**Application Echange de livre**

1. **Couche Models (Dossier Models)**
2. **Classes :**

* 1 User peut avoir plusieurs Books (OneToMany)
* 1 Book appartient à 1 User (ManyToOne)
* BookExchange : On peut échanger un livre plusieurs fois
* 1 propriétaire peut faire plusieurs échanges
* 1 livre peut faire plusieurs échanges.

|  |  |  |
| --- | --- | --- |
| Classe User | Classe Book | Classe BookExchange  *« Classe contenant les données du livre échangé »* |
| Id : int | Id : int | Id : int |
| Name : string | ISBN : string | CreationDate : DateTime |
| Email : string | Title : string | Book : Book |
| Password : string | Author : string | BookId : int |
| TotalPoints : int | Price : double | OldOwner : User  *« L’ancien propriétaire du livre »* |
|  | EditionDate : DateTime | OldOwnerId : int |
|  | State : BookState *« enum »* |  |
|  | *enum* BookState :  {VERY\_GOOD, GOOD, MEDIOCRE, WORN} |  |
|  | Owner : User  *« Propriétaire du livre »* |  |
|  | OwnerId : int *« Champs optionnels »* |  |
|  | PointsValue : int |  |

1. **Context**

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

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
* Utilisation de l’approche **code first** :

1. Créer les classes
2. Créer le context + les dbSet
3. Activer la migrations (à exécuter qu’une seule dans le projet) => **enable-migrations**
4. A chaque mise à jour des classes objets => **add-migration nomMigration**
5. **Update-database**
6. **Couche DAO (Dossier Repositories)**

|  |  |
| --- | --- |
| IUserRepository | IBookRepository |
| FindAll()  *« Retourne la liste des Users »* | FindAll()  *« Retourne la liste des livres du User*  *»*  *« On inclut l'entité User (le propriétaire du livre) lors du chargement des books »* |
|  | SaveOrUpdate(BookExchange exc)  *« Permet de mettre à jour ou de sauvegarder le livre échangé »* |

1. **Couche Services (Dossier Services)**

|  |  |
| --- | --- |
| IBookService | IUserService |
| FindAll() | FindAll() |
| ValidateExchange(Book b, int newOwnerId)  Met à jour le nouveau propriétaire du livre   * b : livre échangé * newOwnerId: Contient l’Id du Nouveau propriétaire du livre |  |

1. **Couche Présentation ---A suivre (Cours WPF)**