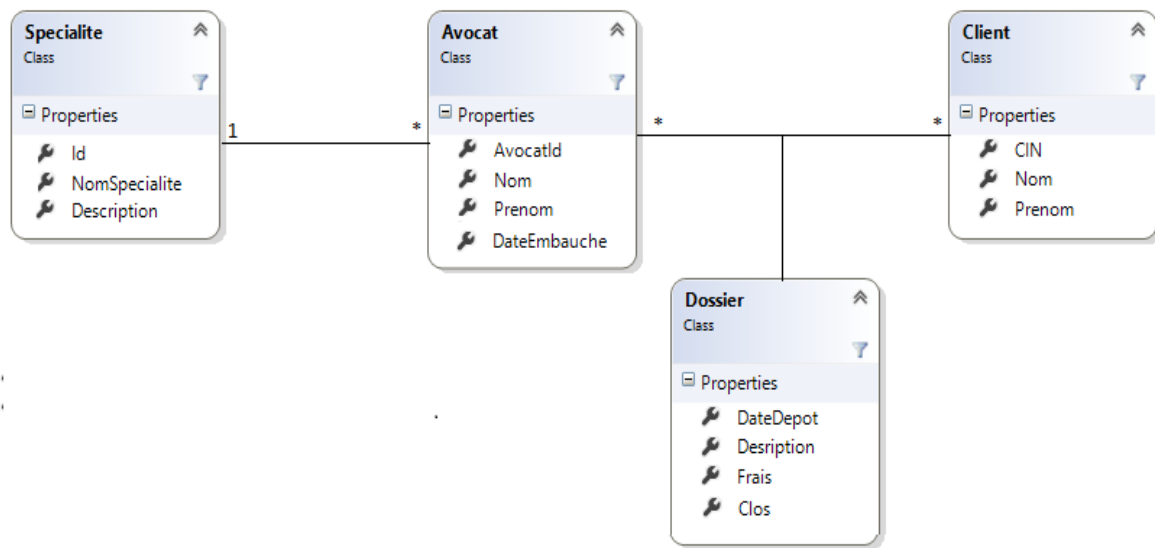


## Enoncé du problème :

On se propose de modéliser une application web qui va gérer les activités d'un certain nombre d'avocat qui partagent un même bureau. Cette application va gérer leurs clients ainsi que les dossiers qu'ils traitent.

Ci-dessous un diagramme de classe qui représente cette situation.



NB : Le champs Clos de la classe Dossier est un champs booléen qui désigne si le dossier a été traité ou si il est encore en cours.

## Partie I: Entity-Framework (7pts)

1. Créer les entités en respectant le diagramme de classes (1.5 pt)

2. En utilisant les annotations: (2.5pts)

- a. Configurer la clé primaire composée de trois de la classe **Dossier** sachant que **DateDépôt** fait partie de cette clé.
- b. Ajouter, là où il faut, une annotation pour que chaque classe ait une clé primaire.
- c. La propriété **Description** de la classe **Dossier** doit être de type Multiligne.
- d. Les propriétés **DateEmbauche** et **DateDepot** doivent être des dates valides.
- e. Les propriétés **Nom** et **Prenom** dans la classe **Avocat** doivent être obligatoires avec un message d'erreur s'ils ne sont pas introduits par l'utilisateur de l'application.

2. Configurer, en utilisant Fluent API, la relation 1..\* entre **Avocat** et **Specialite** en précisant **SpecialiteFK** comme clé étrangère et en désactivant la Cascade On Delete. (2 pt)

3. Créer le contexte et générer la base de données. (1 pts)

### Partie II: Services (6pts)

Dans la couche Service créer les interfaces et les classes nécessaires qui comportent des méthodes permettant de :

- 1- Retourner les clients, d'un avocat donné, qui ont déposé des dossiers dans les 10 derniers jours. (2 pts)
- 2- Retourner la spécialité qui contient le maximum d'avocats. (1 pt)
- 3- Retourner le nombre de Dossiers par spécialité (spécialité de l'avocat qui traite le dossier). (1 pt)
- 4- Implémenter une méthode qui prend en paramètre un dossier et fais une réduction de 30% sur les frais si l'avocat qui le traite a déjà traité plus que 3 dossiers (ayant un état clos) pour le client ayant déposé le dossier. (2 pts)

### Partie III: MVC (7pts)

1. Alimenter manuellement la base de données par 2 spécialités, 2 avocats et 2 clients. (0.5pt)
2. Réaliser une vue qui permet de Créer un dossier comme le montre la figure suivante. (3 pts)

### Creation Dossier

Description	<input type="text" value="Arnaque lors de la vente d'une villa."/>
Frais	<input type="text" value="650"/>
DateDepot	<input type="text" value="18/01/2019"/>
Avocat	<input type="text" value="Sarah Ben Salah"/>
Client	<input type="text" value="Sallami Mostapha"/>
Clos	<input type="checkbox"/>
<input type="button" value="Create"/>	

3. Rediriger vers une vue index qui liste l'ensemble des dossiers en permettant de rechercher les dossiers par nom d'avocat. (2 pts)

### Liste des dossiers

Avocat

[Create New](#)

Description	Frais	Clos	
Demande de divorce.	1000	<input checked="" type="checkbox"/>	<a href="#">Avocat  </a>
Arnaque lors de la vente d'une villa.	650	<input type="checkbox"/>	<a href="#">Avocat  </a>

4. Le lien « Avocat » de la vue Index ci-dessus doit diriger vers une vue détails qui donne les détails relatifs à l'avocat qui traite le dossier. (1.5 pt)

### Avocat

Nom	Ben Salah
Prenom	Sarah
DateEmbauche	03/10/1999