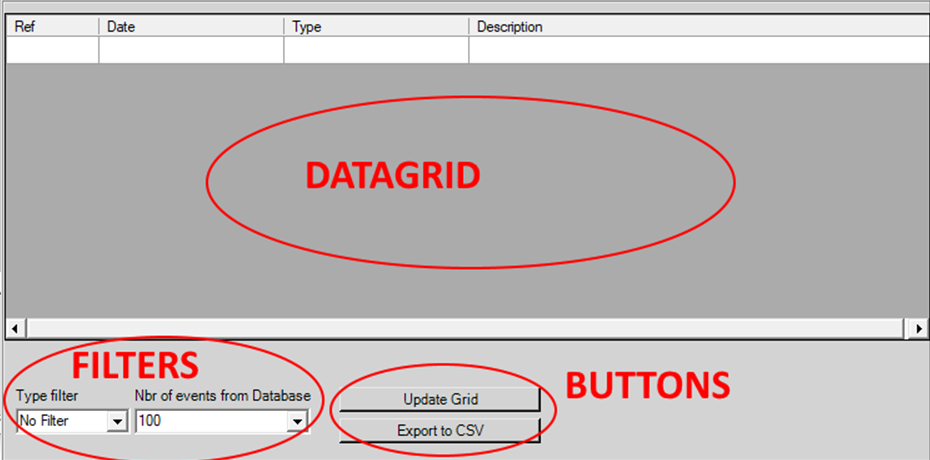
Plateforme.net III (C#) – 2IG

Examen de première session F.Van Goethem - 29/06/2020

|  |
| --- |
| L’examen consiste en une mise à jour du projet développé durant le quadrimestre, dont le contenu est défini dans le cahier des charges « Interface\_CSharp\_V07052020 », disponible sur e-campus.  Le code rédigé durant l’examen est l’unique partie du projet soumise à évaluation pour la première session d’examens. Ce code doit obligatoirement être placé entre balises pour être évalué ! Voici un exemple :  Ces balises sont :  /\*\*\*\*\*\*CODE EXAMEN: DEBUT\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*CODE EXAMEN: FIN\*\*\*\*\*\*\*\*\*\*\*/  L’évaluation porte sur :   * La programmation orientée objets (35%) * La manipulation des fichiers (15%) * La gestion d’erreurs (15%) * La manipulation des bases de données (35%)   En solution alternative, vous pouvez remplacer la base de données par une liste d’objets dans votre code. La grille d’évaluation sera alors la suivante :   * La programmation orientée objets (35%) * Les collections (10%) * La manipulation des fichiers (15%) * La gestion d’erreurs (15%) * ~~La manipulation des bases de données~~ (~~35%~~ -> **-25%**)   Cette solution alternative engendre la perte de 25% des points.  La mise à jour à implémenter dans votre projet est détaillée ci-dessous. |

**Fonctionnalité à ajouter :** Ajoutez à votre application une page d’historique d’évènements (Logs) en utilisant une base de données de type Access.

**Exemple d’interface :**

****

**Base de données DB\_Logs, table LogsTable :**

|  |  |  |
| --- | --- | --- |
| **Colonne** | **Propriétés DB** | **Description** |
| LogRef | Clé primaire, indexé sans doublon, long int, auto-incrémentée. | Référence numérotée |
| LogDate | Texte court 50 caractères, Null interdit, Champ vide interdit, non-indexé | Date et heure de l’évènement (DateTime.Now) |
| LogType | Texte court 50 caractères, Null interdit, Champ vide interdit, indexé avec doublon. | 3 types possibles :   * User : login, création, suppression utilisateur * ID : Création/Suppression d’ID * Alarm : Alarmes sur mesures |
| LogDescription | Texte court 50 caractères, Null interdit, Champ vide interdit, non-indexé, | Doit comprendre pour un type donné  - User : nom d’utilisateur enregistré  - ID : ID créé ou supprimé + config associée.  - Alarm: ID, description d’alarmes, valeur associée |

Les caractéristiques de la base de données sont :

* Chaque évènement est directement envoyé dans la base de données.
* Il n’y a pas de limitation dans le nombre d’évènements stockés dans la base de données.
* L’accès à la base de données en mode « déconnecté » (utilisation du dataset) doit être privilégié.

**Datagrid :**

Un Datagrid permet l’affichage des données extraites de la base de données suivant les filtres imposés par l’utilisateur.

**Filtres :**

Deux filtres permettent la sélection du type (pas de filtre, User, ID, Alarm) ainsi que du nombre d’évènements (ex : 10, 100, 1000) à afficher dans le datagrid, en partant du plus récent.

**Buttons :**

L’interface doit aussi contenir

* Un bouton de mise à jour du Datagrid depuis la base de données suivant les filtres.
* Un bouton permettant la sauvegarde des logs en fichier CSV, suivant les filtres.