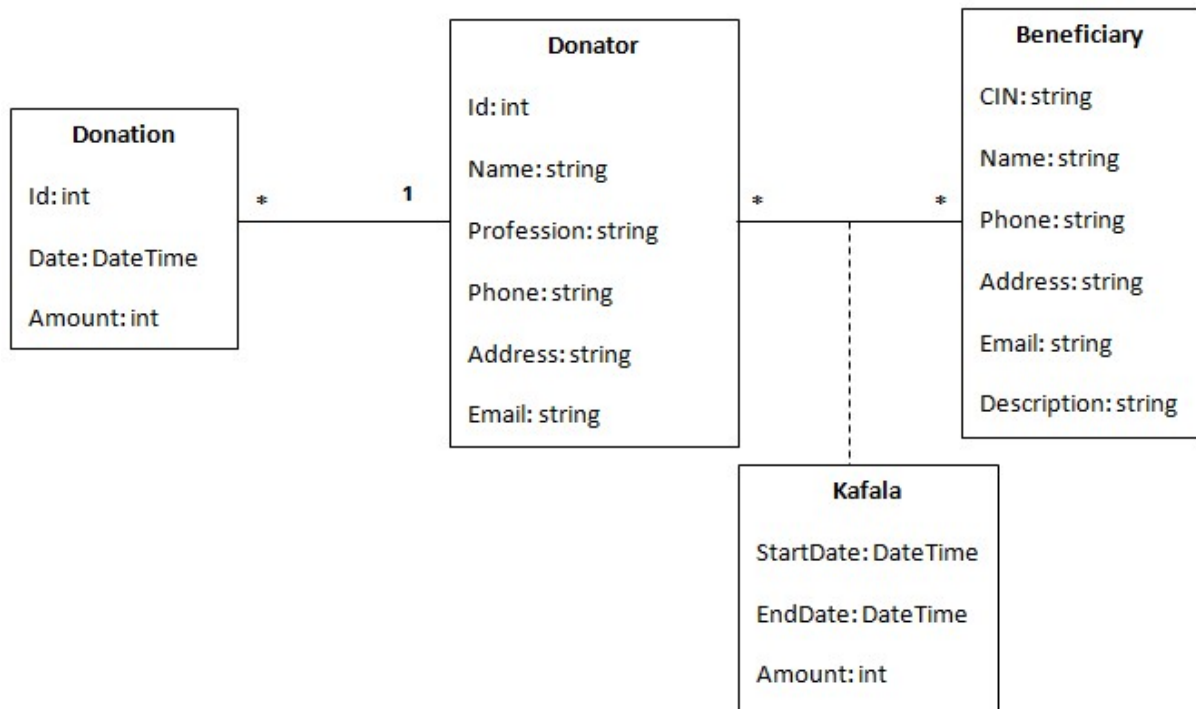
 <p>esprit Se former autrement HONORIS UNITED UNIVERSITIES</p>	<h1>EXAMEN</h1>	
	<p>Semestre : <input type="checkbox"/> 1 <input checked="" type="checkbox"/> 2</p> <p>Session : <input checked="" type="checkbox"/> Principale <input type="checkbox"/> Rattrapage</p>	
<p>Module : Architecture des systèmes d'information I (.Net) Enseignants : Équipe .Net Classes : 4 INFINI & 4 NIDS</p>		
<p>Documents autorisés : <input checked="" type="checkbox"/> OUI <input type="checkbox"/> NON Nombre de pages : 4 pages</p> <p>Date : 17/05/2023 Heure : 9h Durée : 1h30m</p>		
<p>ETUDIANT(e) N° Carte : Nom et Prénom :</p>		<p>Classe : Salle :</p>

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

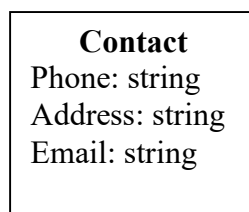
Ci-dessous une partie simplifiée du modèle des données d'une application de gestion des dons pour une association.

Avec : **Kafala** représente un engagement monétaire mensuel d'un donateur pour un bénéficiaire.



Partie I: Entity-Framework (10 pts)

1. Implémenter le diagramme de classes. (3 pts)
2. Utiliser les annotations pour respecter les contraintes suivantes :
 - a. Configurer les propriétés clés primaires là où il le faut. (0.5 pt)
 - b. Les propriétés **Amount** représentent des sommes monétaires. (0.5 pt)
 - c. Les propriétés **Phone** représentent des numéros de téléphone valides. (0.5 pt)
 - d. Les propriétés **Date** et **StartDate** sont obligatoires. (0.5 pt)
 - e. La propriété **CIN** représente un numéro de carte d'identité nationale valide, composé de huit chiffres. (0.5 pt)
3. Dans une classe de configuration, configurer la relation {**Donator**, **Beneficiary**, **Kafala**} sachant que **DonatorFk** et **BeneficiaryFk** sont des clés étrangères. (1.5 pts)
4. Créer la classe **Contact** suivante, faire le nécessaire pour l'utiliser dans le diagramme de classes, configurer cette classe comme un type d'entité détenue et mettre à jour la BDD. (1.5 pts)



5. Utiliser une migration nommée **InitMig** pour générer une base de données nommée **AssociationDB**. (1.5 pts)

Partie II: Services (4 pts)

Implémenter les services suivants :

6. Renvoyer la liste des bénéficiaires profitant d'une « Kafala » jusqu'à l'instant. (1 pt)
7. Renvoyer le total des dons pour une période passée en paramètre. (1 pt)
8. Renvoyer la liste des donateurs qui peuvent être sollicités pour une « Kafala ». Ce sont des donateurs actifs mais qui n'ont aucune « Kafala » en cours. (2 pts)

Partie III: ASP MVC (6 pts)

9. Alimenter la BDD par deux donateurs, deux bénéficiaires et une « Kafala ». (0.5 pt)
10. Implémenter une page **Index** qui permet de lister les donateurs. (1.5 pt)

Index

[Create New](#)

Id	Name	Profession	
1	Rahma	Teacher	Edit Details Delete
2	Mohamed	Engineer	Edit Details Delete

11. Implémenter une page de création d'une « Kafala ». (2 pts)

Create

Kafala

StartDate

EndDate

Amount

DonatorId

BeneficiaryId

[Back to List](#)

12. Dans la page **Index** créée précédemment, ajouter un lien **Kafala** pour rediriger vers la liste des « Kafala » associée à chaque donateur. (2 pts)

Index

[Create New](#)

Id	Name	Profession	
1	Rahma	Teacher	Edit Details Delete Kafala
2	Mohamed	Engineer	Edit Details Delete Kafala

Index

[Create New](#)

StartDate	EndDate	Amount	DonatorId	BeneficiaryId	
01/01/2023 00:00:00		150	1	0000001	Edit Details Delete

Bon travail 😊