

Titre professionnel : Concepteur Développeur d'application

Evaluation : Développer des composants d'accès aux données avec un ORM.

Compétences évaluées :

- Mettre en place une base de données
- Développer des composants dans le langage d'une base de données
- Développer des composants d'accès aux données

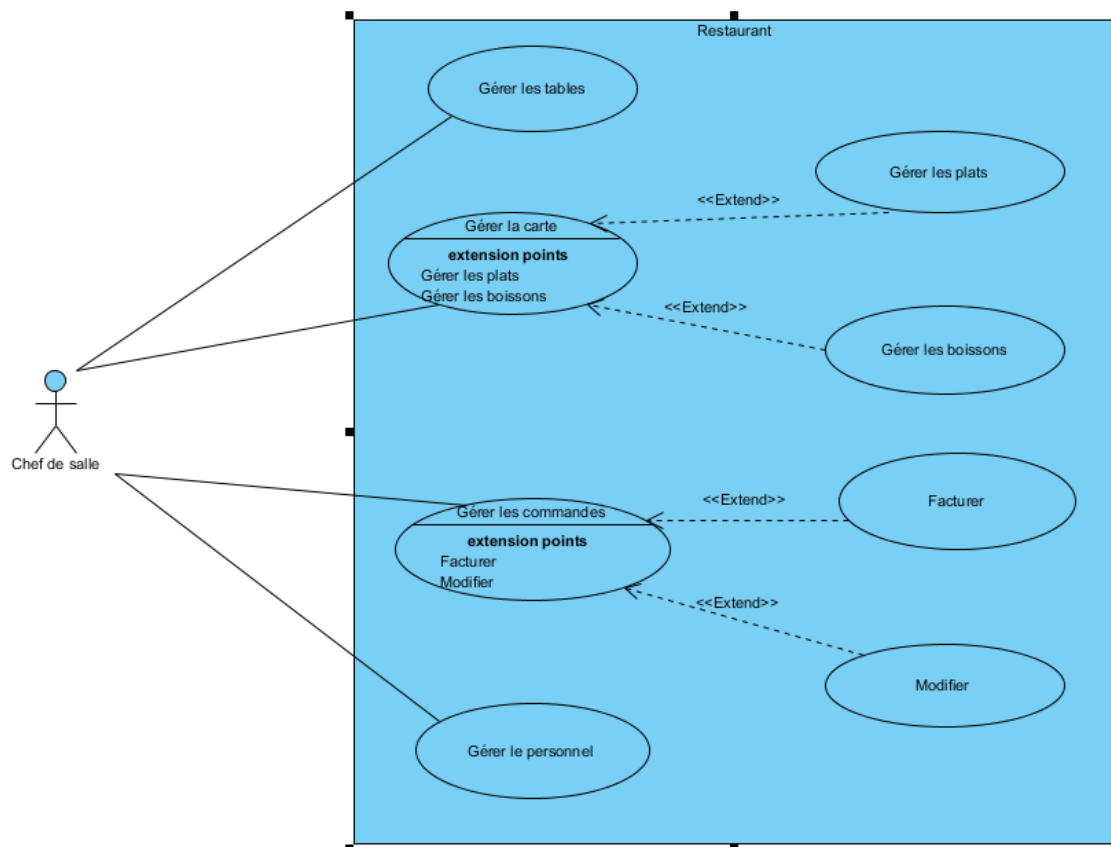
Durée : 3h

Documents personnels autorisés.

Consigne : créer un document word dans lequel vous enregistrerez, les screens de code demandés.

1. Présentation du contexte.

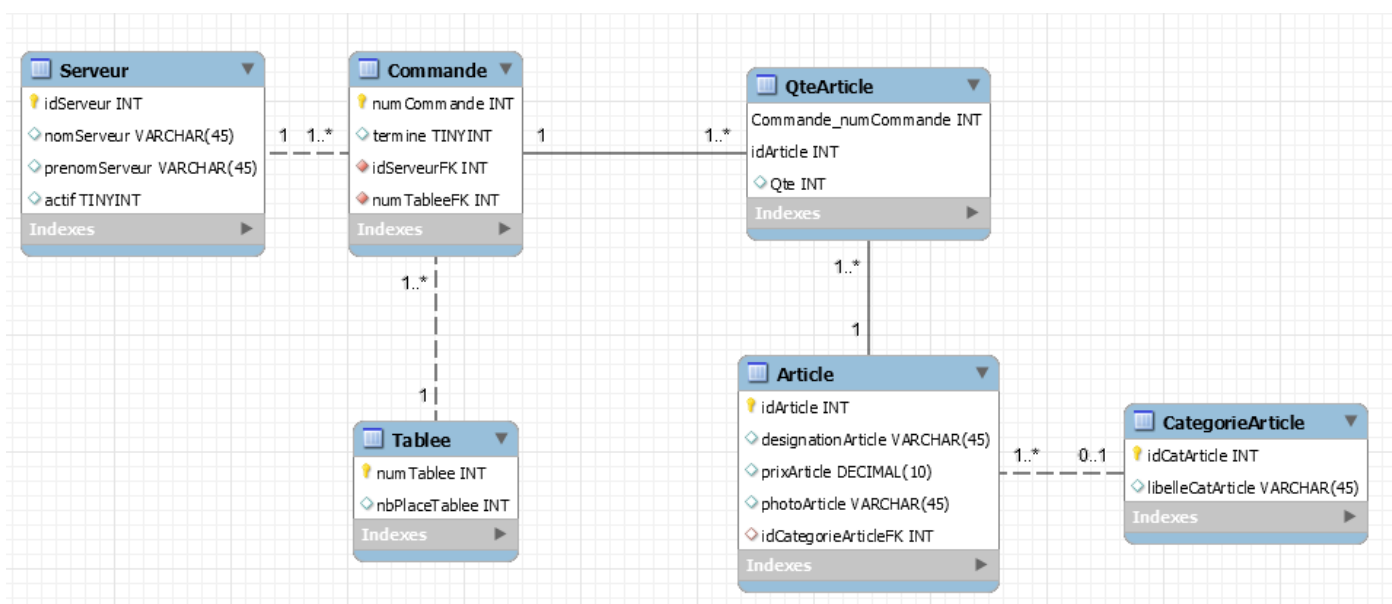
On souhaite développer une application qui doit permettre l'informatisation d'un restaurant. Un TPV (Terminal Point de Vente ou POS en anglais), placé sur le bar, assure la gestion générale du restaurant (gestion des tables, de la carte, du personnel et des commandes).



Contraintes de développement :

- SGBD MySQL ou Microsoft SQL Server
- Ordinateur compatible PC avec système d'exploitation Windows 10
- EDI Visual Studio 2017 ou supérieur
- Langage C#
- ORM (Objet Relational Mapping) : Entity Framework

Schéma conceptuel de la base de données :



- La classe (ou entité) « Serveur » permet de recenser le personnel qui travaille en salle. L'attribut « actif » permet de savoir si la personne est en activité ou pas.
- La classe « Tablee » permet d'enregistrer les tables existantes dans la base de données. Chaque table est caractérisée par un numéro et un nombre maximal de place.
- La classe « catégorieArticle » permet de classer les différents articles par catégorie. Les principales catégories sont : les boissons, les plats, les entrées, les desserts, ...
- La classe « Article » : Un article représente un élément proposé sur la carte indépendamment de sa catégorie. Un article est caractérisé par désignation, un prix, une photo de cet article et une référence à sa catégorie.
- La classe « Commande » : Elle référence le serveur qui a passé la commande et le numéro de la table concernée. Les articles commandés quant à eux sont enregistrés dans la table QteArticle.

Partie 1 : Création la base de données et d'un jeu de tests

- Créer la base de données « tpvrestaurant » puis sa structure à partir du script fourni. (Vous adapterez le script si nécessaire).
- Écrire les scripts permettant d'enregistrer dans la table « CategorieArticle » les tuples suivants :

+ Options				idCatArticle libelleCatArticle	
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	Plat
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	Dessert
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	Entrée
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	Boisson

- Ecrire les script SQL permettant d'enregistrer dans la table « Article » les tuples suivants :

+ Options				idArticle designationArticle prixArticle photoArticle idCategorieArticleFK				
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	Poulet Frites	12	NULL	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	Pizza	13	NULL	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	Coca	3	NULL	4
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	Glace	6	NULL	2
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	Salade	6	NULL	3

- Ecrire les scripts SQL permettant d'enregistrer dans la table « serveur » les tuples suivants :

				idServeur nomServeur prenomServeur actif			
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	NomServeur1	PrenomServeur1	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	NomServeur2	PrenomServeur2	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	NomServeur3	PrenomServeur3	0

- Ecrire les scripts SQL permettant d'enregistrer dans la table « Tablee » les tuples suivants :

				numTablee nbPlaceTablee	
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	4
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	2
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	4
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	2
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	4
<input type="checkbox"/>	Éditer	Copier	Supprimer	6	8

- Ecrire les scripts SQL permettant d'enregistrer 2 commandes.

				Commande_numCommande idArticle Qte			
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	1	2	
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	3	2	
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	4	2	
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	2	2	
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	3	2	
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	5	2	

				numCommande termine idServeurFK numTableeFK			
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	NULL	1	2
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	NULL	2	4

Requêtes MySQL

- Ecrire et tester la requête SQL qui renvoie le nom du serveur chargé du suivi de la commande d'une table.
- Ecrire et tester la requête SQL qui renvoie la désignation de tous les articles d'une Commande.

Écriture d'un TRIGGER (Déclencheur)

- Ecrire et tester un trigger qui permet de supprimer tous les articles commandés lorsqu'on supprime une commande.

Création du projet et mise en place de l'ORM : Entity framework

- Créer un nouveau projet Visual Studio : Application WinForm en langage C#.
- Mapper votre application avec votre base de données.
Pour cela installer les **paquets NuGet** et utiliser l'utilitaire **Scaffold-DbContext** (Voir TP)

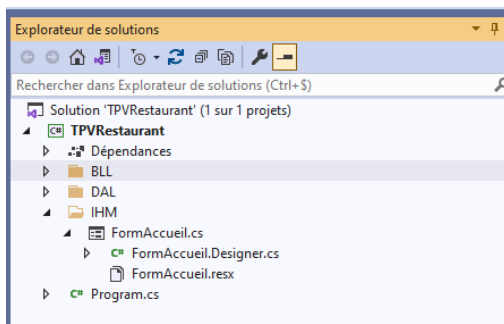
Par exemple : (à adapter en fonction de votre SGBD):

```
Scaffold-DbContext "server=localhost;port=3306;user=root;password=;database=TPVRestaurant"
MySQL.Data.EntityFrameworkCore -OutputDir DAL -f
```

Développement de l'application.

On respectera ici un développement en couche.

- Une couche « DAL » : cette couche permettra d'ajouter des fonctionnalités aux classes déjà générées par EntityFramework.
- Une couche métier : « BLL » et permettant les traitements spécifiques aux objets de l'application.
- Une couche présentation qu'on appellera « IHM ».
- Structurer le projet en créant l'arborescence suivante :



- Déplacer « Form1 » dans IHM et renommer la « FormAccueil »

Si vous n'avez pas réussi le « mappage », demandez un projet « préparé » au formateur afin de pouvoir continuer

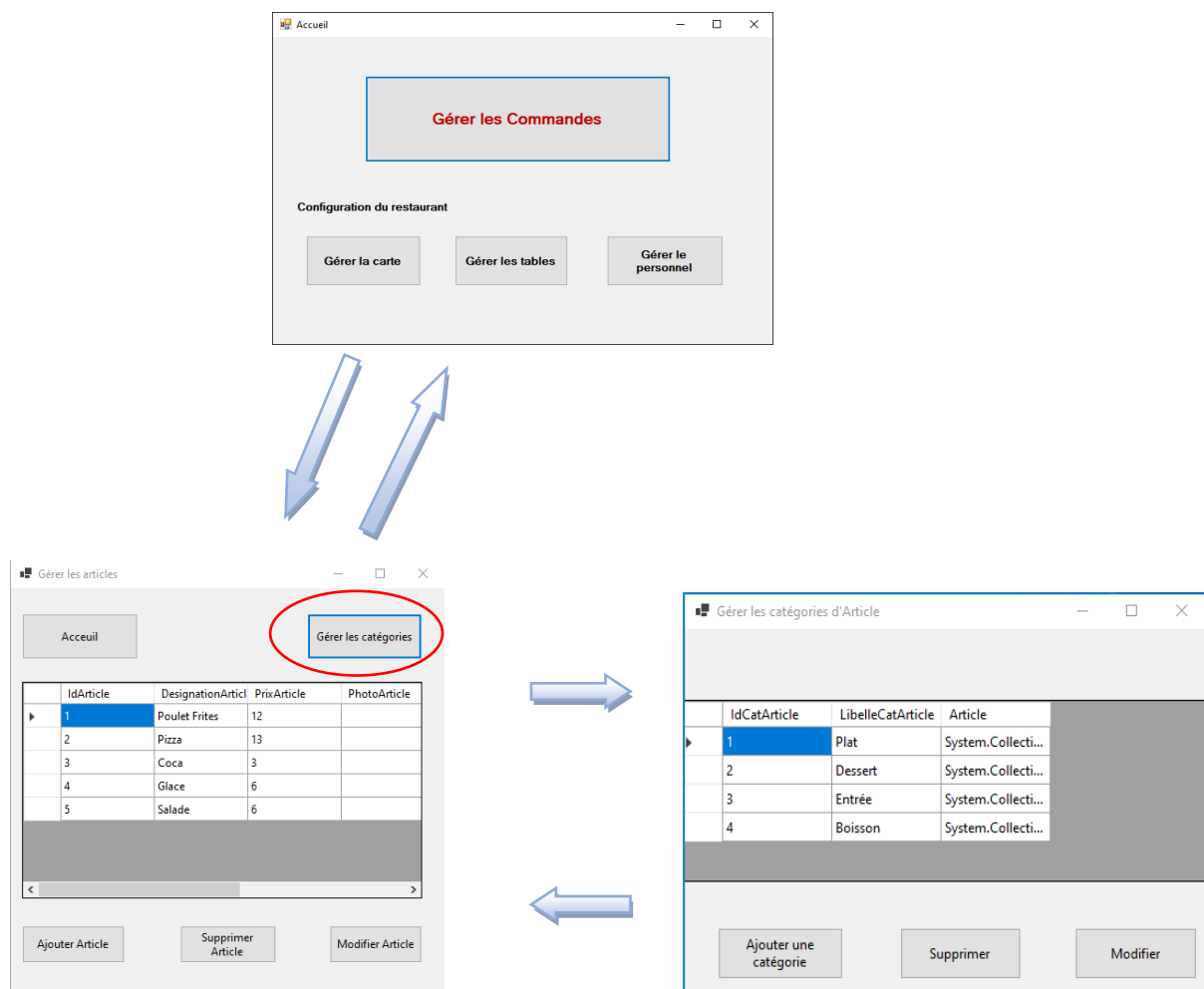
IHM principale

- Réaliser l’IHM d’accueil permettant d’accéder aux fonctionnalités attendues (voir diagramme des cas d’utilisation).
Vous pouvez vous inspirer de cet exemple de réalisation:



On se limitera au cas d’utilisation « Gérer la carte » :

En cliquant sur bouton « Gérer la carte », on doit faire apparaître l’IHM de gestion de la carte et masquer l’IHM d’accueil. En cliquant sur le bouton de retour à l’accueil sur la gestion de la carte, on doit faire apparaître l’IHM d’accueil et fermer l’IHM de gestion de la carte.



- Développer les fonctionnalités **Ajouter, Supprimer et Modifier** pour les Catégories d'article.

Gérer les catégories d'Article

	IdCatArticle	LibelleCatArticle	Article
▶	1	Plat	System.Collecti...
	2	Dessert	System.Collecti...
	3	Entrée	System.Collecti...
	4	Boisson	System.Collecti...

Ajouter une catégorie
Supprimer
Modifier

✚ Pour aller plus loin...

- Développer les fonctionnalités **Ajouter, Supprimer et Modifier** pour les articles.

Gérer les articles

Acceuil
Gérer les catégories

	IdArticle	DesignationArticl	PrixArticle	PhotoArticle
▶	1	Poulet Frites	12	
	2	Pizza	13	
	3	Coca	3	
	4	Glace	6	
	5	Salade	6	

Ajouter Article
Supprimer Article
Modifier Article

Ajouter un article à la carte
Retour
Ajouter un article à la carte
Nom de l'article
Prix :
Lien photo :
Catégorie
Plat
Ajouter

Ajouter un filtre dans la fenêtre article qui permet d’afficher uniquement les articles en fonction de la catégorie choisie.

Compte rendu de test

- Tester et vérifier les fonctionnalités de l’IHM de gestion de la carte.
- Rédiger un compte rendu qui recense tous les éventuels dysfonctionnements.
- Proposer des solutions afin d’améliorer cette fonctionnalité.

FIN