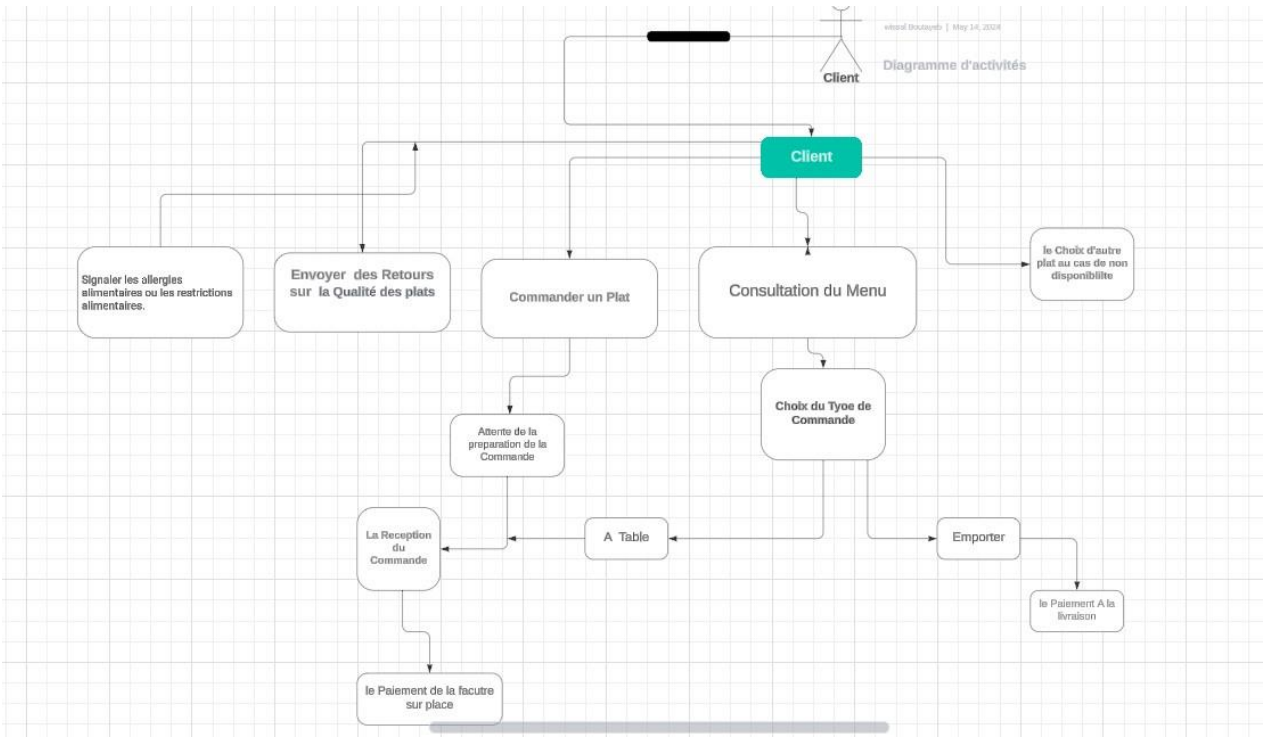
Diagramme d'activité :

Serveur :





Client :



Technologies Utilisées

Java : Langage de programmation utilisé pour développer à la fois le client et le serveur de l'application.

Java RMI (Remote Method Invocation) : Permet la communication entre le client et le serveur en invoquant des méthodes à distance. Utilisé pour transmettre les données des commandes et des menus entre le client et le serveur.

Java Swing : Bibliothèque Java pour créer des interfaces graphiques riches et interactives.

Parmi les methodes inclut :

JFrame: Pour la fenêtre principale

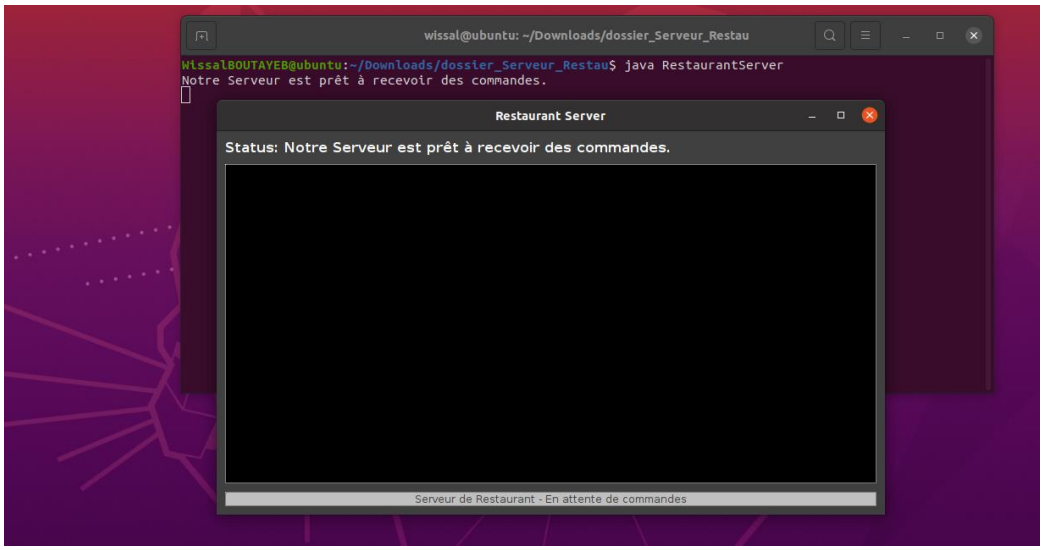
JPanel: Pour organiser les composants.

JTextArea: Pour afficher du texte multi-lignes.

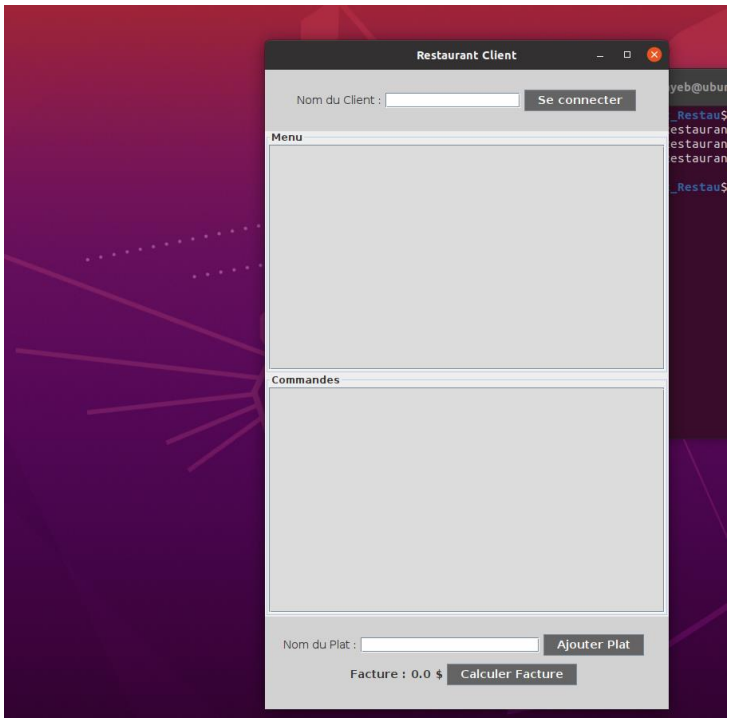
JButton: Pour les boutons d'action.(……)

Développement du Projet

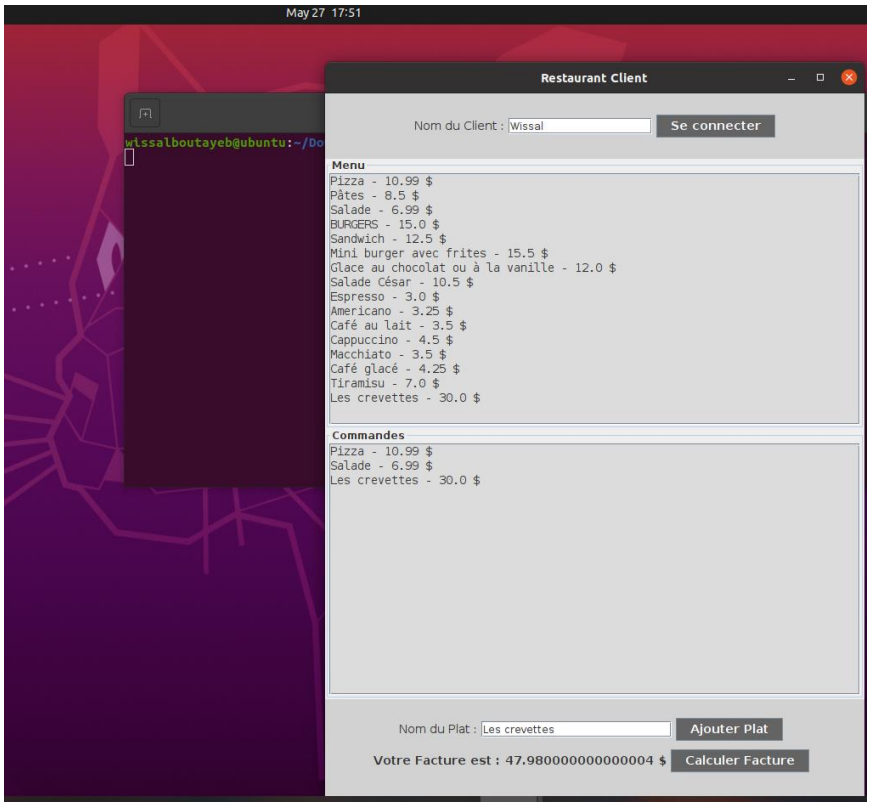
Interface du Serveur :



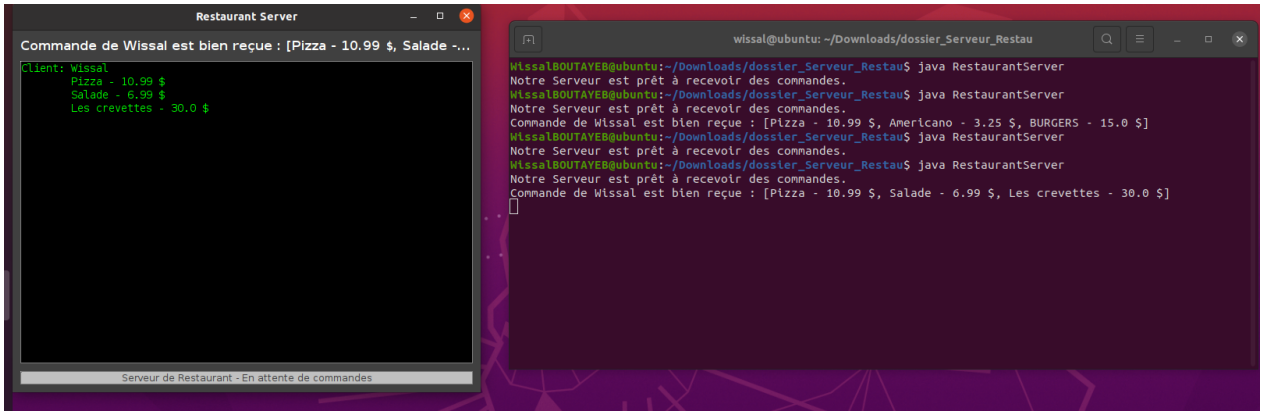
Interface du Client :



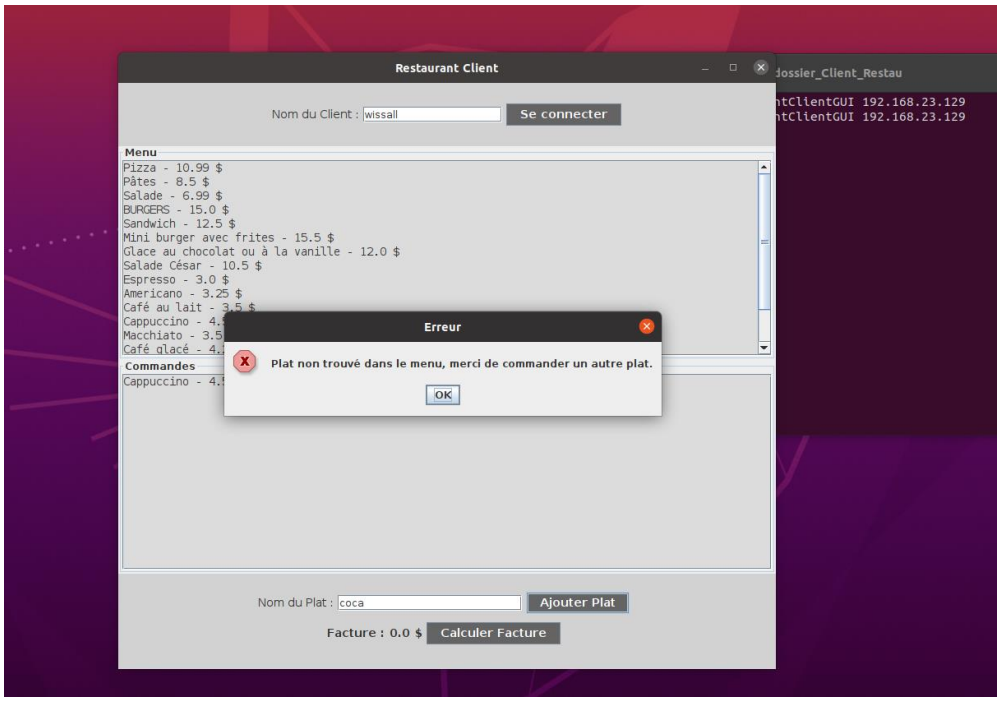
- <u>L'Etude de cas lorsqu'un client commande un plat :</u>



Le serveur enregistre la commande de client :



Le cas de la demande de client d'un plat qui ne se trouve pas dans le menu :



Problèmes Rencontrés et Solutions Apportées :

Problèmes Techniques :

1- **Problèmes de Connexion RMI :**

Problème : Nous avons rencontré des difficultés lors de l'établissement de la connexion RMI entre le client et le serveur. Le client n'arrivait pas à localiser le service distant, ce qui entraînait des erreurs de NamingException.

Solution : Après analyse, nous avons découvert que le problème provenait de la configuration incorrecte du registre RMI et de l'URL utilisée pour la recherche du service. Nous avons corrigé l'URL en veillant à inclure l'adresse IP et le port corrects du serveur RMI. De plus, nous avons ajouté un délai de démarrage pour le registre RMI afin de garantir qu'il était opérationnel avant que le client tente de se connecter.

2- **Erreurs de Synchronisation** :

Problème : Lorsqu'il y avait plusieurs clients accédant au service en même temps, nous avons observé des incohérences dans les données des commandes, telles que des doublons ou des commandes manquantes.

Solution : Nous avons implémenté des mécanismes de synchronisation au niveau du serveur en utilisant des blocs synchronisés (**synchronized**) et des collections thread-safe comme **ConcurrentHashMap** pour gérer les commandes et les clients. Cela a permis de garantir l'intégrité des données lors de l'accès concurrent.

Conclusion

Le projet de gestion des commandes d'un restaurant a abouti à la création d'une application robuste et performante utilisant Java, RMI et Swing. En répondant efficacement aux besoins de consultation du menu, de prise de commandes et de calcul des factures, l'application a su intégrer des fonctionnalités essentielles tout en garantissant une performance optimale et une expérience utilisateur intuitive. Les défis techniques, tels que les problèmes de connexion RMI et de synchronisation, ont été résolus avec succès, renforçant ainsi la fiabilité et la robustesse du système. Ce projet démontre l'efficacité de l'architecture client-serveur et l'importance d'une conception soignée dans le développement de solutions logicielles.