

Module : Architecture n-tiers .NET

Enseignants : Syrine KHIARI, Youssef BOUMAIZA, Ghassen SMIDA

Classes : 4 ERP-BI1, 4 ERP-BI2, 4 ERP-BI4

Documents autorisés : OUI ☒ NON ☐

Nombre de pages : 3

Date : 29 Octobre 2014

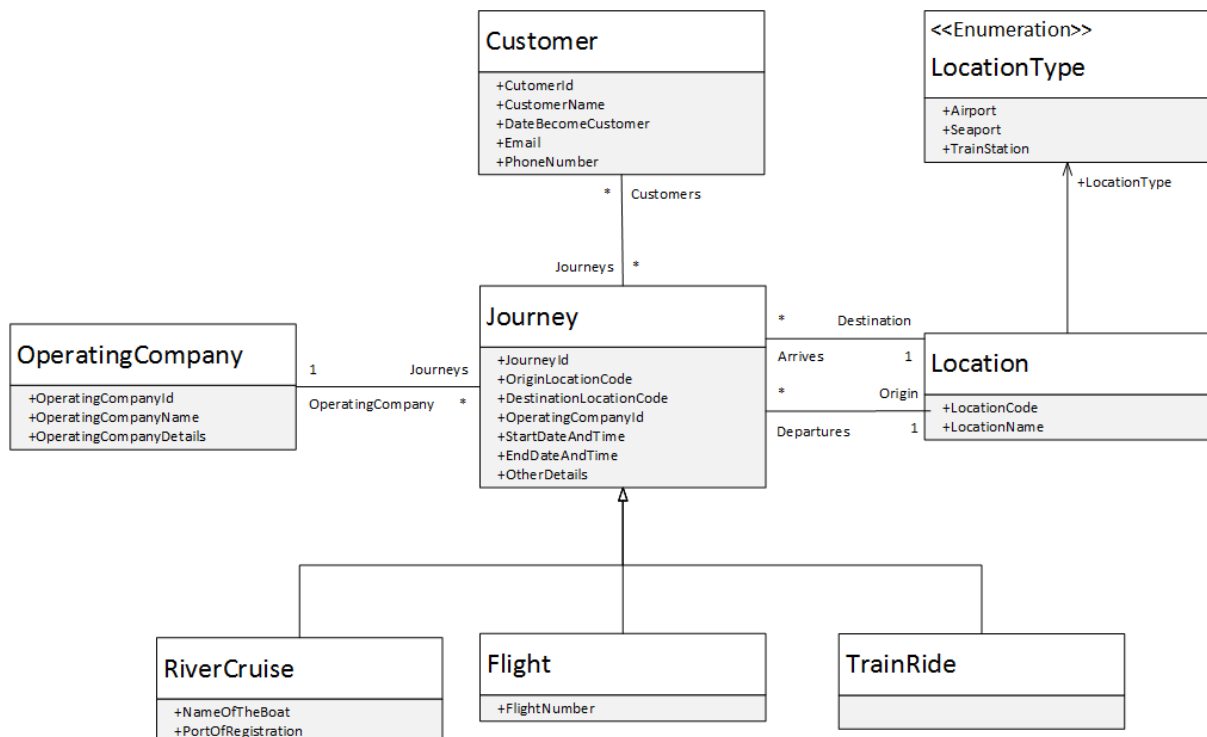
Heure 9h

Durée : 1h30mn

Etude de cas :

On se propose de développer une application web pour une agence de voyage à fin de permettre à sa clientèle d'effectuer des réservations ainsi qu'à l'agence d'analyser les besoins de ses clients.

Soit le diagramme de classes entités suivant qui décrit notre modèle de données :



Partie I : Entity Framework

1. On vous demande de développer l'entité « Journey » dans le dossier Entites du projet « Odyssey.Domain » selon le diagramme de classe fournit, utiliser les Annotations pour assurer les configurations suivantes (**3pt**):
 - a. « StartDateAndTime » : est obligatoire avec un affichage au niveau des vues : « Journey Start ».
 - b. « OtherDetails » : un texte multi lignes avec un affichage de « Other details »
 - c. Ajouter aussi la ou les propriétés de navigation selon le diagramme de classes.
2. Dans le dossier Configurations du projet Odyssey.Data, développer, en utilisant la fluent API, la classe de configuration JourneyConfiguration qui permet :

- a. De configurer la relation Many to Many entre les entités Journey et Customer, spécifier « Travels » comme nom de table d'association. **(1pt)**
 - b. De configurer le mapping entre les entités Journey, TrainRide, RiverCruise et Flight de façon à que ces entités soient mappés à une seule Table nommé «Journeys», la colonne de la base de données qui permet de différencier entre les entités héritées doit être nommée «JourneyType» **(1pt)**
 - c. De configurer les relations Many To One entre l'entité Journey et les entités Location et OperatingCompany. **(1pt)**
3. Dans la classe OdysseyContext qui représente le contexte :
 - a. Ajouter les dbSet nécessaires des entités. pour les nommer choisissez les entités au pluriel. **(1pt)**
 - b. Activer la classe de configuration de la question précédente. **(1pt)**
4. Implémenter les méthodes suivantes dans la classe JourneyServices du projet Odyssey.Service
 - a. `public ICollection<Journey> GetComingJourneys()` qui permet de retourner les prochains Journeys. La propriété `DateTime.Now` retourne la date et l'heure actuelle du système. **(1pt)**
 - b. `ICollection<Journey> GetJourneysByCustomer(int idCustomer)` qui permet de retourner la liste des Journeys par rapport à un Customer donné. **(1pt)**
5. Implémenter la méthode `public ICollection<Location> GetBestDestinations(int top)` qui permet de retourner les meilleures top destinations par rapport à leurs fréquences. **(3pt)**

Partie II : MVC

Toute cette partie se déroulera dans le projet ASP MVC intitulé « Odyssey.Web ».

Dans le contrôleur JourneysController (2.5pt):

1. L'action Index retourne la liste des prochaines Journeys.
2. Ajouter la vue associée à l'action de l'index qui permet d'afficher la liste des journeys. Ensuite la vue index contient un formulaire qui permet la saisie d'un idCustomer dans TextBox et un bouton submit. Le formulaire pointe vers l'action Journeys du contrôleur JourneysController

Index

Top

[Create New](#)

| LocationName | OperatingCompanyName | LocationName | Journey Start | EndDateAnd Time | Other Details |
|------------------------|----------------------|------------------|------------------------|-----------------------|--------------------|
| Tunis Carthage Airport | Airline | Heathrow Airport | 12/31/2014 12:00:00 AM | 1/10/2014 12:00:00 AM | Christmas holidays |

3. L'action Journeys retourne la liste des Journeys par rapport à idCustomer.

Dans le contrôleur LocationsController :

1. Développer l'action BestDestinations la liste du top 10 destinations. (1pt)
2. Ajouter la vue partielle _BestDestinations associée à l'action précédente qui permet l'affichage du résultat du Helper CloudTage des Location retournées par l'action BestDestinations. Cette vue contient un helper CloudTag(string name, string code) qui affiche un ActionLink vers l'action Details du contrôleur LocationsController utiliser le htmlAttribute @style qui donne une valeur aléatoire du font-size la méthode Next(8,32) d'une instance de la classe Random retourne une valeur aléatoire entre 8 et 32 le résultat sera par exemple : @style = "font-size : 10px". (3pt)
3. Ajouter l'affichage de la vue partielle _BestDestinations dans l'index des Locations. (0.5pt)

Grand Central Station Heathrow
Airport