

Notes techniques explicatives

Cooking

➤ Le code

Pour la structure du code il nous a d'abord fallu estimer s'il était intéressant ou non d'implémenter des classes. Etant donné que chaque table et ses instances sont déjà contenus dans une base de données SQL, il ne nous semblait à priori pas pertinent de les intégrer. En effet, nous n'avions pas particulièrement besoin d'importer des données en mémoire, d'autant plus que chaque méthode de classe pourrait être effectuée à l'aide d'une simple écriture de requête SQL. Malgré la longueur potentielle du script, nous avons décidé de garder cette solution, même si nous savions que les classes auraient sûrement permises d'alléger la structure du code.

Toutefois, bien que les requêtes SQL soient simples à intégrer dans C#, elles nécessitent plusieurs lignes de code et on s'est alors rapidement retrouvé avec un fichier très long en écriture. Pour y remédier, nous avons multiplié les zones de régions par « fonctionnement » : regroupement de toutes les méthodes servant à cooking ensemble, les méthodes pour le client tels que l'inscription, connexion, commande... bien entendu certaines méthodes, tels que l'affichage des plats, n'ont pas été écrites à deux reprises pour cooking et le client par exemple.

En dehors de la connexion à la base de données, tout a été codé en dehors du Main afin d'avoir une navigation fluide de sorte qu'il est possible de revenir en arrière, passer d'un menu à un autre sans jamais être interrompu et avoir à exécuter le code à nouveau. C'est pourquoi toutes les méthodes intègrent en paramètre une instance de la classe « MySqlConnection » qui recevra toujours la même instance de connexion associé à notre base de données cooking.

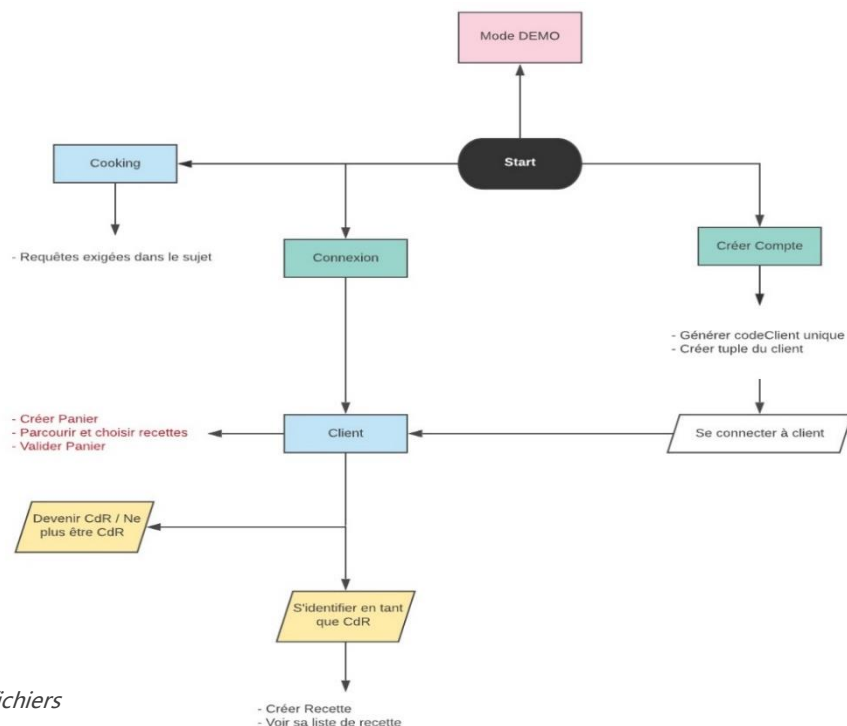


Image déposée avec les autres fichiers

Par ailleurs nous avons fait le choix dans notre code de définir une semaine du lundi au lundi, et non pas du jour J-7 au jour J.

De plus, au cours de la rédaction du code, nous avons modifié notre base de données pour alléger la complexité et la longueur des requêtes SQL, en intégrant, par exemple un compteur de vente total et un compteur de vente à la semaine pour les recettes. Pour pallier aux problèmes ajoutés avec cette solution, nous avons ajouté des fonctions de remise à zéro et de mise à jour de la base de données.

➤ Database et SQL

Un choix de fusion ou non était à faire après une relation [1, 1] dans notre modèle entité/association entre les classes clients et CdR. Etant donné qu'il n'y a pas de grande distinction entre un client et un créateur de recette, et qu'il est possible de passer de l'un à l'autre très rapidement et facilement, nous avons estimé qu'il était plus pertinent de les fusionner.

- Voici le schéma relationnel obtenu :

Fournisseur (codeFournisseur, nomF, telephoneF)

Produit (codeProduit, nomP, catégorie, stock, stockMax, stockMin, unite, dernieUtilisation)

Recette (codeRecette, nomR, type, descriptif, veg, prixR, remuneration, #codeClient, nombreCommandeSemaine, nombreCommande)

Panier (codeCommande, date, prixP, #codeClient)

Client (codeClient, nomC, prenomC, telephoneC, usernameC, mdpC, createur, cook, nombreCommandeCdR)

Fournie (#codeFournisseur, #codeProduit)

ConstitutionRecette (#codeRecette, #codeProduit, quantiteProduit)

ConstitutionPanier (#codeCommande, #codeRecette, quantiteRecette)

Unicité / identifiant / #clé étrangère

La création de la table « fournie » vient du fait que selon notre vision, un produit peut provenir de plusieurs fournisseur (relation [n, n] avec les fournisseurs qui fournissent plusieurs produits).

Certaines contraintes sur les attributs ont ensuite été ajoutées lors de la création des tables sur MySQL, par exemple le type d'une recette ne peut être qu'une entrée, un plat, un dessert ou un fromage.

Par ailleurs, cooking a été instancié en tant que CdR afin d'y associer les recettes créées. Il n'est toutefois pas supprimable avec le gestionnaire cooking permettant de supprimer des créateurs de recettes dans C#.

Remarque :

Nous avons décidé de rédiger notre programme en anglais par habitude ainsi que pour respecter les « conventions » internationales. En travaillant ainsi, il nous est plus facile d'être relu par quelqu'un ne connaissant pas le français.