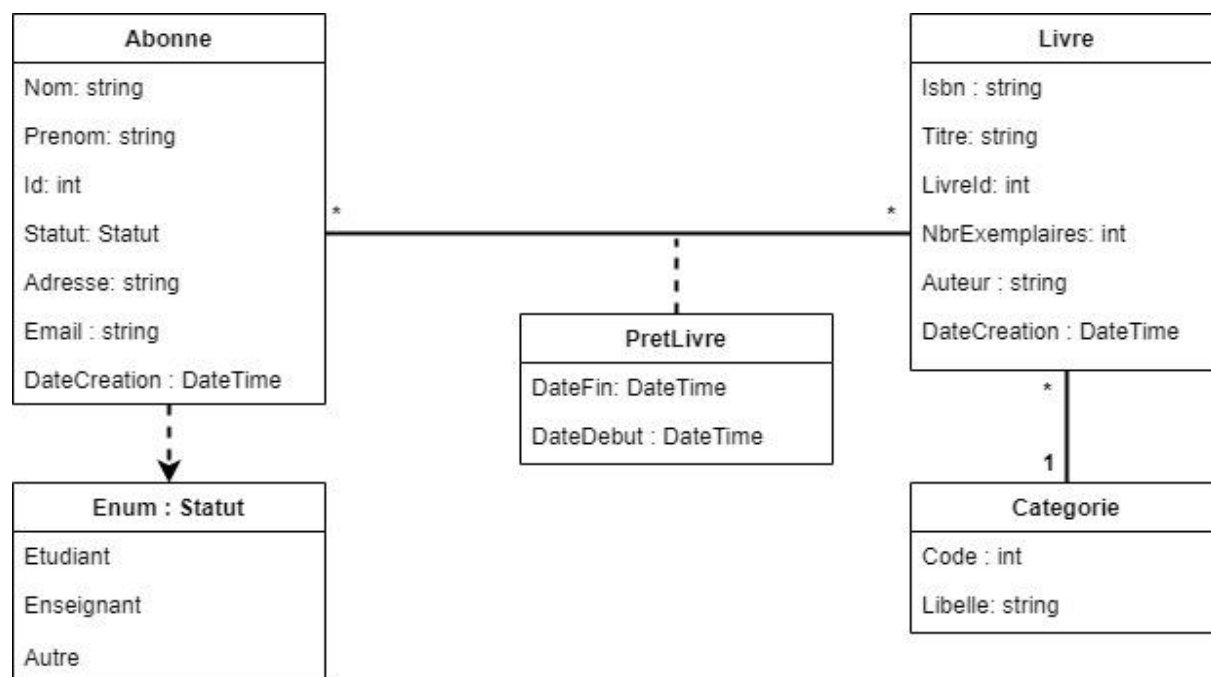
 esprit <small>Ecole Supérieure Privée d'Ingénierie et de Technologies</small>	<h2 style="margin: 0;">EXAMEN</h2> <p>Semestre : <input type="checkbox"/> 1 <input checked="" type="checkbox"/> 2</p> <p>Session : <input checked="" type="checkbox"/> Principale <input type="checkbox"/> Rattrapage</p>		
Module : Architecture des systèmes d'information I (.Net) Enseignants : Équipe .Net Classes : 4 TWIN Documents autorisés : <input checked="" type="checkbox"/> OUI <input type="checkbox"/> NON Nombre de pages : 3 Date : 17/05/2022 Heure : 8h30 Durée : 1h30m			
<table style="width: 100%; border: none;"> <tr> <td style="width: 60%; border: none; vertical-align: top;"> ETUDIANT(e) N° Carte : Nom et Prénom : </td> <td style="width: 40%; border: none; vertical-align: top;"> Classe : Salle : </td> </tr> </table>		ETUDIANT(e) N° Carte : Nom et Prénom :	Classe : Salle :
ETUDIANT(e) N° Carte : Nom et Prénom :	Classe : Salle :		

NB : -Toute carte Wifi active est une tentative de fraude
 -Respectez l'architecture vue en cours et les patrons de conception étudiés.

Enoncé du problème :

On souhaite développer une application pour la gestion d'une bibliothèque. L'application doit permettre aux abonnés d'emprunter des livres. Soit le diagramme de classe de l'application suivant :



Partie I: Entity-Framework (10pts)

1. Créer les entités en respectant le diagramme de classes. (3 pts)
2. En utilisant les annotations :
 - a. Configurer la propriété **Isbn** en tant que propriété obligatoire et la propriété **DateCreation** de la classe **Livre** en tant que date valide. (1 pt)
 - b. La propriété **Code** de la classe **Categorie** doit être une clé primaire. (1 pt)
3. En utilisant Fluent API :
 - a. Créer une classe de configuration associée à la classe **PretLivre** et la configurer comme porteuse de données entre **Livre** et **Abonne** avec une clé primaire composée des trois propriétés **LivreFk**, **AbonneFk** et **DateDebut**. (2 pts)
 - b. Créer une classe de configuration associée à la classe **Categorie** et configurer la relation entre **Livre** et **Categorie** . (1 pt)
4. Utiliser les migrations pour générer la base de données nommée sous la forme **BibliothequeNomPrenom**. (2 pts)

Partie II: Services (4pts)

Dans les classes de services spécifiques, créer:

5. Une méthode qui retourne le livre le plus emprunté. (1 pt)
6. Une méthode qui retourne les livres empruntés entre deux dates (passées en paramètres). (1 pt)
7. Une méthode qui retourne la liste des catégories des livres empruntés par des abonnés ayant un statut passé en paramètre. (2 pts)

Partie III: ASP MVC (6pts)

8. Alimenter manuellement la base de données par une catégorie, deux abonnés et deux livres. (0.5 pt)
9. Créer et tester une vue “Index” qui permet d'afficher la liste des emprunts en affichant le nom de l'abonné et le titre du livre. (1.5 pts)

DateDebut	DateFin	Abonne	Livre	
25/04/2022 20:47:00	27/04/2022 20:47:00	Ab1-nom	titre1	Edit Details Delete

10. Ajouter à la vue “Index” créé précédemment une zone de recherche par date de début et date de fin. (1.5 pts)

Index

Date Debut: Date Fin:

[Create New](#)

DateDebut	DateFin	Abonne	Livre	
26/04/2022 11:33:00	30/04/2022 11:34:00	Ab1-nom	titre1	Edit Details Delete
30/04/2022 11:34:00	10/03/2022 11:34:00	Ab2-nom	titre2	Edit Details Delete

11. Créer et tester la vue “Create” qui permet d’ajouter un prêt dans la base de données. (2.5 pts)

DateDebut

DateFin

Abonne

Livre

Bon travail 😊