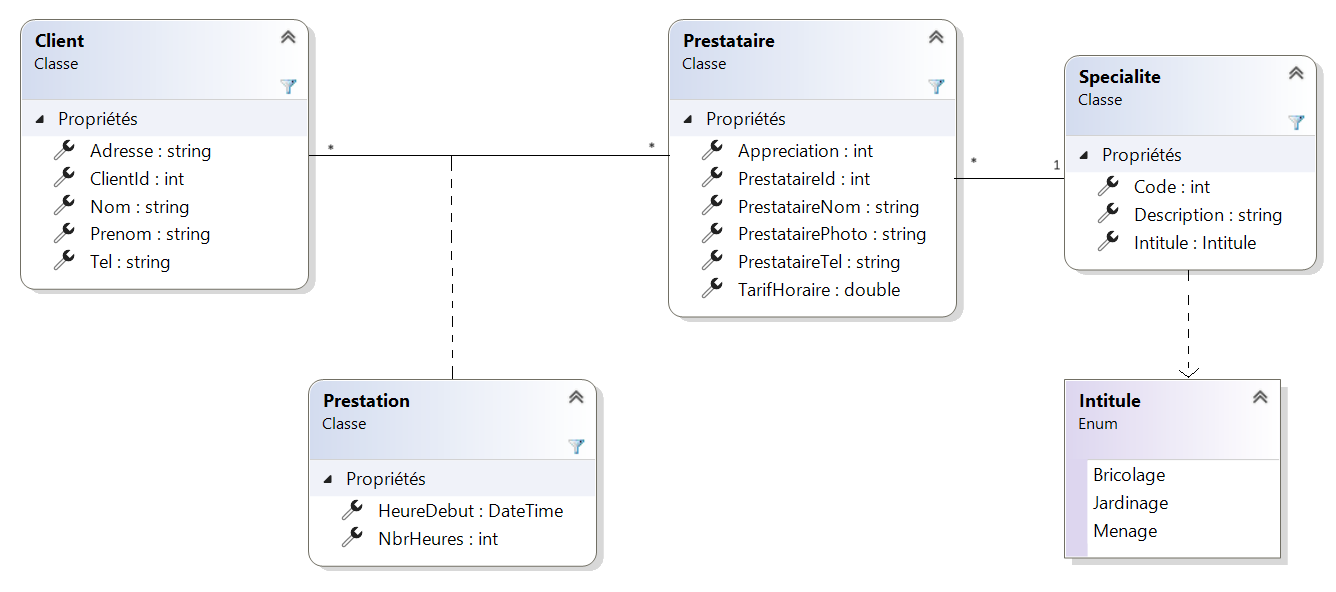
| EXAMEN  **Semestre** : ⃞ 1 🗹 2  **Session** : 🗹 Principale ⃞ Rattrapage | |  |
| --- | --- | --- |
| **Module** : Architecture des systèmes d’information I (.Net)  **Enseignants** : Équipe .Net  **Classes** : | | |
| **Documents** **autorisés**  : 🗹 OUI ⃞ NON **Nombre** **de** **pages** : 4 pages | | |
| **Date** : **Heure** : **Durée** : 1h30m | | |
| Classe : ………………  Salle :……………… | **ETUDIANT(e)**  N° Carte : …………………………………………………………...  Nom et Prénom :……………………………………………… .. | |

**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 pts)**
   1. la clé primaire de la classe **Specialite** . **(0.25 pt)**
   2. la clé étrangère de la relation entre **Specialite** et **Prestataire** pour qu'elle soit mappée vers une colonne nommée **SpecialiteFK. (0.25 pt)**
   3. 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)**
   4. la propriété **TarifHoraire**, de la classe **Prestataire**,pour qu’elle soit une valeur monétaire. **(0.25 pt)**
   5. 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: **(1.5 pts)**
   1. 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)**
   2. la clé primaire de la classe **Prestation** afin qu’elle soit composée par les trois propriétés **ClientFK,**  **PrestataireFK** et **HeureDebut.** **(0.5 pt)**
4. Faire le nécessaire pour générer la base de données nommée sous la forme **NomPrenomPrestation**. **(1 pts)**
5. Effectuer les modifications suivantes et mettre à jour la base :
   1. Créer un type détenu nommé **Coordonnees** contenant les propriétés **Nom,**  **Prenom, Tel et Adresse** et faites les modifications nécessaires dans la classe **Client. (1 pt)**
   2. Mettre à jour la base. **(0.5 pt)**

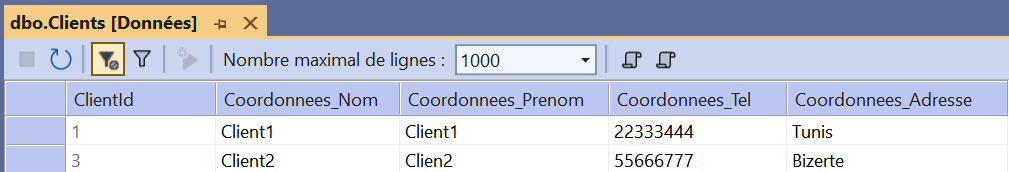
**Partie II: Services (5 pts)**

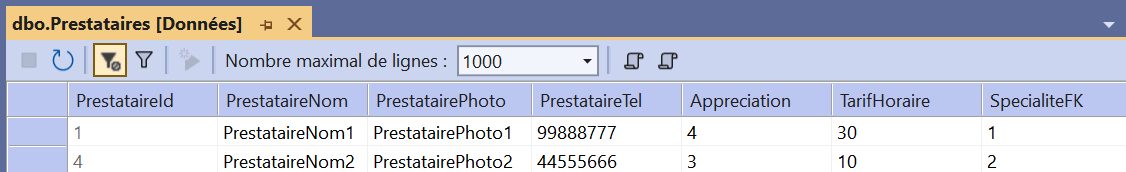
Dans la couche **Service** créer les interfaces et les classes nécessaires pour implémenter les méthodes suivantes permettant de :

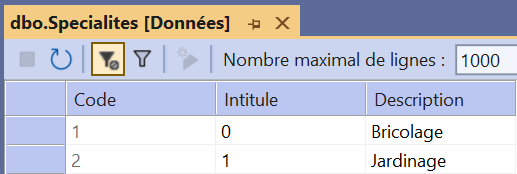
1. retourner les 3 meilleurs prestataires de service bricolage. **(1 pt)**
2. retourner la moyenne de tarif horaire d’une spécialité dont le code est passé en paramètre.**(1 pt)**
3. retourner la liste des prestations demandées par un client donné regroupé par intitulé de spécialité.**(1.5 pts)**
4.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

**Partie III: ASP MVC (6 pts)**

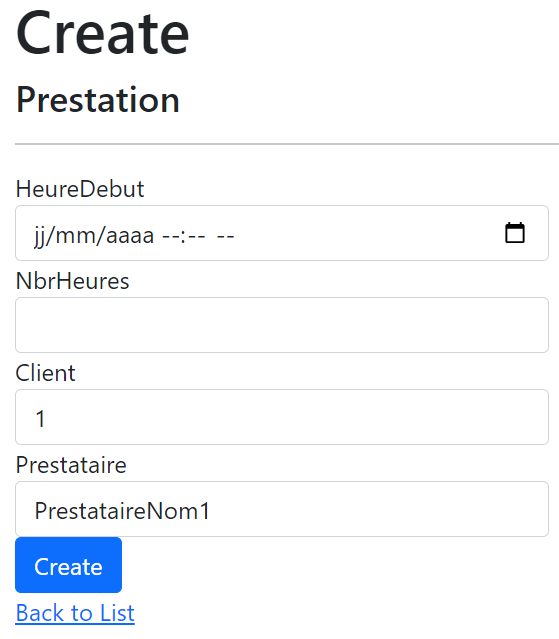
1. Alimenter la BDD par deux prestataires, deux clients et deux spécialités. **(0.5 pt)**

****

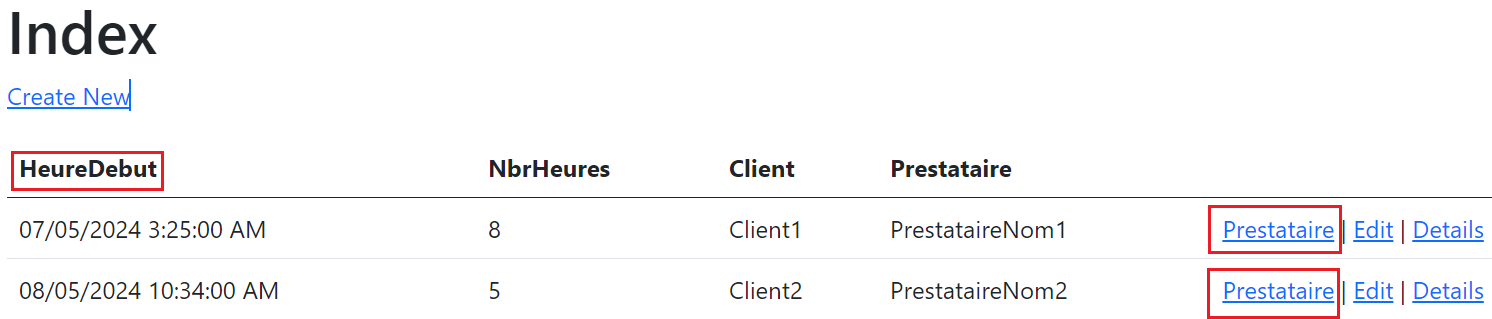
****

****

1. Réaliser une vue qui permet de créer une prestation comme le montre la figure suivante, tout en choisissant le prestataire et le client à partir de deux listes déroulantes. **(2.5 pts)**

****

1. Rediriger vers une vue Index qui liste l’ensemble des prestations ordonnées par date.**(1.5 pts)**
2. Dans la vue Index, ajouter un lien qui redirige, pour chaque prestation, vers les détails du prestataire correspondant. **(1.5 pts)**

****

