



ÉCOLE
D'INGÉNIEURS
PARIS-LA DÉFENSE

Rapport du projet Cooking



I. Diagramme E/A

Lors la modélisation de notre diagramme entité-association, nous avons effectué différents choix, en particulier concernant les cardinalités de nos associations. Ainsi, nous avons considéré qu'un produit pouvait être fourni par un seul et unique fournisseur, tandis qu'un fournisseur se devait de fournir au moins un produit pour figurer par la suite dans notre base de données. De la même façon, une recette peut être réalisée par un seul et unique cuisinier, tandis qu'un cuisinier peut tout de même réaliser plusieurs recettes, ou encore une recette est proposée par un seul et unique Créateur de Recettes.

Par ailleurs, nous avons choisi de modéliser l'association entre l'entité Recette et l'entité Produit par une association Est Constitué (1,N),(1,N), qui a pour propriété quantitéUtilisée, c'est-à-dire un entier qui définit la quantité d'un produit (exprimée selon l'unitéDeQuantité de ce dernier) utilisée dans l'élaboration d'une recette. Suivant cette même logique, nous avons associé les entités Recette et Client à travers cette fois-ci l'association Commande (0,N),(0,N) (dans la mesure où une recette peut ne pas être commandée par un client, tout comme un client peut ne pas commander une recette), qui a pour propriété quantité, qui correspond au nombre de recette commandé pour un client associé.

II. Base de données

Nous avons généré notre base de données à travers l'exécution d'un script MySQL, après traduction du diagramme E/A. Pour ce faire, nous avons d'abord généré une base de données appelée cooking, puis chaque table. D'ailleurs, nous avons décidé de fusionner les entités Client et Créateur de Recette en une seule et même table appelée Client, dans la mesure où il y avait une association (1,1) du côté de l'entité Créateur de Recette. Nous avons retenu idCompte comme clef primaire pour la table Client nouvellement créée du fait qu'un Créateur de Recette est forcément un client. Pour distinguer un Créateur de Recette d'un simple client, ce dernier aura un solde cook initialisé à NULL, tandis que celui d'un CdR prendra pour valeur 0 à la création du compte.

Nous avons ensuite rempli « manuellement » notre base de données, en ajoutant des entrées pour chacune de nos tables.

III Codage de notre solution

Nous avons codé notre application Cooking en utilisant WPF. Lors de l'initialisation de l'application, nous demandons à l'utilisateur de saisir via une boîte de dialogue l'identifiant et le mot de passe du serveur local utilisé pour accéder à ses bases de données MySQL.

Puis l'utilisateur se retrouve sur la fenêtre principale de Cooking. Plusieurs choix s'offrent à lui : il peut se connecter en saisissant son identifiant (qui correspond à la clef primaire d'une instance de la table Client), et son mot de passe, accéder au mode démo, à l'interface de gestion de Cooking, ou bien tout simplement créer un compte s'il n'est pas encore client.

Lorsqu'il se connecte et accède au menu client, il peut effectuer sa commande, gérer son compte (comme changer son mot de passe par exemple), ou bien accéder au menu CdR pour créer une recette ou encore consulter son solde cook et les recettes qu'il a créées. Si jamais le client n'est pas encore un CdR, il peut en faire la demande en cliquant sur le bouton d'accès au menu CdR via une boîte de dialogue qui lui propose le changement de statut.