BTS SIO 2 SLAM - PPE

Contexte GSB

Travail n°5 : Clôture et mise en remboursement automatique des fiches

Le cahier des charges de l’application Frais GSB stipule que la fiche d’un visiteur est clôturée au dernier jour du mois. Cette clôture sera réalisée par l’application selon l’une des modalités suivantes :

* A la première saisie pour le mois N par le visiteur, sa fiche du mois précédent, N-1, est clôturée si elle ne l’est pas déjà.
* Au début de la campagne de validation des fiches par le service comptable, un script est lancé qui clôture toutes les fiches non clôturées du mois qui va être traité.

Nous nous intéresserons ici à la deuxième éventualité

D’autre part, il est dit que la mise en paiement est faite au 20 du mois suivant la saisie par les visiteurs.

Nous voudrions répondre à ces deux objectifs en développant une application C# avec VS.Net.

Cette application va devoir permettre, au début de la campagne de validation, c’est-à-dire à partir du 1er jour du mois N, la clôture de toutes les fiches créées le mois N-1.

Elle permettra, d’autre part, à partir du 20è jour du mois N la mise en remboursement des fiches validées du mois N-1.

Nous allons pour cela réaliser les tâches suivantes :

Tâche 1 : Création de la classe d’accès aux données

* Créez un nouveau projet C# Windows Form
* Créez une classe que vous allez nommer **ConnexionSql**. Cette classe va nous permettre d’accéder à la base de données et récupérer les informations utiles. Il faudrait, avec l’utilisation du design singleton, ne permettre l’ouverture que d’une seule connexion à la fois.
* Les différentes variables serveur, base de données, utilisateur et mot de passe, qui vont nous permettre de construire la chaîne de connexion, doivent bien sûr être complètement paramétrables.

Tâche 2 : Création de la classe de gestion des dates

Pour ne récupérer que les fiches créées du mois précédent, il faudrait d’abord récupérer le mois courant. Pour cela on peut créer une classe **GestionDate** avec différentes méthodes qui vont nous permettre, à partir de la date du jour (DateTime.now) de récupérer le mois courant, le mois précédent etc….

Il est obligatoire dans ce cas-là de créer aussi un **projet de tests unitaires** qui va valider les différentes fonctions développées.

**Tests Intégration**

Pour tester le tout, Allez dans le code de la feuille et, en variable de classe, déclarez et instanciez un objet de type ConnexionSql. Le paramètre représente la chaîne de connexion. Pour afficher les lignes récupérées, il faut mettre dans le formulaire des composants graphiques (ListView ou DataGridView) et les lier à un DataReader, à charge pour ce dernier de récupérer toutes les fiches créées du mois précédent.

Tâche 3 : Création de la classe principale

Pour que l’ordinateur contrôle régulièrement les différentes échéances, il faut positionner un **Timer**. On peut paramétrer ensuite l’intervalle d’exécution de ce composant (30000 = 30 secondes). Pour les tests, pensez à mettre des intervalles courts.

Derrière la procédure événementielle (Tick) de ce Timer, on va appeler les méthodes qui permettront la connexion à la base de données, la récupération des fiches créées du mois N-1 et leur mise à jour, en les mettant à l’état ‘CL’.

En supposant que la campagne de validation va se passer entre le 1er et le 10 du mois courant, on va, en comparant les dates, s’assurer que l’on se trouve bien dans cet intervalle-là.

De la même manière, à partir du 20è jour du mois, on va mettre à jour les fiches validées du mois précédent en les passant à l’état ‘RB’.

Pensez à refermer la connexion après chaque passage du Timer.

Testez en générant des lignes appropriées dans votre base de données, en modifiant les dates de clôture et de mise en remboursement et en raccourcissant les intervalles du Timer !

**On voudrait aussi que cette application s’exécute de manière continue en tâche de fond sans qu’elle ne soit nécessairement visible à l’utilisateur. Trouvez le moyen le plus approprié pour le faire.**

**Remarque**: Votre application doit comporter aussi une gestion des erreurs (exceptions) qui va nous assurer un comportement adéquat à tout instant de cette application. Pensez aussi à faire des commentaires à vos classes et méthodes pour ensuite générer une documentation XML (ou HTML).