**Quiz**

1. **Couche Models**

* 1 **Category** contient plusieurs Quiz
  + Relation OneToMany ==> Many (Quiz) et One (QuizCategory)
* 1 **Quiz** peut contenir (1..\*) **QuizQuestion** (OneToMany)
* 1 **QuizQuestion** peut contenir (1..\*) **QuizResponse** (OneToMany)
* 1 User appartient à (1..\*) QuizTest (OneToMany)
* 1 Quiz peut contenir (1..\*) QuizTest (OneMany)

|  |  |  |  |
| --- | --- | --- | --- |
| **QuizCategory** | **Quiz** | **QuizQuestion** | **QuizResponse** |
| Id : int | Id : int | Id : int | Id : int |
| Title : string | Title : string | QstText : string | RespText : string |
|  | Category : QuizCategory | MultipleChoice : bool | Correct : bool |
|  | **CategoryId : int** | NumOrder : int | Question : QuizQuestion |
|  |  | Quiz : Quiz | QuestionId : int |
|  |  | QuizId : int |  |

|  |  |
| --- | --- |
| **QuizTest** | **User** |
| Id : int | Id : int |
| CreationDate : DateTime | Name : string |
| User : User | Email : string |
| UserId : int | Password : string |
| Quiz : Quiz | TotalPoints : int |
| QuizId : int |  |
| Score : int |  |

1. **Context**

Créer le context (qui me généra la chaine de connexion dans app.config) + les DbSet

Modifier la chaine de connexion en ajoutant le nom de votre serveur et le nom de la base de données

* Activer les migrations (à exécuter qu'une seule fois dans le projet) enable-migrations
* Faire un add-migration nomMigration

***Remarque*** : A chaque mise à jour des classes objet - add-migration nomMigration

* update-database

1. **Couche Repositories**

|  |  |
| --- | --- |
| **IUserRepositoty** | **IQuizRepository** |
| FindAll() : List<User> | FindQuizzes() : List<Quiz> |
|  | FindQuizById (int quizId) : Quiz |
|  | FindQuestion (int quizId, int numOrder) : QuizQuestion |
|  | FindResponses (int questionId) : List<QuizResponse> |
|  | FindResponseById (int responseId): QuizResponse |
|  | CountQuestions(int quizId) : int |

1. **Couche Services**

|  |  |
| --- | --- |
| **IUserService** | **IQuizService** |
| FindAll() : List<User> | FindQuizzes() : List<Quiz> |
|  | FindQuizById(int quizId) : Quiz |
|  | FindQuestion(int quizId, int numOrder) : QuizQuestion |
|  | FindResponses(int questionId) : List<QuizResponse> |
|  | FindResponseById(int responseId) : QuizResponse |
|  | CountQuestions(int quizId) :int |

1. **Couche Présentation (WPF)**
2. **Liste de Fenêtre**

|  |  |
| --- | --- |
| **Liste de fenêtre** | **Description** |
| **QuizWindow.xaml** | Contient :   * Un Timer *(Name = TBTimer) :* ***TextBlock*** * Titre du Quiz *(Name = TBQuizTitle) :* ***TextBlock*** * Titre de la question *(Name = TBQuestionTitle) :* ***TextBlock*** * Liste de réponses à une question *(Name = PanelResponses) :* ***StackPanel*** * Un bouton pour passer à la question suivant *(Name = BtnNext)* |
| Fenêtre pour le choix du Quiz **QuizChoiceWindow.xaml** | Contient :   * Un ComboBox pour le choix du quiz * Un bouton « Démarrer » permettant de commencer le Quiz |

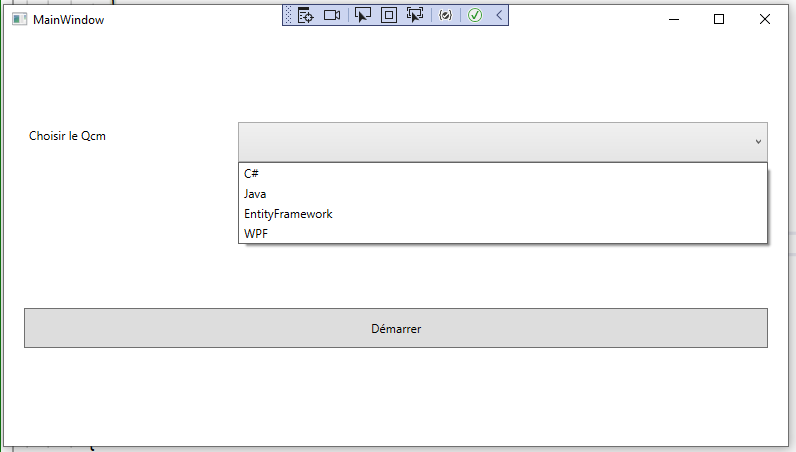
1. **QuizChoiceWindow.xaml.cs**

|  |  |
| --- | --- |
| Méthodes | Descriptions |
| Window\_Loaded | Charge les Quizzes dans le comboBox au lancement de l’application |
| BtnStart\_Click  Démarre le Quiz | On sélectionne un Quiz dans le comboBox :   * Si on a sélectionné un Quiz. * Récupérer le quiz sélectionné * Créer une fenêtre QuizWindow et lui donner le quiz * Sinon on affiche un message « Vous devez choisir un Quiz» |

1. **QuizWindow.xaml.cs**

|  |  |
| --- | --- |
| Méthodes | Descriptions |
| TimerJob (object sender, EventArgs e) | Méthode qui sera déclenchée à chaque Tick (intervalle)   * Si le temps > 0. * On met à jour le timer (interface graphique)   Sinon on passe à la question suivante |
| Window\_Loaded | Récupère le nombre de question  Mets à jours le titre du Quiz  Appel la méthode **LoadQuestion()** : charge une question |
| LoadQuestion() | Charge une question :   * Arrêter le timer * Initialiser TBTimer = 5 * Démarrer le Timer * On récupère la question dans la base de données et on met à jour le titre de la question sur l’interface graphique. * On récupère la liste des réponses de la question * On efface le PanelReponses * Si c’est une question à choix multiple * On parcourt la liste des réponses * Pour chaque réponse * On crée un checkbox et on met à jour le « content » du checkbox avec la réponse. * On ajoute la réponse dans la liste des réponses (PanelResponses) * Sinon c’est une question unique * On parcourt la liste des réponses * Pour chaque réponse * On crée un radioButton et on met à jour le « content » du radioButton avec la réponse. * On ajoute la réponse dans la liste des réponses (PanelResponses) |
| BtnNext\_Click | Permet de passer à une question suivante   * Vérifier si la réponse utilisateur est correcte * Si c’est une question à choix multiple * On parcourt la liste des réponses de cette question * On met chaque réponse dans une variable CheckBox * Si la réponse a été cochée * Si la réponse cochée est correcte * Son score augmente * Sinon * Son score diminue * Si c’est une question à choix unique * On parcourt la liste des réponses de cette question * On met chaque réponse dans une variable RadioButton * Si la réponse a été sélectionnée * Si la réponse sélectionnée est correcte * Son score augmente * Vérifier s'il reste des questions => suivante ou afficher le score * S'il y a des questions on incrémente le nombre de question et on charge la question * Sinon on arrête le timer * Panel des réponses et le bouton Next deviennent invisible * On met à jour le titre de la question en affichant son score |
|  |  |

**QuizChoiceWindow**



**QuizWindow**

