Projet Cooking BDD

Concept :

Nous avons interprété le sujet de la sorte : coder une application WPF permettant à des clients de commander des recettes créées par des créateurs de recettes. Cela implique alors une interface de création de recette, une interface de commande mais aussi une gestion de compte et de solde.

Il y a quelques « bonus » tels que : un classement des recettes les mieux vendues, des créateurs vendant le plus de recettes.

Le tout étant connecté à une base de données afin de sauvegarder toutes informations.

Notre méthode de programmation :

Dès lors que l’on devait passer par notre base de données nous avons appliqué une méthode semblable pour presque toutes les requêtes SQL. En prenant l’exemple d’une simple sélection mais avec certaines condition un peu complexe nous ne passions jamais par des JOIN ON ou des GROUP BY, en effet nous faisions tout simplement un SELECT … FROM … WHERE …, puis avec une boucle while(reader.Read()){} nous récupérons chaque chaîne de caractères (par exemple les pseudos) avec des espaces entre elles afin de les .split(‘ ‘) dans une liste. C’est donc plus avec des manipulations C# que nous nous sommes débrouillés qu’avec SQL. Cependant nous avons quand même utilisé certains aspect un minimum « avancés » du SQL avec des ORDER BY, DELETE, UPDATE.

Problème(s) rencontré(s) :

Notre principal obstacle s’est montré dès le début de notre avancée sur le projet : comment transmettre des variables d’une fenêtre à une autre ? Par exemple après s’être connecté on arrive sur la page « Mon compte » mais comment la fenêtre « Mon Compte » va savoir le compte de quelle personne afficher ? Après avoir beaucoup réfléchi nous avons trouvé un système, peut-être pas très optimisé, on ne sait pas, mais il fonctionne très bien. Lors de l’instanciation de la fenêtre « Mon Compte » on transmet le contenu de la textbox « identifiant » à un textblock qui est caché dans la fenêtre « Mon Compte ».

Enfin on a eu un peu de mal à résoudre le problème suivant : lorsqu’on appelait certaines fenêtres en leur transmettant l’id\_client du client connecté elles « recevaient » bien ce paramètre mais les outils qui était censés utiliser cet id\_client ne le reconnaissait pas. On ne comprenait pas comment deux éléments de la même page pouvait « réagir différemment » vis-à-vis de e paramètre transmis. Finalement on s’est rendus compte que l’id\_client que l’on transmettait au moment de l’instanciation de la fenêtre n’était pas le problème mais en fait c’était toutes les fonctions qui était contenues dans le constructeur qui posaient problème. En effet elle utilisait l’id\_client alors qu’il n’était même pas encore transmis… C’et pour quoi on a créer, sur plusieurs pages, une fonction « tout() » qui contient tout ce qu’il y avait dans le constructeur, on pouvait alors appeler tout() juste après avoir transmis l’id\_client à la fenêtre.

Erreur(s) persistante(s) :

Lorsque l’on passe une commande on achète tout ce qu’il y a dans le panier donc logiquement on a acheté tout ce dont on avait besoin. Cependant nous avons pris l’initiative de mettre un bouton permettant de repasser une commande tout de suite. Or en utilisant cette fonction vous tomberez sur un payement refusé peut importe votre solde et le montant du panier. Nous avons essayé de résoudre ce problème mais c’est la seule (on espère) erreur qui persiste encore.

Particularité de notre projet :

Il faut d’abord savoir que pour ce projet nous avons commencé à travailler bien à l’avance ce qui fait que lors du TD de revue du schéma E/A nous avions déjà pratiquement terminé toute notre application. Or c’est lors de ce TD que notre professeur nous a fait remarquer que nous n’utilisions pas de table commande. En effet nous avons dans un premier temps même pas pensé à faire de table commande car nous n’en avons pas ressenti le besoin. Beaucoup plus tard, lors du TD et après avoir vu quelques mises à jour du sujet nous nous sommes rendu compte que pour certaines demandes du cahier des charges nous étions presque obligés de créer une table commande. Notre code était malheureusement déjà beaucoup trop avancé pour changer un élément aussi majeur que celui-ci. C’est pourquoi nous avons crée une table commande qui ne sert strictement qu’à être affiché dans le document XML afin de remplir cette charge du Cahier.

C’est donc aussi pour cette raison que nous avons délibérément choisi de faire l’impasse ou plutôt d’apporter une modification concernant certains éléments comme le classement par semaine des créateurs de recettes, nous avons fait un classement général sans notion de temps.

Conclusion :

Nous nous sommes beaucoup investis sur ce projet en termes de temps nous espérons donc que vous le remarquerez car même si parfois nous avons pris des libertés par rapport au sujet nous avons essayé d’être le plus cohérent possible par rapport à une vraie application de commande de plats.

Annis MECILI et Yannis MEGHAGHI ESILV TD-J