

Ce TP se déroulera sur deux séances. Le travail est à rendre à la fin de la dernière séance (2eme) au plus tard le samedi à 23h59. Merci de tout envoyer sous format .zip et à nommer comme suit **TPX_NOM_PRENOM**

N'hésitez pas à commenter vos codes !

Merci de respecter le nommage du fichier zip ainsi que la deadline !

Malgré le nombre important de succursales du magasin LeadelMarché, toutes les démarches sont effectuées sous format papier. Vous allez créer un système pour aider ce magasin à automatiser tout son process car les gérants ne sont pas convaincus de la nécessité de passer au format électronique.

Ce scenario est très commun pour les développeurs de Technologies de l'Information, où l'utilisateur refuse de changer sa façon de travailler par crainte de ne pas maîtriser le nouveau système ou de sa complexité.

Votre objectif est de créer un système user friendly qui s'adapte au mieux au besoin du client et qui lui donne envie de changer de méthode de travail.

En utilisant Java, vous allez créer un système CRUD (Create, Read, Update, Delete) pour gérer le magasin. Le système aura plusieurs modules : stock, points de vente, management du personnel et management des clients, le tout connecté à une Base de données (BD).

Il y aura donc un menu principal permettant de sélectionner le module sur lequel l'utilisateur va travailler. Pour les modules inventaire, management de personnel et la BD, il y aura seulement une instance de chacun. Cependant, il y aura plusieurs points de vente.

BD :

- Vous pouvez utiliser des fichiers de texte pour enregistrer vos données. Vous pouvez créer autant de fichiers que vous jugez nécessaires, mais gardez à l'esprit que plus vous en utilisez (si vous ne réfléchissez pas profondément au modèle), plus il est difficile de l'étendre et cela sera source d'erreurs.
- Bonus : Pour ceux qui maîtrisent SQL, il est possible d'utiliser ce langage pour gérer votre base de données.

Stock :

- Permet d'ajouter des nouveaux produits et les gérer par la suite.
- Les données à sauvegarder sont (liste non exhaustive) : nom du produit, description, type de produit (seulement 4 types : aliments, médecines, appareil électronique et autres), prix, pays d'origine, pourcentage de TVA, quantité en stock, nom de fournisseur.
- Pas de code barre dans un premier temps : le caissier(ère) sélectionne le produit et la quantité et le prix sera calculé en fonction, pour les produits à peser, il sélectionnera le type de produit et le poids.

Management du personnel :

- Permet d'ajouter du personnel et de les gérer par la suite.
- Les données à sauvegarder sont (liste non exhaustive) : nom, prénom, adresse personnelle, adresse de lieu de travail, poste/fonction, supérieur immédiat, numéro de badge.
- Chaque personne aura un seul lieu de travail, un seul poste/fonction et un seul supérieur immédiat.

Management des clients :

- Permet d'ajouter des nouveaux clients et les gérer par la suite.
- Les données à sauvegarder sont (liste non exhaustive) : nom, prénom, numéro carte de fidélité, mail, code postal.
- Il doit exister un client « Anonyme » à utiliser pour les clients non-inscrits.

Points de vente :

- Permet de réaliser et enregistrer une vente.
- Les données à sauvegarder sont (liste non exhaustive) : Nom du point de vente, numéro du badge du caissier/ère, numéro carte fidélité du client, les produits vendus et la quantité (avec prix individuel et avec la quantité), sous total, TVA, total.
- La fenêtre d'interaction doit avoir des boutons avec des fonctionnalités pour rendre plus simple le travail du caissier/ère (ex : une petite calculatrice)
- Chaque vente doit réduire le stock de l'inventaire.
- Après chaque vente, un ticket est créé pour le client. Il s'agit d'un document texte qui résume les achats du client et les réductions qu'il a pu avoir à la suite de promotions périodiques ou des promotions applicables avec une carte de fidélité. Ce document peut être imprimé ou envoyé par mail au client.

Considérations/Restrictions :

- Utiliser et respecter tous les principes de l'approche orientée objet (pas de méthode hors d'un objet par exemple)
- Pensez à la scalabilité : votre client vous demande un changement de dernière minute (supprimer un article, réduire la quantité d'un article, changer le mode de paiement, ...)
- Pour la tâche DELETE, l'astuce est de ne pas effacer les données, mais les cacher avec une variable Boolean appelée « active ». Cela permet de ne pas perdre l'information.
- Chaque module doit permettre une recherche par nom, prénom et identifiant (selon le cas). Pour le nom et le prénom, la recherche doit permettre des recherches incomplètes ; ex : Si nous écrivons « em » le système doit afficher « Émilie, Emma, Emmanuelle, etc » ; Si nous écrivons « emma » le système doit afficher « Emma, Emmanuelle ». Il faut ignorer les différences entre accents, majuscules et minuscules.
- Vous devez impérativement, ajouter quelques données pour tester votre système (au moins 10). Ainsi, le client peut commencer à tester votre système. Si vous lui laissez la tâche fastidieuse de commencer à zéro, il ne le testera pas du tout !
- Comme dans tous les systèmes, il est possible que le client final n'exprime pas dans son besoin toutes les exigences et les fonctions dont il aura besoin. Vous êtes donc libre d'ajouter des nouvelles fonctionnalités, que vous jugerez utiles et essentielles pour le client. N'oubliez pas de préciser ces fonctionnalités et leurs justifications dans votre compte rendu.
- Développer avec une interface graphique facile à utiliser et intuitive (pensez à mettre des tooltips et boutons d'aide). Il faut considérer que le système sera utilisé par des personnes sans aucune connaissance en informatique.

Travail demandé :

1. Donner un diagramme de classe modélisant cet énoncé.
2. Réaliser un diagramme de communication décrivant le processus à mettre en place.
3. En se basant sur les modèles des deux questions précédentes, donner une implémentation en Java du processus
4. Ajouter une interface graphique à votre programme en utilisant le modèle MVC

Améliorer votre programme comme suit :

5. Ajouter le traitement des produits par leurs codes-barres
6. Pour les produits alimentaires : Prendre en compte ceux qui se vendent à la pièce et sans code barre (ex : avocats, kiwi, ...). Un prix sera prédéfini par produit et peut augmenter/baisser.
7. Prendre en compte des caisses rapides :
 - a. Les produits peuvent être scannés puis posés sur une balance qui permet de comparer le poids fourni par le code barre au poids réel et valider le produit,
 - b. Il est aussi possible de sélectionner manuellement le produit pour préciser le nombre de pièces (cela s'applique à ceux qui n'ont pas de code barre)
 - c. Il est possible de faire appel à un(e) caissier(ère) si besoin d'aide ou en cas de blocage de la caisse rapide

8. Gestion du personnel : le gestionnaire du personnel peut saisir les congés/RTT de son personnel et l'outil sera capable de lui fournir un emploi de temps optimiser.
 - a. Il faut savoir que le magasin est ouvert du Lundi au samedi de 08h à 20h
 - b. Chaque employé doit travailler au plus 08h par jour avec une heure de pause déjeuner et deux pauses de 15 minutes. La pause déjeuner s'applique uniquement à ceux qui travailleront 7 & 8h par jour. Pour moins de 7h de travail, il y aura une seule pause de 15 minutes.
 - c. Les contrats disponibles sont des CDI (contrat à durée indéterminée) de 35 heures par semaine, des CDI étudiants qui varient entre 7 et 21 heures par semaine, des CDD (contrat à durée déterminée) et des intérimaires qui sont appelés durant les grandes vacances pour remplacer le personnel (période Juillet/Aout) et durant les périodes de soldes.
 - d. Le magasin est ouvert les 2 premiers dimanches des soldes et le personnel est payé 200%
 - e. Le programme doit être capable d'alerter le gestionnaire du personnel en cas de manque d'effectif quelques mois à l'avance afin de pouvoir réserver des ressources intérimaires.
9. Gestion du stock :
 - a. Alerter le gestionnaire quand il y a moins de XX% du produit afin de passer ces commandes.
 - b. Permettre au gestionnaire de consulter le stock des autres points de vente afin de renseigner le client et l'envoyer au magasin le plus proche.
10. Gestion des promotions : comme tout magasin, il est important d'attirer les clients par des promotions régulières. Une promotion a une durée renouvelable, un type (2 achetés 3ème offert, 30% sur le 2ème produit identique, ...), avec une application immédiate ou sur la cagnotte de la carte de fidélité
11. Il est aussi souhaitable que l'outil puisse fournir des statistiques sur la vie du magasin : nombre de carte de fidélité créés sur une période saisie par le gestionnaire, le nombre de produits vendus pour une promotion donnée, le nombre d'absence du personnel.
 - a. Le gestionnaire doit avoir accès à une liste déroulante des statistiques possibles et de choisir la période pour laquelle il souhaite avoir cette information.
 - b. Il peut aussi avoir la possibilité de comparer la même statistique sur deux périodes différentes (soldes A et A-1).
 - c. Il sera aussi possible de créer une nouvelle statistique qui enrichira la liste déroulante
 - d. Les statistiques peuvent être présentées sous plusieurs formes (camembère, barre, tableau, ...) et peuvent être envoyées par mail au gestionnaire afin de les partager avec ces supérieurs.