A picture containing logo

Description automatically generated

Projet – Bottin de téléphone

Livrable 1 (20%)

2022-08-15

# Objectif

Mettre en pratique les notions de base de données à l’intérieur d’une application.

Transposer la théorie des requêtes SQL dans une programmation orienté objet.

Comprendre le mécanisme sous-jacent d’un ORM classique.

# Énoncé

À la suite de la conception de votre schéma de base de données on vous demande de créer l’application de carnet d’adresse. On vous demande de commencer par les commandes liées à la gestion de la structure de la base de données.

# Besoins non-fonctionnels

* Votre exécutable doit se nommer bottin.exe
* L’application doit être développé en C#
* L’application est une application console
* Vous devez utiliser Mysql.Data.MysqlClient
* L’utilisation de « Entity » Framework ou autre n’est pas permise
* L’importation d’un script ou fichier SQL n’est pas permise

# Besoins fonctionnels

Liste des besoins fonctionnels de votre application

* L’application offre des options pour les actions possibles
* L’application offre les options suivantes :
  + --migrate : pour créer les tables
  + --rollback : supprimer toutes les tables
  + --refresh : recréer la base de données, soit supprimer toutes les tables et les recréer en une seule action
  + --seed: insert des données fictives dans la base de données
* On peut combiner certaines options.
  + --migrate --seed : va créer la base de données et ajouter les données fictives
  + --refresh --seed : va supprimer les tables, recréer la base de données et ajouter les données fictives

# Critère de corrections

|  |  |
| --- | --- |
| Les besoins non-fonctionnels sont respecté | 1% |
| L’option pour la création des tables est présente et fonctionnelle | 2% |
| L’option pour la suppression des tables est présente et fonctionnelle | 2% |
| L’option pour recréer la base de données est présente et fonctionnelle | 2% |
| L’option pour créer les données fictives est présente et fonctionnelle | 2% |
| Les instructions SQL sont dans le programme | 1% |
| Chaque table doit être représenté par une classe qui permet de la créer | 2% |
| Chaque classe pour créer une table doit posséder la fonction ‘’**migrate’’** qui contiendra le code pour créer ladite table | 2% |
| Chaque classe qui sert à créer une table doit être contenu dans le dossier **migrations** | 2% |
| Vous devez créer une classe pour la gestion de la connexion à la base de données | 2% |
| Vous devez respecter le diagramme de base de données et ses contraintes | 2% |

# Remise:

Date de remise : Cours 6 23h59

Vous devez remettre :

* Votre diagramme de base de données
* Votre dépôt final doit contenir le fichier .exe
* Une remise régulière sur Bitbucket est exigée tout au long du projet. (Minimum tous les vendredis.)

# Bitbucket :

Je tiens pour acquis que les connaissances de gestionnaire de version sont bien acquises étant donné qu’un vidéo devrait vous avoir été fourni dans le cours de web.

L’invitation à votre projet devra m’être envoyé le plus tôt possible.

# Références :

* <https://dev.mysql.com/doc/connector-net/en/connector-net-connections-string.html>
* <https://dev.mysql.com/doc/connector-net/en/connector-net-programming-mysqlcommand.html>
* <https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-sql-command.html>
* <https://docs.microsoft.com/en-us/dotnet/api/system.console?view=net-6.0>
* <https://docs.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-console?view=vs-2022>