

Rapport : Projet BDD

Introduction :

Objectif : Créer un Système qui permet d'assurer la gestion des commandes, des clients et des produits en stock.

Conception et développement du système :

Après avoir identifié longuement et soigneusement les **besoins** de Monsieur BelleFleur, nous avons effectué le **diagramme E/A** suivant :

Choix des tables de notre DataBase Fleurs sur MySQL :

- « Le designer chargé de la création des bouquets standards ne désire pas avoir dans la base de données la quantité de chaque item (fleur ou accessoire) utilisé dans la réalisation de ces bouquets. Selon lui, cette information n'a aucune valeur ; c'est le prix du bouquet qui détermine les quantités », cet énoncé nous a amené à créer la table BouquetStandard avec notamment l'attribut prix ainsi que la table CompositionBouquet_Standard avec notamment l'attribut quantité.
-
- « le client doit remplir un bon de commande [...] les clients commandent un seul arrangement floral par bon de commande [...] commande personnalisée si le client commande différents items (i.e. fleurs et accessoires) pour réaliser un arrangement floral personnalisé » à l'aide de ces informations, nous avons choisi de créer la table CompositionCommande_Personnalise qui regroupe ces informations.

Connaitre les compositions des bouquets qu'il soit standard ou personnalisé permet également une meilleure gestion du stock.

Nous souhaitons répondre le plus fidèlement aux consignes présentées.

De plus, nous voulions, en tant qu'ingénieure, répondre au cahier des charges et respecter l'ensemble des contraintes.

Nous avons codé l'interface utilisateur en csharp via Visual Studio. Afin d'assurer l'accès à notre base de données, nous avons installé le `package Nuget MySQL Connection` « `using MySql.Data.MySqlClient; »`.

L'interface utilisateur s'appuie principalement sur 2 axes.

Dans un premier temps, nous avons codé l'interface utilisateur du point de vue du `Client` :

Identification du Client :

S'il s'agit d'un nouveau client, ce dernier doit fournir certains renseignements : nom, prénom, numéro de téléphone, courriel, mot de passe, adresse de facturation, carte de crédit.

Avant d'enregistrer ces données dans la base de données, le système vérifie s'il existe déjà un autre client ayant le même courriel. Si tel est le cas, le système doit afficher un message d'avertissement demandant au client de choisir une autre combinaison, courriel - mot de passe.

Si le client n'est pas à sa première commande, il entre dans le système à l'aide de son adresse email et de son mot de passe.

Choix de la commande : bouquet standard ou commande personnalisé.

Livraison de la commande

Le client indique l'adresse de livraison, le message accompagnant l'arrangement floral et la date de livraison désirée. Toutes ces informations sont stockées dans la table Livraison.

De plus, si le client a effectué une commande pour un bouquet standard, alors le système informe le client d'une éventuelle pénurie avec le message d'alerte suivant qui s'affiche dans la console
« Attention : si la date de livraison souhaitée est <= 3 jours, il peut y avoir une difficulté d'approvisionnement »

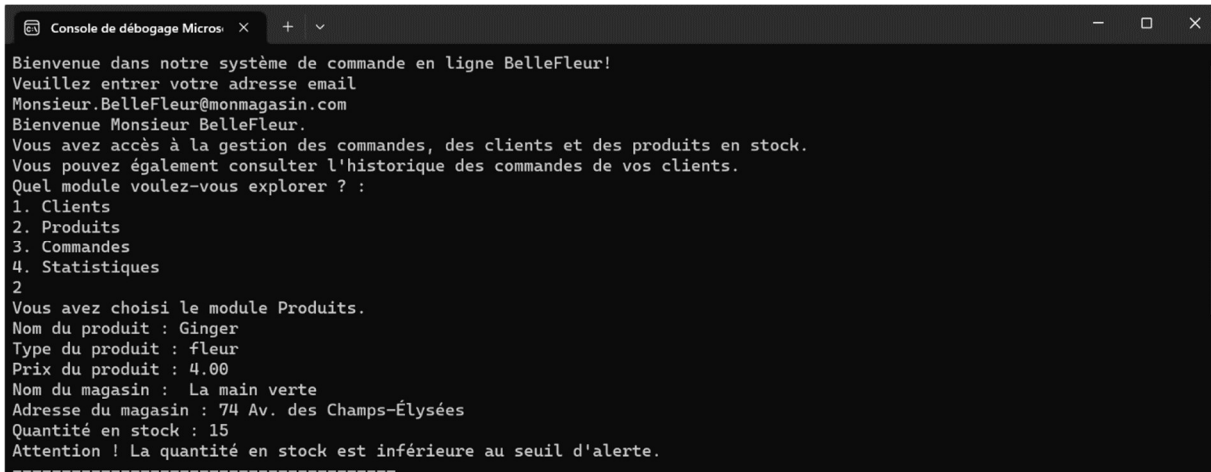
Proposer une remise sur le prix total de la commande en fonction du statut du client

Dans un premier temps nous avons codé l'interface utilisateur du point de vue du `Fleuriste` :

Le fleuriste accède aux différents modules, en entrant son adresse mail :
« Monsieur.BelleFleur@monmagasin.com »

Le fleuriste accède aux `modules` suivants :

- Client : ses caractéristiques, son statut de fidélité et son historique de commande
- Produit : Récupérer les informations de chaque produit et son stock dans chaque magasin
- Commande : Etats des commandes par date et par magasin, et nombre de commande par date.
- Statistique : **Requête synchronisée** pour les statistiques par an sur les bouquets, Requête avec **auto-jointure** pour les statistiques par mois sur les clients, requête avec **union** pour les statistiques par mois sur les produits et une Requête pour les statistiques sur les commandes.



```
Console de débogage Micros. X + -
Bienvenue dans notre système de commande en ligne BelleFleur!
Veuillez entrer votre adresse email
Monsieur.BelleFleur@monmagasin.com
Bienvenue Monsieur BelleFleur.
Vous avez accès à la gestion des commandes, des clients et des produits en stock.
Vous pouvez également consulter l'historique des commandes de vos clients.
Quel module voulez-vous explorer ? :
1. Clients
2. Produits
3. Commandes
4. Statistiques
2
Vous avez choisi le module Produits.
Nom du produit : Ginger
Type du produit : fleur
Prix du produit : 4.00
Nom du magasin : La main verte
Adresse du magasin : 74 Av. des Champs-Élysées
Quantité en stock : 15
Attention ! La quantité en stock est inférieure au seuil d'alerte.
-----
```

Vous pouvez constater que nous notifions au fleuriste une alerte stock qui permet à ce dernier d'optimiser sa gestion de stock.

Test et validation

Dès qu'un utilisateur passe une commande, celle-ci est ajoutée à la base de données sur MySQL WorkBench cf les tables : Commande, Livraison, CompositionCommande_Personnalise . Nous avons vérifié nos résultats, par exemple, par le biais de la requête suivante : « SELECT * FROM fleurs.commande ; » après avoir inscrit nos choix de commande dans l'interface utilisateur sur visual studio (code en csharp).

Ainsi notre interface est fonctionnelle car les modifications que l'on apporte sont mises à jour dans notre base de données.

Conclusion / Axe d'amélioration

Il aurait été plus esthétique de réaliser une interface graphique. Cependant, nous avons rejoint l'ESILV en A3, à la suite de deux années de classes préparatoires aux grandes écoles et avons pris le langage C/C++ au premier semestre.

Pour finir, nous souhaitons ajouter que nous avons trouvé ce projet très pertinent. Ce dernier nous a apporté énormément de connaissances et a cultivé notre envie de poursuivre au sein de la majeure Data & IA.