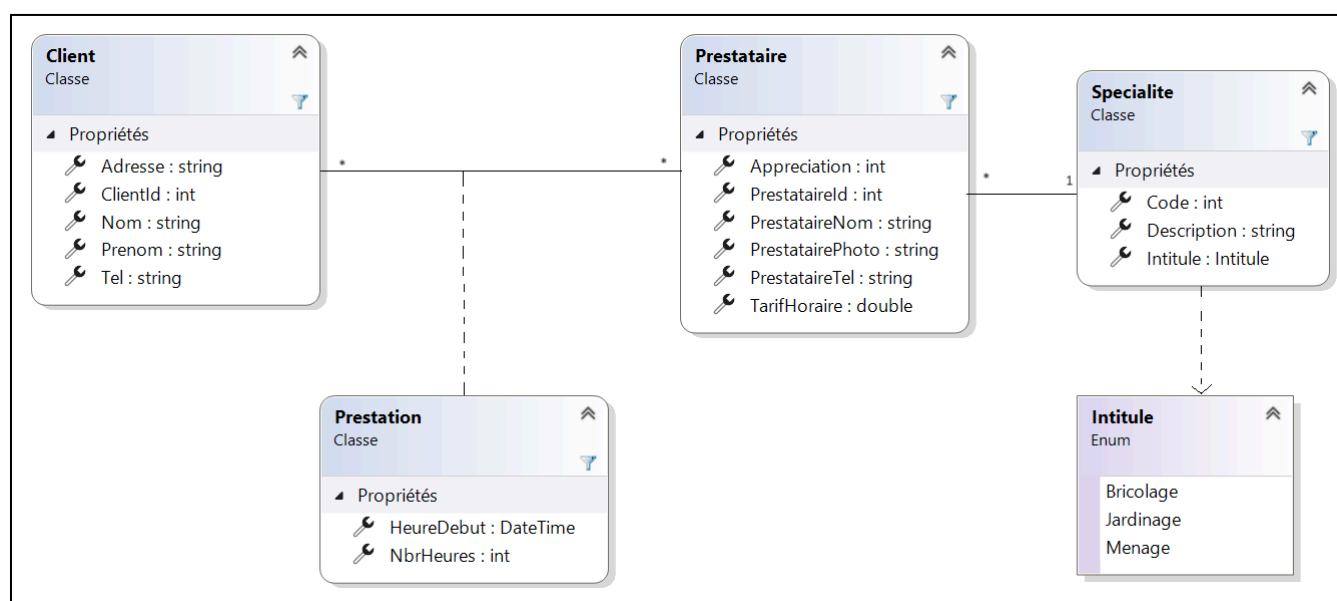
 <b>Se former autrement</b> <b>HONORIS UNITED UNIVERSITIES</b>	<h2 style="margin: 0;">EXAMEN</h2> <p style="margin: 10px 0;">Semestre <input type="checkbox"/> 1      <input checked="" type="checkbox"/> 2</p> <p style="margin: 0;">Session : <input checked="" type="checkbox"/> Principale    <input type="checkbox"/> Rattrapage</p>						
<b>Module :</b> Architecture des systèmes d'information I (.Net) <b>Enseignants :</b> Équipe .Net <b>Classes :</b> 4SIM/INFINI							
<b>Documents autorisés :</b> <input checked="" type="checkbox"/> OUI <input type="checkbox"/> NON <b>Nombre de pages :</b> 4 pages <b>Date :</b> 14/05/2024 <b>Heure :</b> 9h <b>Durée :</b> 1h30m							
<table style="width: 100%; border: none;"> <tr> <td style="width: 60%; border: none;"><b>ETUDIANT(e)</b></td> <td style="width: 40%; border: none;"></td> </tr> <tr> <td style="border: none;">N° Carte : .....</td> <td style="border: none;">Classe : .....</td> </tr> <tr> <td style="border: none;">Nom et Prénom : .....</td> <td style="border: none;">Salle : .....</td> </tr> </table>		<b>ETUDIANT(e)</b>		N° Carte : .....	Classe : .....	Nom et Prénom : .....	Salle : .....
<b>ETUDIANT(e)</b>							
N° Carte : .....	Classe : .....						
Nom et Prénom : .....	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 désire développer une application Web de gestion des prestations de services à domicile (bricolage, jardinage, ménage...). Pour développer cette application, on a réalisé le diagramme de classes simplifié suivant:



## Partie I: Entity-Framework (9 pts)

1. Créer les entités et les relations entre eux en respectant le diagramme de classes. **(2.5 pts)**
2. En utilisant les annotations de données, configurer: **(1.5 pt)**
  - a. la clé primaire de la classe **Specialite**. **(0.25 pt)**
  - b. la clé étrangère de la relation entre **Specialite** et **Prestataire** pour qu'elle soit mappée vers une colonne nommée **SpecialiteFK**. **(0.25 pt)**
  - c. la propriété **Appreciation**, de la classe **Prestataire**, pour qu'elle soit un entier compris entre 0 et 5 avec un message d'erreur lors de violation. **(0.5 pt)**
  - d. la propriété **TarifHoraire**, de la classe **Prestataire**, pour qu'elle soit une valeur monétaire. **(0.25 pt)**
  - e. la propriété **NomPrestataire**, de la classe **Prestataire**, pour qu'elle soit une chaîne de caractères de longueur maximale 20. **(0.25 pt)**
3. En utilisant les configurations de l'API Fluent, configurer: **(2.5 pts)**
  - a. la relation entre **Client**, **Prestataire** et **Prestation** afin que les clés étrangères de cette relation soient mappées vers des colonnes nommées **ClientFK** et **PrestataireFK**. **(1 pt)**
  - b. la clé primaire de la classe **Prestation** afin qu'elle soit composée par les trois propriétés **ClientFK**, **PrestataireFK** et **HeureDebut**. **(0.5 pt)**
  - c. la propriété **HeureDebut** de la classe **Prestation** pour qu'elle soit obligatoire et mappée dans une colonne nommée "HeureRDV". **(1 pt)**
4. Faire le nécessaire pour générer la base de données nommée sous la forme **NomPrenomPrestation**. **(1 pt)**
5. Effectuer les modifications suivantes et mettre à jour la BDD : **(1.5 pt)**
  - a. Créer un type détenu nommé **Coordonnees** contenant les propriétés **Nom**, **Prenom**, **Tel** et **Adresse** et faire les modifications nécessaires dans la classe **Client**. **(1 pt)**
  - b. Mettre à jour la BDD. **(0.5 pt)**

## Partie II: Services (5 pts)

Dans la couche **Service** créer les interfaces et les classes nécessaires pour implémenter les méthodes suivantes permettant de :

1. retourner les 2 meilleurs prestataires de service bricolage. **(1 pt)**
2. retourner la moyenne de tarif horaire d'une spécialité dont le code est passé en paramètre. **(1 pt)**
3. retourner la liste des prestations demandées par un client donné regroupé par intitulé de spécialité. **(1.5 pt)**
4. retourner les gains du prestataire le moins cher pendant l'année en cours. **(1.5 pt)**

## Partie III: ASP MVC (6 pts)

1. Alimenter la BDD par deux spécialités et deux clients. **(0.5 pt)**
2. Réaliser une vue qui permet de créer un prestataire comme le montre la figure suivante, tout en adaptant la vue pour uploader la photo du prestataire et pour choisir sa spécialité à partir d'une liste déroulante. **(2.5 pts)**

## Create

### Prestataire

---

PrestataireNom

PrestatairePhoto  

Choisir un fichier

Aucun fichier choisi

PrestataireTel

Appreciation

TarifHoraire

SpecialiteFK  



1

▼

Create

[Back to List](#)

3. Rediriger vers une vue Index qui liste l'ensemble des 2 meilleurs prestataires de service bricolage. (1.5 pts)

Index						
<a href="#">Create New</a>						
PrestataireNom	PrestatairePhoto	PrestataireTel	Appreciation	TarifHoraire	Specialite	
Prestataire1		12345678	5	25,000 DT	Bricolage	<a href="#">Prestations</a>   <a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Prestataire4		87654321	5	30,000 DT	Bricolage	<a href="#">Prestations</a>   <a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

4. Dans la vue Index, ajouter un lien qui redirige, pour chaque prestataire, vers les prestations du prestataire correspondant. (1.5 pts)

- Vous devez alimenter la base de données par deux prestations d'un prestataire de votre choix .

# Index

[Create New](#)

HeureDebut	NbrHeures	Client	Prestataire	
08/05/2024 12:00:00 AM	2	Nom1	Prestataire1	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
09/05/2024 12:00:00 AM	3	Nom2	Prestataire1	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

Bon travail 😊