

Problème (MOM)

Une pizzeria désire informatiser son système de commande – production – livraison afin d'améliorer son rendement. Voici comment elle aimerait que le processus se déroule une fois informatiser.

Lorsqu'un client téléphone au restaurant, le commis qui prend l'appel commence toujours par demander au client s'il en est à sa première commande dans cette entreprise. Si le client a déjà commandé, le commis lui demande son numéro de téléphone afin de rechercher les coordonnées du client et de confirmer son adresse. Lorsque le client en est à sa première commande, le commis doit alors saisir les coordonnées complètes du client (nom, prénom, adresse, numéro de téléphone, et la date de sa première commande). Il saisit ensuite la commande.

Une commande comporte obligatoirement une (ou plusieurs) pizza(s), et éventuellement des produits annexes de **type boissons (coca, jus d'orange etc.)** selon un volume à préciser. Les pizzas sont disponibles en trois tailles différentes (petite, moyenne, grande) et en plusieurs types (par exemple sauce tomate/fromage, végétariennes, toutes garnies, etc.). Le prix d'une pizza dépend de sa taille et de son type.

La commande comporte les informations suivantes : no de commande, heure, date, nom du client, **nom** du commis, items commandés. La ou les pizza(s) commandée(s) est (sont) ensuite préparée(s) par les cuisiniers pour ensuite être transmise(s) à la livraison.

Une fois la commande est passée, un message de confirmation est envoyé à :

- client. Message de prise en charge de sa commande ;
- cuisine. Message de confirmant le choix de pizza ;
- livreur. Message de command avec les cordonnées du client avec un délais de 5 min
- commis. Message de confirmation de l'ouverture de commande.

Le livreur prend alors la ou les pizza(s) et la facture afférente, y ajoute les produits annexes et part effectuer la livraison. Lorsque l'adresse indiquée est correcte et trouvée, il remet la commande au client et reçoit le paiement et il envoie un message de confirmation à la pizzeria. Il retourne alors à la pizzeria avec le double de la facture et l'argent.

À son retour, le commis demande la facture au livreur, saisit le numéro de commande, le numéro du livreur, encaisse l'argent et ferme la commande en envoyant deux messages à

- livreur. Message de fermeture de la commande et le montant de son paiement.
- Commis. Message de confirmation de la fermeture de la commande.

À tout moment, le commis peut connaître l'état d'une commande (en préparation, en livraison ou fermée). Les commandes qui n'ont pas pu être livrées sont marquées comme des pertes par le commis et sont fermées. Chaque commande honorée permet d'augmenter le cumul des commandes réalisées par le client.

1. Veuillez donner le schéma UML correspondant à votre analyse conceptuelle de ce cas.
2. En proposer une application C# qui simule le fonctionnement de la pizzeria avec la communication.

Livable

Le projet se fait en équipe seul ou à 3 (équipe TPs). Elle doit rendre compte du cahier des charges ci-dessus. Tout information non écrite peut être interprétée comme vous le souhaitez.

Vous joindrez un rapport (**maximum** 3 pages) à votre solution C# dans lequel vous noterez vos ajouts si besoin. Vous y expliquerez le diagramme UML de votre solution et les structures C# qui en découlent.

La solution C# se présente sous forme d'une interface graphique (WPF). Vous êtes libres de représenter les informations comme vous le souhaitez.

Module Client / Effectif

L'outil doit pouvoir permettre d'entrer, supprimer ou modifier un nouveau Client et un nouveau Commis et Livreur d'une part ou bien de lire des fichiers Clients, Commis ou Livreurs

Il faut à tout moment pouvoir afficher l'ensemble des Clients et/ou des effectifs selon plusieurs critères : (successivement ou simultanément) Clients :

- Par ordre alphabétique
- Par ville
- Montant des achats cumulé, ce qui permettra de connaître les meilleurs clients

Module Commandes

Pour faciliter la simulation, il vous est demandé de pouvoir charger des fichiers de commandes.

Il faut, par ailleurs, pouvoir créer une nouvelle commande ou la modifier et simuler ses différentes étapes :

- Commande en préparation, en livraison ou fermée.
- Commande encaissée ou à perte. Il faut pouvoir :
- Calculer le prix d'une commande et l'afficher moyennant son numéro
- Afficher une commande moyennant son numéro

Module Statistiques

Il faut également faire des bilans généraux

- Afficher par commis, le nombre de commandes gérées
- Afficher par livreur le nombre de livraisons effectuées
- Afficher les commandes selon une période de temps
- Afficher la moyenne des prix des commandes
- Afficher la moyenne des comptes clients

Module Communication

Il doit être représenté par 4 parties :

- client,
- cuisine,
- livreur,
- commis.

Module Autre qui sera le résultat de votre créativité sur la gestion de cette pizzeria

Dans tous les cas, il faut utiliser tous les concepts :

- POO avec héritage bien-sûr,
- Classe abstraite,
- Interface (utilisation d'interface C# et création d'interface),
- Délégation (utilisation de délégation C#)
- 1 collection générique différente
- utilisation de Message Broker OU async/await au choix