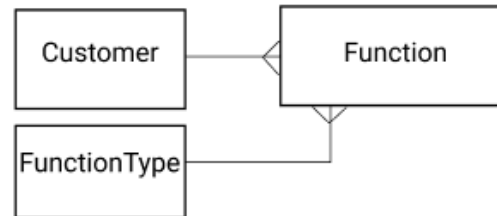


Instructions

Start a new ASP.NET Core MVC application and name it username_Catering_Management, using *your* Blackboard username.

This is a simplified model of a Catering Management solution for a Hotel and Conference Center.

- We will start the database model with 3 entities:
 - Customer
 - Function
 - FunctionType



1. **(20 Marks)** In the Models folder, create classes for the three entities. You will use the Fluent API for some requirements, but use naming conventions and data annotations in the classes to implement the formatting and validation rules as listed below:

Field	Validation	Format/Notes
Customer Class		
ID*		<i>Int Identity PK</i>
FirstName	Required, max length 50	
MiddleName	Max length 30	
LastName	Required, max length 100	
Company Name	Max length 120	
Phone	Required, 10 Digit string	
CustomerCode	Required string, One capital letter followed by 7 numeric digits. UNIQUE	Starts with: "C" for Company clients "G" for Government clients "I" for Internal clients or "A" for all others
FunctionType Class		
ID*		<i>Int Identity PK</i>
Name	Required, max length 120	
Function Class		
ID*		<i>Int Identity PK</i>
Name	String max length 120	
Lobby sign	String max length 60	
Date	Required, DateTime	
DurationDays	Required, Int > 0	Default 1
BaseCharge	Required, Double	
PerPersonCharge	Required, Double	
GuaranteedNumber	Required, Int > 0	Default 1
SOCAN	Required, Double	Default \$50.00
Deposit	Required, Double	
DepositPaid	Boolean	Default False
NoHST	Boolean	Default False
NoGratuity	Boolean	Default False
CustomerID	Required, Int	<i>Foreign Key</i>
FunctionTypeID	Required, Int	<i>Foreign Key</i>

- a. Make sure that you supply user friendly error messages and display names.
2. **(15 Marks)** Create a DbContext named CateringContext and use the Fluent API for the following:
 - a. Add a unique constraint to the CustomerCode property.
 - b. None of the foreign key relationships should use Cascade Delete.
3. Implement a Database Migration and create the Database.*
4. **(10 Marks)** Add code to seed some sample data in each database table.
5. **(15 Marks)** Add controllers and scaffold all CRUD operations for Customer, FunctionType and Function entities.
6. Modify navigation so the user can easily access all your controllers.

Note that the rubric for Part 1A only adds up to 60 marks. The final 40 marks will come from Part 1B.

You will submit your work after both Parts 1A & 1B are complete.

Note that some steps do not have marks assigned. For example, steps such as creating a migration, updating the database, and providing links are so clearly needed that it is expected that you know they are required. Although marks are not assigned, failure to complete them will result in marks being subtracted from your final grade on the evaluation. This is a general principle that will apply to all assignment instructions.