PODOLAK Yan
LIM Kevin

Rapport de Projet : Base de Donnée Cooking

I - Organisation de la Base de données

Table Cooking → Contient les employés de la société

Table client → Table contenant les informations sur l'entité client (nom, téléphone, adresse, est un créateur de recette ou non)

Table créateur de recette → Table contentant contenant le nom des créateurs de recette et leur solde. On relie les tables créateur de recette et client car un client peut être un créateur de recette.

Table recette → Table contenant les informations sur la recette (nom, type, description, prix, nom du créateur et nombre de fois ou elle a été commandée)

Table fournisseur → Table contenant les informations sur l'entité fournisseur (nom, téléphone)

Table Produit → Table contenant les informations sur l'entité produit (nom, catégorie, unité des quantités, stock actuel, stock minimum, stock maximum, nom du fournisseur, référence fournisseur)

Table Commande → Table contenant les informations sur l'association commande : relie un client avec une recette. On a un numéro de commande, et le nombre de fois où la recette est commandée. Si une commande contient plusieurs plats, on aura alors plusieurs commandes avec le même numéro de commande

Table Contient → Cette table relie les tables recette et produit. On a aussi la quantité utilisée pour la recette.

II - Options de Codage

Programmation Orientée Objet

Nous avons décidé fonctionner en POO avec la création de classes. La première des classes que nous avons crée est la classe CoSQL. Elle permet de créer la connexion avec la base de données MySQL et elle contient quelques fonctions nous permettant d'effectuer des requêtes à la base (une permettant de faire des requêtes telles que des insertions, suppression, modification et une autre permettant d'effectuer des requêtes ayant une/des sortie(s)). Les autres classes que nous avons crées correspondent aux différentes entités du schéma MCD à l'exception du « créateur de recettes » que nous avons décidé de fusionner ses propriétés avec celles du client. Pour chacune des classes, nous avons créé deux constructeurs :

- Lectures des propriétés de l'entité : Le premier constructeur prend la clé primaire en paramètre (ainsi que l'objet connexion) et construit l'objet avec toutes les informations de la base.
- Création d'une nouvelle entrée : Le deuxième constructeur permet d'insérer une nouvelle entrée dans la base avec les informations saisies par l'utilisateur (tous les paramètres de la classe/toutes les clés de la table)

Navigation dans l'interface Cooking

Nous avons décidé de réaliser dans un premier temps toute la navigation dans les différentes fonctionnalités à base de « switch ». Cela nous a permis d'organiser le code de manière plus propres, en mettant les menus (liste des fonctionnalités) des différentes entités dans des méthodes différentes. (ex : Menu_CdR, Menu_Cooking, Menu_client, Espace_CdR)

La structure générale de navigation dans l'interface s'effectue dans une fonction Execution où nous appelons les autres fonctions (tels que les menus…)

III - Développements complémentaires

Nous sommes restés sur une navigation dans la console grâce à l'implémentation des couleurs, de la suppression de l'affichage inutile. Nous avons songé à l'implémentation de l'application WPF, cependant le manque de connaissance sur le sujet due à l'absence au premier semestre nous a dissuadé à passer un temps précieux sur une partie facultative au profit d'une structuration optimale du code ainsi qu'à son affichage.

Nous avons réalisé deux tests unitaires que vous retrouverez dans le deuxième projet de la solution.

- Test_Client_Modifier_Tel() : Cette fonction permet de vérifier la connexion à la base de donnée (Atttention à bien mettre le bon mot de passe), la création d'un objet client et la modification du numéro de téléphone de ce client.
- Test_Recette_Modifier_Prix() : Cette fonction nous permet de vérifier la connexion à la base de donnée, la création d'un objet recette, ainsi que la modification du prix de cette recette.