

Présentation du contexte

La filière viti-vinicole en France est présente dans 66 départements. Elle représente, selon les professionnels du secteur, près de 558 000 emplois directs et indirects : vignerons, viticulteurs, négociants, cavistes, fabricants de bouteilles, tonneliers, sommeliers, etc.

La vigne représente 17 % de la richesse agricole nationale, selon l'Insee. Du point de vue des exportations, ce secteur est le premier secteur agroalimentaire excédentaire et le second secteur excédentaire de la France, après l'aéronautique.

La filière viti-vinicole participe aussi à l'attractivité touristique française : dix millions de personnes visitent chaque année musées, domaines et caves, dans différentes régions. Cette filière fait partie du patrimoine culinaire et culturel français ; son importance est majeure dans les métiers de la restauration et de la gastronomie.

La société *VDF SARL* est un des acteurs de cette filière. Cette société commerciale à responsabilité limitée, basée en Bourgogne, s'est spécialisée dans la vente à distance de vins produits dans toutes les régions viticoles de France.

VDF s'est implantée depuis quelques années dans le secteur du commerce en ligne (*e-commerce*) en « Vente de vins et spiritueux ». Son site *Web vinsdefrance.com* connaît une croissance importante de la fréquentation et des ventes, en France comme en Europe.

Elle s'est engagée dans une démarche de qualité grâce à une sélection exigeante des vins, effectuée par son équipe de sommeliers. Elle est en contact direct avec les producteurs (environ 2500 producteurs réguliers) et elle achète des lots de bouteilles de vins sélectionnés auprès de ces producteurs. Elle stocke ces lots dans son entrepôt, et les propose à la vente à sa clientèle, uniquement par le biais de son site *Web*.

Pour l'hébergement de ce site, la société fait appel à un prestataire externe qui lui met à disposition un serveur *Web* et les services associés.

Pour son activité, *VDF* emploie une vingtaine de salariés répartis en plusieurs services : informatique, marketing, pôle vin, graphisme, direction et ressources humaines, comptabilité, client, logistique.

Pour la gestion de son activité, elle utilise un progiciel de gestion intégré (*PGI*) notamment pour la gestion des stocks et de l'entrepôt. L'optimisation logistique participe à la qualité du service de livraison perçue par les clients.

La société a ainsi fidélisé environ 30 000 clients réguliers sur 140 000 clients au total depuis la création de l'entreprise, sur dix ans. Le chiffre d'affaires de l'année antérieure s'est élevé à 12 000 000 €.

Au vu de la croissance de l'entreprise, la société *VDF* prévoit à présent une consolidation de son système d'information, de ses applicatifs et de son site *Web*.

Vous êtes rattaché(e) au service informatique de la société *VDF*, dans une équipe de développement de quatre personnes, sous la direction d'un chef de projet. Votre mission consiste à participer au développement des nouvelles solutions informatiques et au maintien des applications métiers existantes.

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 3 sur 18

Mission 1 – Exploitation et amélioration de la base de données du site Web

Documents à utiliser : 1, 2, 3 et 4

Les pages du site de commerce en ligne *vinsdefrance.com* sont générées dynamiquement à partir des données stockées dans une base de données. Les internautes accèdent à la partie vitrine (*front office*) du site, et les employés habilités de VDF administrent les données du site Web en accédant à l'application arrière-boutique (*back office*) du site, installée chez un hébergeur. Un extrait de la base de données existante est fourni.

En consultant le site, le producteur du vin n° 5429 a constaté des erreurs concernant les cépages (variétés de raisin) et les pourcentages affichés sur le site pour le millésime 2020 de son vin. La fiche du vin indique « CÉPAGE : 85 % Chardonnay, 15 % Merlot » alors que ce millésime est composé de 90 % de cépage Cabernet-sauvignon (n° 108) et 10 % de cépage Merlot (n° 136). Le chef de projet vous demande de préparer les requêtes qui permettront de corriger ces erreurs.

Question 1.1

Écrire les requêtes permettant de corriger ces erreurs.

Le responsable du pôle vin souhaiterait compléter, sur le site vitrine, la rubrique « Culture viticole » à destination des clients, en affichant des informations supplémentaires :

- les noms de tous les cépages utilisés pour les vins « Blanc » ;
- les noms des cépages les plus connus, c'est-à-dire ceux qui entrent dans la composition de plus de 200 vins différents.

Question 1.2

Écrire les requêtes permettant d'obtenir les informations souhaitées.

Le site *vinsdefrance.com* permet actuellement la prise de commande de vins par les clients, mais de nombreuses fonctionnalités supplémentaires sont souhaitées par la direction de l'entreprise : amélioration de la nomenclature des vins, accès aux informations de contact des producteurs, affichage des labels et récompenses d'un vin, mise en place de la gestion des stocks en temps réel et mise en place d'un système d'adhésion au programme dit « privilège » des clients.

Dans le cadre de ces nouvelles fonctionnalités du site, on vous demande de réfléchir aux évolutions nécessaires à réaliser sur la base de données. Pour cela, vous disposez des documents concernant la nomenclature des vins et la description du programme privilège.

Question 1.3

Proposer un schéma de structure de la base de données en complétant l'existant. Seuls les éléments du schéma existant qui sont concernés par l'évolution seront repris dans le schéma proposé.

IMPORTANT : la candidate ou le candidat présentera les évolutions de la structure de la base de données en adoptant le formalisme de son choix (schéma entité-association, diagramme de classes ou encore schéma relationnel).

Avant de s'engager dans une commande, il est important que le client visualise en temps réel sur le site, la quantité de bouteilles disponibles en stock d'un lot. On vous demande de réfléchir à une solution permettant de maintenir cette information à jour dans la base de données.

Question 1.4

Proposer et décrire, sans la réaliser, une solution permettant de maintenir à jour automatiquement la quantité de bouteilles disponible en stock d'un lot.

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 4 sur 18

Mission 2 – Gestion des évènements privilège

Documents à utiliser : 5, 6, 7, 8 et 9

IMPORTANT : la candidate ou le candidat peut choisir de présenter les éléments de code à l'aide du langage de programmation de son choix ou de pseudo-code algorithmique.

La société *VDF* propose des évènements à ses clients privilégiés (*VIP clients*). Il s'agit, au travers de ces évènements, de participer à des expériences uniques : des cours de perfectionnement sur la connaissance des vins, des dîners d'exception avec des chefs étoilés, etc.

Chaque année, la société *VDF* édite un catalogue de ces évènements.

Un client privilégié, désirant s'inscrire à un événement, transmet sa demande à la société *VDF*. Une fois réceptionnée, la demande est vérifiée puis enregistrée. La participation à un événement est facturée au client en fonction du nombre de personnes qui l'accompagnent.

Pour chaque événement, un nombre limité de places est proposé afin de fournir une prestation de haute qualité aux participants.

La société *VDF* souhaite disposer d'indicateurs de performance, comme la mesure de la rentabilité. Un événement est rentable si le nombre de participants est supérieur au nombre minimum d'inscriptions requis, cependant tout événement proposé est maintenu même si sa rentabilité n'est pas assurée.

Une application, conçue avec un tableur, est utilisée pour la gestion de ces évènements.

À l'occasion de cette évolution, il a été décidé de remplacer l'application actuelle par une application C#, orientée objet, associée à une base de données de gestion des évènements.

On vous demande de participer à la création des classes métier qui seront utiles à l'application C#.

Question 2.1

Écrire le constructeur de la classe *ClientVIP*.

Question 2.2

Écrire la méthode *NombreTotalParticipants* de la classe *Evenement*.

Question 2.3

Écrire la méthode *InscrireUnClientVIP* de la classe *Evenement*.

Question 2.4

Écrire la méthode *ObtenirEvenementsRentables* de la classe *Catalogue*.

L'application C# affiche la liste des seuls évènements rentables du catalogue dans une interface graphique. La maquette de cette interface a été créée.

Votre second travail est de terminer la création de la procédure événementielle exécutée lors du chargement de cette interface.

Question 2.5

Compléter le code de la procédure événementielle *ConsulterEvenementsRentables_Load* de l'application C# qui affiche les évènements rentables du catalogue.

IMPORTANT : une méthode statique est appelée en la préfixant par le nom de la classe à laquelle elle appartient : *NomDeLaClasse.NomDeLaMethode()*

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 5 sur 18

Mission 3 – Gestion des coffrets œnologiques

Documents à utiliser : 10, 11, 12, 13 et 14

Dans le but de développer son offre commerciale, la société *VDF* propose un nouveau service à l'ensemble de sa clientèle, privilégiée ou pas : le coffret œnologique. Tous les mois, une sélection de deux bouteilles, sélectionnées par un maître sommelier, est livrée aux clients abonnés à ce service. Trois formules sont proposées :

- le tire-bouchon : un abonnement découverte, constitué de rapports prix-plaisir exceptionnels ;
- la vie en bio : un abonnement constitué d'une sélection de vins biologiques ;
- le décanteur : un abonnement composé des appellations les plus prestigieuses.

Le site marchand de la société *VDF* a été complété avec l'intégration d'un module consacré aux abonnements. Ce module est basé sur une infrastructure logicielle de type *framework* qui intègre une architecture de développement modèle vue contrôleur (*MVC*) et s'appuie sur une base de données spécifique aux abonnements.

Un tableau de bord de suivi des abonnements, sur lequel il vous est demandé de travailler, est en cours de développement dans la partie arrière-boutique (*backoffice*) du module.

Question 3.1

Présenter les atouts et les contraintes d'une architecture modèle vue contrôleur.

Cette infrastructure logicielle intègre un composant de type *ORM* (*Object-Relational Mapping*) qui permet de lier chaque classe de la couche modèle de l'architecture *MVC* à la table correspondante dans la base de données. Ainsi, seuls des objets de l'architecture *MVC* sont utilisés pour les interactions avec les tables de la base de données. Le langage *SQL* d'interrogation et de manipulation de données n'est pas utilisé dans la partie arrière-boutique (*backoffice*) de l'application *Web* et est remplacé par ces objets de l'architecture *MVC Objet*.

Le code *PHP* actuel de la vue *tousLesAbonnementsVue.php*, chargé d'afficher le détail des abonnements, est incomplet car il ne correspond pas à la maquette attendue de l'interface.

Deux informations sont manquantes :

- le titre et le prix de la formule de chaque abonnement ;
- le nombre d'abonnements.

Question 3.2

Apporter les modifications nécessaires au code *PHP* de la vue *tousLesAbonnementsVue.php* afin de l'adapter à la maquette attendue.

IMPORTANT : indiquer sur votre copie les numéros des lignes concernées.

Deux autres informations sont présentes dans la maquette de l'interface et dans le code *PHP* de la vue *tousLesAbonnementsVue.php* :

- le nombre de clients abonnés ;
- le nombre d'abonnements moyen par client abonné.

Cependant, ces informations ne s'affichent actuellement pas dans la vue *tousLesAbonnementsVue.php*. Cette erreur est due au fait que le contrôleur *abonnementContrôleur.php* ne transmet pas à la vue la donnée nécessaire à l'affichage de ces deux informations.

Question 3.3

Apporter les modifications nécessaires au code *PHP* du contrôleur *abonnementContrôleur.php* pour permettre l'affichage de ces deux informations dans la vue *tousLesAbonnementsVue.php*.

IMPORTANT : indiquer sur votre copie les numéros des lignes concernées.

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 6 sur 18

Mission 4 – Audit de sécurité du site Web

Documents à utiliser : 15

Dans le cadre de la sécurisation des applications *Web* de la société *VDF*, vous participez à l'audit de sécurité des systèmes d'authentification des sites de boutique en ligne (*front office*) et d'arrière-boutique (*back office*).

Les clients s'authentifient sur le site <https://vinsdefrance.com> avec leur adresse électronique et leur mot de passe.

Question 4.1

Expliquer ce que la sécurisation du site *Web* avec le protocole *HTTPS* permet de garantir aux clients de la société.

Question 4.2

Expliquer pourquoi l'utilisation de la méthode *GET* dans la page *HTML* d'authentification n'est pas judicieuse. Proposer une correction.

Parmi toutes les failles de sécurité possibles sur le site *Web*, on s'intéresse plus particulièrement à celle de l'injection *SQL* sur la page d'authentification.

La requête utilisée pour l'authentification d'un client dans la page *connect.php* est :
\$reqSQL = " select * from client where courriel = '\$_GET[adr]' and motpasse = '\$_GET[mdp]' ";

Sur la page d'authentification, un utilisateur a saisi uniquement l'information suivante :

Adresse courriel ' or 1=1 --

où -- désigne le début d'un commentaire *SQL*.

Question 4.3

- Donner le texte de la requête qui va être exécutée sur le serveur.
- Expliquer en quoi l'exécution de cette requête présente des risques pour *VDF* et ses clients.
- Proposer une solution pour se protéger contre cette faille d'injection *SQL*.

Pour pouvoir ajouter ou modifier la fiche d'un vin, ajouter un lot en vente ou visualiser les commandes, plusieurs personnes de l'entreprise sont habilitées à gérer les données du site *vinsdefrance.com*. Ces gestionnaires s'authentifient avec leurs identifiant et mot de passe sur l'application d'arrière-boutique installée chez un hébergeur et accessible via l'adresse <https://gestion.vinsdefrance.com>.

Actuellement, l'application d'arrière-boutique utilise le même mécanisme d'authentification que le site boutique mais il semble particulièrement important de sécuriser davantage l'authentification des gestionnaires de l'application d'arrière-boutique.

Question 4.4

Proposer trois mesures techniques à mettre en place dans l'application pour renforcer l'authentification sur le site d'arrière-boutique.

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 7 sur 18

DOCUMENT 1 : Extrait du schéma de données de la base de données existante
--

Le cépage désigne la variété du plant de vigne cultivée. Le millésime désigne l'année de production du vin. Un vin fait l'objet d'un assemblage d'un seul ou de plusieurs cépages selon les appellations. Cet assemblage peut varier selon les millésimes en fonction du choix du producteur dans le respect des contraintes de l'appellation.

Schéma relationnel

Couleur (id, nom)

Clé primaire : id

Cepage (id, nom)

Clé primaire : id

Vin (id, idCou)

Clé primaire : id

Clé étrangère : idCou en référence à id de Couleur

VinMillesime (idVin, millesime)

Clé primaire : idVin, millesime

Clé étrangère : idVin en référence à id de Vin

Composer (idVin, millesime, idCep, pourcentage)

Clé primaire : idVin, millesime, idCep

Clés étrangères : idVin, millesime en référence à idVin, millesime de VinMillesime
idCep en référence à id de Cepage

LotEnVente (id, nbBouteillesInitial, volumeUnitaire, prixUnitaire, nbBouteillesRestant, idVin, millesime)

Clé primaire : id

Clé étrangère : idVin, millesime en référence à idVin, millesime de VinMillesime

Client (id, courriel, motPasse, nom, rue, cp, ville)

Clé primaire : id

Commande (id, dateHeureCommande, idCli)

Clé primaire : id

Clé étrangère : idCli en référence à id de Client

Concerner (idCom, idLot, quantite)

Clé primaire : idCom, idLot

Clés étrangères : idCom en référence à id de Commande
idLot en référence à id de Lot

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 8 sur 18

Schéma conceptuel des données

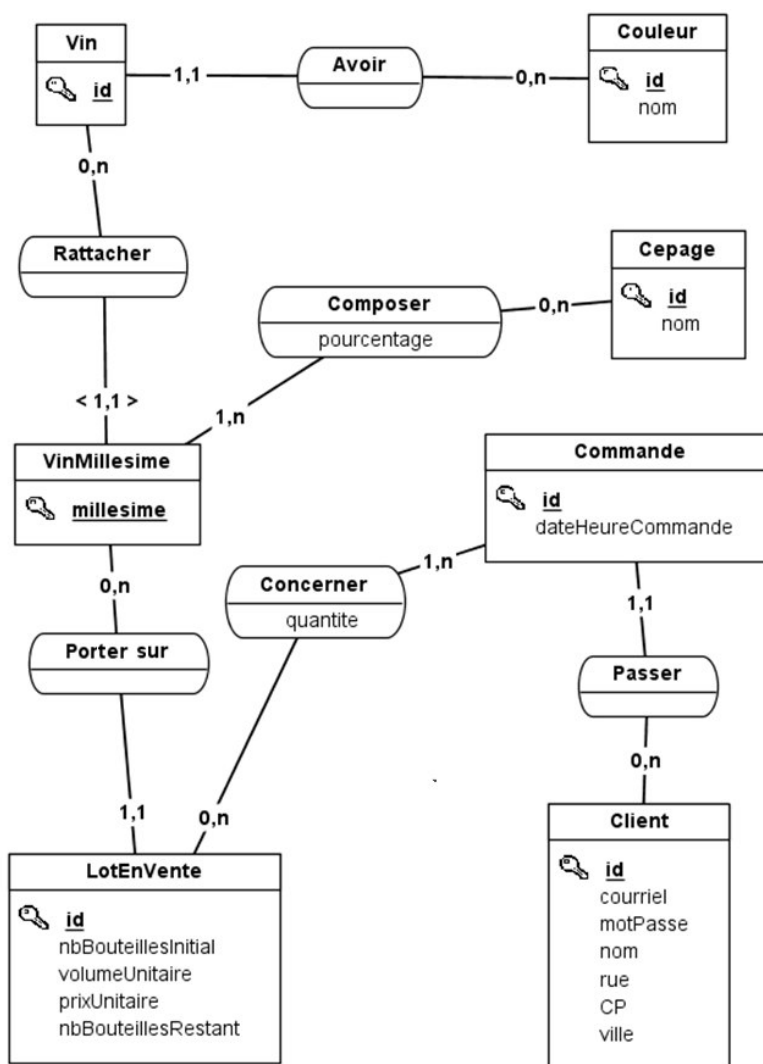
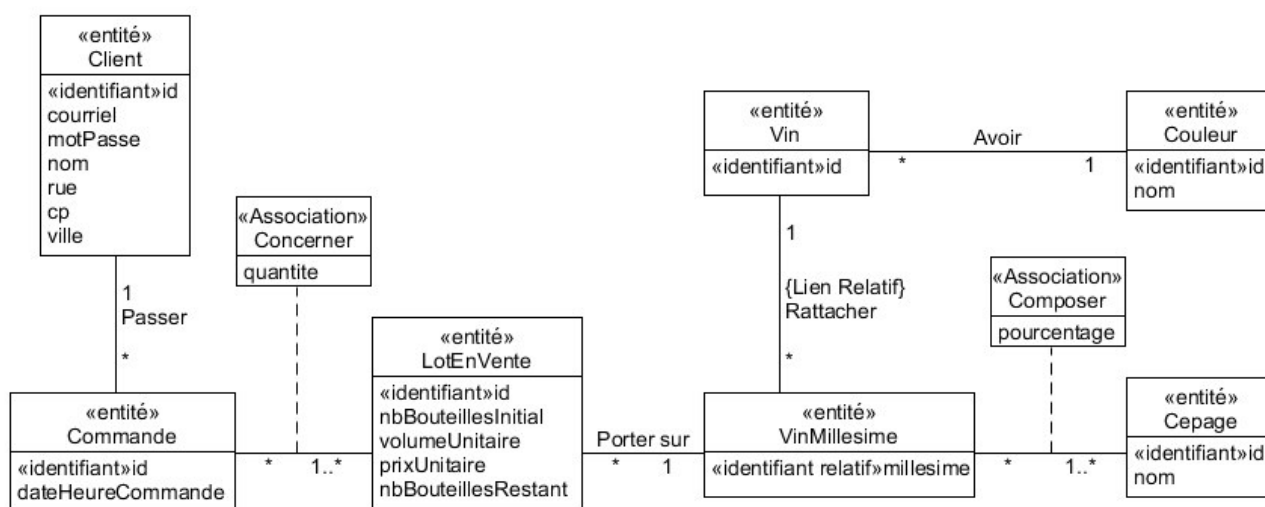


Diagramme de classes



DOCUMENT 2 : Régions et appellations viticoles

Chaque vin appartient à une appellation et chaque appellation est classée dans une des seize régions viticoles. Toutes les appellations existantes, plus d'un millier en France, sont à référencer dans la base de données selon la structure suivante :


RÉGIONS (16)	APPELLATIONS (1313)
1 Alsace	1-1 Alsace, 1-2 Alsace grand cru Altenberg de Bergbieten, ..., 1-53 Alsace Klevener de Heiligenstein, 1-54 Crémant d'Alsace
2 Beaujolais	2-1 Beaujolais, 2-2 Beaujolais Beaujeu, 2-3 Beaujolais Blacé, 2-4 Beaujolais Cercié, 2-5 Beaujolais Chânes, ..., 2-41 Régnié, 2-42 Saint-Amour
3 Bordeaux	3-1 Côtes de Bordeaux, 3-2 Côtes de Bordeaux Blaye, 3-3 Côtes de Bordeaux Cadillac, ..., 3-45 Sainte-Foy-Bordeaux, 3-46 Sauternes
4 Bourgogne	4-1 Aloxe-Corton, 4-2 Aloxe-Corton premier cru, ..., 4-489 Mâcon Verzé, ..., 4-541 Nuits-Saint-Georges, ..., 4-812 Vougeot premier cru Les Petits Vougeots
...	...
16 Sud-Ouest	16-1 Coteaux du Quercy, 16-2 Béarn, 16-3 Bergerac, ..., 16-30 Tursan

DOCUMENT 3 : Fiche vin

L'entreprise *VDF* est en contact direct avec les producteurs. Il faut pouvoir connaître les informations de contact d'un producteur (nom, téléphone, adresse courriel, adresse) en plus des vins qu'il produit.

Chaque vin est présenté sur le site comme l'exemple ci-dessous :

NICOLAS MAILLET · MÂCON VERZÉ Le Chemin Blanc · 2019 · BLANC






RÉGION : Bourgogne
APPELLATION : Mâcon Verzé
MENTION : Le Chemin Blanc
MILLÉSIME : 2019

COULEUR : Blanc
CÉPAGE : 100% Chardonnay
ALCOOL : 13.5 %

PRÉSENTATION DU MILLÉSIME

Le nez fin et complexe dévoile des arômes de pêches blanches, de poires, et de fleurs d'acacia. La bouche est puissante, avec une belle acidité et une longueur en bouche impressionnante. Un vin vif, éclatant, tranchant : un grand vin de gastronomie, à long potentiel de garde.

LABELS



RÉCOMPENSES

- Guide Bettane & Desseauve - 2021 - Evaluation : Sélectionné
- Guide Les Meilleurs Vins de France - 2021 - Evaluation : 2 étoiles

Le millésime d'un vin correspond à son année de production. Par exemple la fiche présentée ci-dessus référence le vin blanc *Mâcon Verzé de mention "Le Chemin Blanc"* produit en 2019.

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 10 sur 18

Pour un même vin, plusieurs millésimes sont commercialisés et certaines informations peuvent varier selon les millésimes, comme le degré d'alcool, les labels et les récompenses obtenues.

Les labels permettent de certifier qu'un vin respecte un cahier des charges de production. On souhaite afficher les logos des labels et, lors du clic sur un des logos, afficher son libellé ainsi qu'une explication précise. L'attribution des labels peut évoluer selon les millésimes d'un vin.

Les récompenses sont importantes dans le métier du vin pour valoriser un produit. Une quinzaine d'organismes (code et libellé) attribuent chaque année des récompenses. Cela donne lieu à une évaluation textuelle du vin : « Médaille d'or », « 16.5/20 », « 2 étoiles », etc.

Un organisme ne peut pas récompenser plusieurs fois le même millésime d'un vin.

Pour la commercialisation, la société *VDF* fait l'achat d'un lot de bouteilles auprès du producteur concerné. Un lot concerne toujours un seul millésime. Ce lot sera stocké dans l'entrepôt et fera l'objet d'une mise en vente sur le site *Web*.

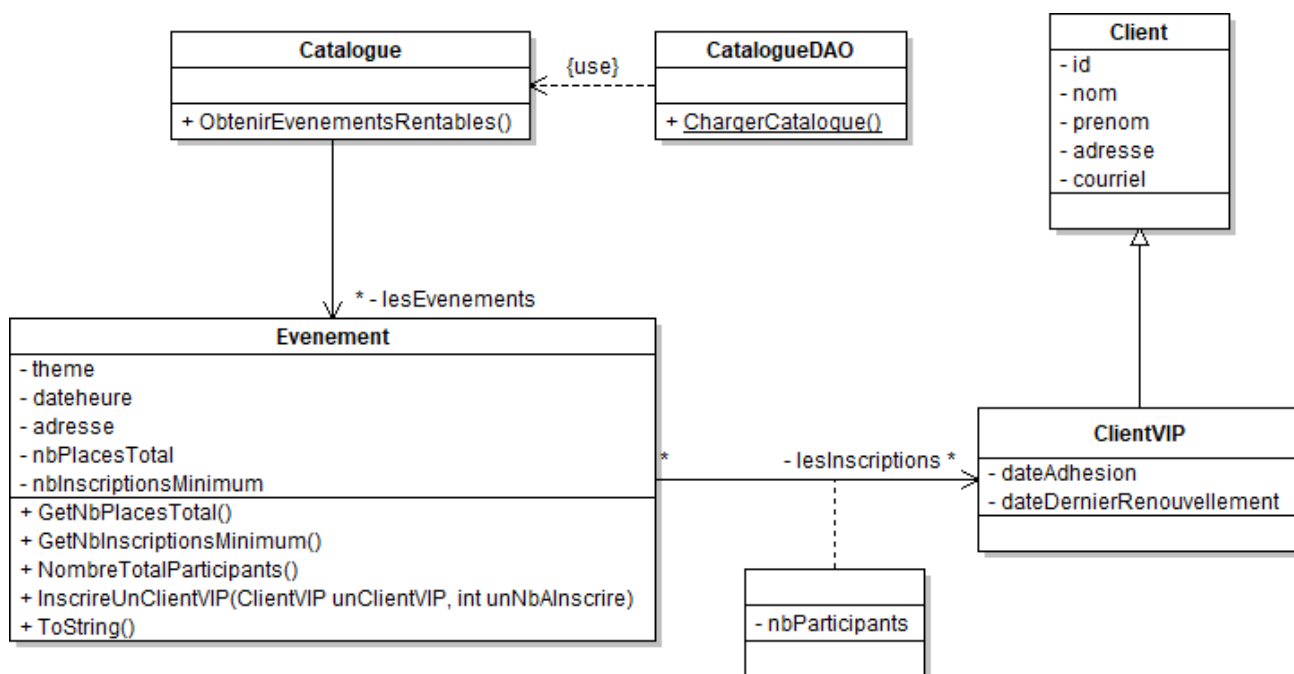
DOCUMENT 4 : Programme privilège des clients

Certains clients peuvent adhérer à un programme de fidélité en payant une adhésion annuelle et devenir ainsi des clients privilégiés (clients *VIP*) de *VDF*.

Pour ces clients privilégiés on devra mémoriser la date de première adhésion et la date du dernier renouvellement d'adhésion.

Ces clients privilégiés peuvent également sélectionner sur leur profil trois régions viticoles de prédilection. Ces informations pourront servir, par exemple, à leur envoyer des alertes ciblées lors de la mise en vente de nouveaux lots.

DOCUMENT 5 : Diagramme de classes correspondant à la gestion des évènements privilège



DOCUMENT 6 : Description textuelle des classes métier

La classe CatalogueDAO contient des méthodes d'accès à la base de données de gestion des événements. Cette classe est associée à la classe Catalogue.

Les méthodes de la classe CatalogueDAO sont en lien avec des objets métiers de la classe Catalogue et avec la base de données.

```
public class Catalogue
{
    // Attribut privé

    private List<Evenement> lesEvenements;

    // Constructeur qui instancie la collection lesEvenements

    public Catalogue ()
    { ... }

    // Méthode qui retourne la collection des événements rentables du catalogue. Un événement
    // est
    // rentable lorsque le nombre de participants est supérieur au nombre minimum d'inscriptions
    // défini par la société VDF.

    public List<Evenement> ObtenirEvenementsRentables()
    {
        // À compléter sur votre copie
    }
} // Fin classe Catalogue

public class CatalogueDAO
{
    // Méthode qui instancie et retourne un objet catalogue. Cet objet contient
    // la collection renseignée de tous les événements présents dans la base de données.
    // Méthode statique, à portée de classe.

    public static Catalogue ChargerCatalogue()
    { ... }

} // Fin classe CatalogueDAO
```

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 12 sur 18

```

public class Evenement
{
    // Attributs privés
    private string theme;           // cours de découverte d'un vin, dîner,...
    private DateTime dateheure;
    private string adresse;        // lieu où se déroule l'évènement
    private int nbPlacesTotal;      // nombre total de places proposées pour l'évènement
    private int nbInscriptionsMinimum; // nombre d'inscriptions minimum pour
                                    // que l'évènement soit rentable.

    private Dictionary<ClientVIP, int> lesInscriptions; // dictionnaire contenant le nombre de
                                                         // participants de chaque client privilégié
                                                         // - en clé, un objet de la classe ClientVIP
                                                         // - en valeur, le nombre de participants (y compris le client
    privilégié).

    // Constructeur qui initialise les attributs privés et instancie le dictionnaire.
    public Evenement(string unTheme, DateTime uneDateHeure, string uneAdresse,
        int unNbPlacesTotal, int unNbInscriptionsMini){ ... }

    public int GetNbPlacesTotal()
    {
        return nbPlacesTotal;
    }

    public int GetNbInscriptionsMinimum()
    {
        return nbInscriptionsMinimum;
    }

    // Méthode qui calcule le nombre de participants à l'évènement
    public int NombreTotalParticipants()
    {
        // À compléter sur votre copie
    }

    // Méthode qui inscrit un client privilégié à l'évènement si le nombre de places disponibles est
    // suffisant.
    // Renvoie TRUE si l'inscription est réalisée, et FALSE si le nombre de places disponibles est
    // insuffisant ou si le client est déjà inscrit.
    // Le paramètre unNbAInscrire contient le nombre de personnes à inscrire
    // (y compris le client privilégié).

    public boolean InscrireUnClientVIP(ClientVIP unClientVIP, int unNbAInscrire)
    {
        // À compléter sur votre copie
    }

    // Méthode redéfinie de la classe Object. Elle retourne une chaîne de caractères qui décrit
    // l'objet courant.

    public override string ToString()
    { ... }
} // Fin classe Evenement

```

```

public class Client
{
    // Attributs privés

    private string id;
    private string nom;
    private string prenom;
    private string adresse;
    private string courriel;

    // Constructeur qui initialise les attributs privés.

    public Client(string unId, string unNom, string unPrenom, string uneAdresse, string
unCourriel)
    { ... }

} // Fin classe Client

public class ClientVIP : Client           // La classe ClientVIP hérite de la classe Client
{                                         // Cette classe sert à représenter les clients privilégiés

    // Attributs privés

    private DateTime dateAdhesion; // date d'adhésion au programme « privilège »
    private DateTime dateDernierRenouvellement; // date du dernier renouvellement d'adhésion

    // Constructeur

    public ClientVIP( // À compléter sur votre copie )
    {
        // À compléter sur votre copie
    }

} // Fin classe ClientVIP

```

DOCUMENT 7 : Maquette de l'interface graphique de consultation des événements rentables

L'interface affiche dans une grille de données (identifiant *grdEvenementsRentables*) tous les événements rentables du catalogue.

Évènements rentables
11 Avril - Initiation œnologie - Château de Besseuil - 25 inscriptions
09 Mai - Dîner Fête des saveurs - Château de Chevignes - 32 inscriptions
15 Mai - Perfectionnement œnologie - Château de Chaintré - 15 inscriptions

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 14 sur 18

DOCUMENT 8 : Extrait du code C# de la procédure événementielle de consultation des événements rentables *ConsulterEvenementsRentables_Load*

// La procédure événementielle ConsulterEvenementsRentables_Load affiche dans une grille de données la liste des seuls événements rentables du catalogue.

Catalogue leCatalogue;

List<Evenements> laListeEvenementsRentables;

// À compléter sur votre copie

// L'objet leCatalogue est chargé à partir de la base de données.

// La collection des événements rentables est chargée à partir du catalogue.

// La collection des événements rentables est liée avec une grille de données afin d'afficher son contenu automatiquement.

grdEvenementsRentables.DataSource = laListeEvenementsRentables;

DOCUMENT 9 : Exemple d'utilisation d'une collection et d'un dictionnaire

COLLECTION : en C#, la collection de type List<T> permet d'enregistrer des éléments de type T où T peut être n'importe quel type d'éléments (int, string, nom de classe, etc).

L'exemple ci-dessous permet de manipuler une collection d'entiers.

Le principe est le même, quel que soit le type des éléments.

List<int> mesNombres; *// déclaration d'une collection d'entiers*

mesNombres = new List<int>(); *// instantiation de la collection*

mesNombres.Add(250); *// ajout d'un entier à la collection*

mesNombres.Count(); *// retourne le nombre d'éléments présents dans la collection*

// parcours de la collection

foreach (int unNombre in mesNombres)

{

Console.WriteLine(unNombre); *// affichage de chaque élément*

}

mesNombres.RemoveAt(1); *// suppression du 2e élément (situé à l'indice 1)*

Console.WriteLine(mesNombres[0]); *// affichage du 1er élément (situé à l'indice 0)*

DICTIONNAIRE : un dictionnaire est un ensemble d'éléments (clé, valeur) permettant d'extraire une valeur (de type TypeValeur) à partir de sa clé (de type TypeClé). À une clé présente dans le dictionnaire correspond une seule valeur. Les types TypeValeur et TypeClé peuvent être n'importe quel type d'éléments (int, string, nom de classe, etc).

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 15 sur 18

L'exemple ci-dessous permet de manipuler un dictionnaire

Le principe est le même, quel que soit le type des éléments.

Dictionary< TypeClé, TypeValeur > monDico; // déclaration d'un dictionnaire composé
// d'éléments (clé, valeur) de type (TypeClé, TypeValeur)

monDico = new Dictionary< TypeClé, TypeValeur >(); // instanciation du dictionnaire
monDico.Add(uneClé, uneValeur); // ajout d'un élément (uneClé, uneValeur) au dictionnaire

```
foreach (var unElement in monDico)      // parcours de tous les éléments du dictionnaire
{
    Console.WriteLine(unElement.Key)      // affiche la clé de l'élément en cours,
                                           // de type TypeClé
    Console.WriteLine(unElement.Value)      // affiche la valeur de l'élément en cours,
                                           // de type TypeValeur
}
```

monDico[uneClé] // désigne la valeur associée à la clé uneClé, de type TypeValeur

monDico.ContainsKey(uneClé) // renvoie true si le dictionnaire contient
// un élément correspondant à la clé spécifiée

DOCUMENT 10 : Extrait du schéma relationnel concerné par les coffrets œnologiques
--

Formule (id, titre, prix)

Clé primaire : id

Client (id, nom, adresse, courriel)

Clé primaire : id

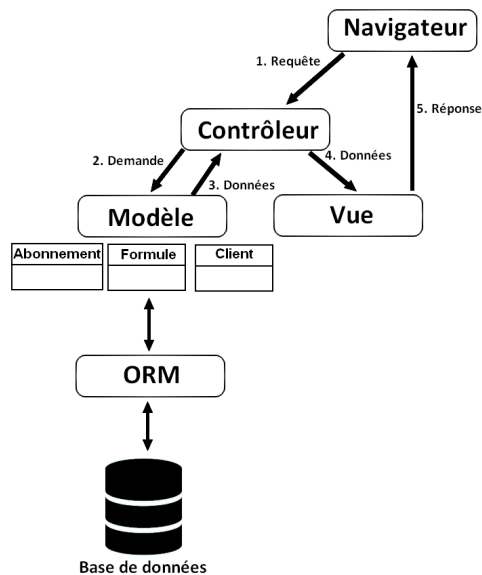
Abonnement (id, dateDebut, dateFin, idCli, idFle)

Clé primaire : id

Clés étrangères : idCli en référence à id de Client
idFle en référence à id de Formule

DOCUMENT 11 : Architecture de développement MVC et technologie ORM

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 16 sur 18



DOCUMENT 12 : Extrait du code *PHP* des classes modèles *formuleModele.php*, *clientModele.php* et *abonnementModele.php*

model.php

```

abstract class Model
{
    public static function all()
    {
        // retourne toutes les données de la table associée
    }
}

```

formuleModele.php

```

class FormuleModele extends Model
{
    // Lien entre la classe modèle FormuleModele
    // et la table Formule de la base de données.

    protected $table = 'Formule';
}

```

clientModele.php

```

class ClientModele extends Model
{
    // Lien entre la classe modèle ClientModele
    // et la table Client de la base de données.

    protected $table = 'Client';
}

```

abonnementModele.php

```

class AbonnementModele extends Model
{
    // Lien entre la classe modèle AbonnementModele et la table Abonnement de la base de
    // données.
    protected $table = 'Abonnement';

    // Méthode d'accès aux données de la formule à partir de l'objet Abonnement courant.
    // Lien réalisé à l'aide de la classe modèle de la formule et du nom de la clé étrangère idFle.
    public function laFormule()
    {
        return $this->belongsTo('App\FormuleModele','idFle');
    }

    // Méthode d'accès aux données du client à partir de l'objet Abonnement courant.
    // Lien réalisé à l'aide de la classe modèle du client et du nom de la clé étrangère idCli.
    public function leClient()
    {
        return $this->belongsTo('App\ClientModele','idCli');
    }
}

```

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 18 sur 18

Extrait de la liste des abonnements
...
Du 01-03-2021 au 01-08-2021 : Client Decrion, Impasse du pré fleuri - Tournus => Formule Le Tire Bouchon (24.9 €)
Du 01-03-2021 au 01-07-2021 : Client Garric, Route de Bram - La Salle => Formule La vie en Bio (34.9 €)
Du 01-04-2021 au 01-06-2021 : Client Felix, Impasse Beau Site - Cluny => Formule Le Tire Bouchon (24.9 €)
Du 01-04-2021 au 01-10-2021 : Client Decrion, Impasse du pré fleuri - Tournus => Formule Le Décanteur (44.9 €)
Du 01-05-2021 au 01-09-2021 : Client Segaud, Plateau de la Iye - Verzé => Formule Le Décanteur (44.9 €)
Du 01-07-2021 au 01-12-2021 : Client Renard, Place de Torpes - Saint Laurent => Formule La vie en Bio (34.9 €)
Du 01-08-2021 au 01-12-2021 : Client Felix, Impasse Beau Site - Cluny => Formule Le Décanteur (44.9 €)
...
Nombre d'abonnements : 2700 Nombre de clients abonnés : 2000 Nombre d'abonnements moyen par client abonné : 1.4

Code de la vue *tousLesAbonnementsVue.php*

1. // La vue *tousLesAbonnementsVue.php* est appelée à partir de la méthode
2. // *tousLesAbonnements* du contrôleur *abonnementContrôleur.php*
3. <h1>Extrait de la liste des abonnements</h1>
4. <?php
5. foreach(\$listeAbonnements as \$uneligne)
6. {
7. \$periode = "Du ". \$uneligne->dateDebut . " au " . \$uneligne->dateFin;
8. \$infosClient = \$uneligne->leClient()->nom . ", " . \$uneligne->leClient()->adresse;
9. echo "
" . \$periode . " : Client " . \$infosClient . "
";
10. echo " => Formule";
11. }
12. echo "
Nombre de clients abonnés : " . \$nbClients ;
13. echo "
Nombre d'abonnements moyen par abonné : " . \$nbAbonnements/\$nbClients;
14. ?>

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2021
U5 : Production et fourniture de services informatiques	Durée : 4 heures
Code sujet : SI5SLAM	Page 19 sur 18

DOCUMENT 14 : Extrait du code *PHP* du contrôleur *abonnementControleur.php*

```
1. // Référence aux classes modèles en lien avec les tables de la base de données.
2. use App\FormuleModele;
3. use App\ClientModele;
4. use App\AbonnementModele;

5. class AbonnementControleur extends Controller
6. {
7. // La méthode tousLesAbonnements() est appelée lors du chargement du tableau de bord
8. // de suivi des abonnements aux formules.


9. public function tousLesAbonnements()
10. {

11. // La méthode all associée à la classe modèle AbonnementModele retourne une liste de
12. // toutes les données de la table Abonnement et des tables associées.
13. $listeAbonnements = AbonnementModele::all();

14. // La méthode count retourne le nombre d'éléments de la liste $listeAbonnements.
15. $nbAbonnements = $listeAbonnements->count();

16. // Appel de la vue tousLesAbonnementsVue.php avec la transmission des données
17. // $listeAbonnements et $nbAbonnements.
18. $lavue = 'tousLesAbonnementsVue';
19. return view($lavue,['listeAbonnements'=>$listeAbonnements, 'nbAbonnements'
    =>$nbAbonnements]);
20. }
21. }
```

DOCUMENT 15 : Page d'authentification du site vitrine *vinsdefrance.com*

	<p><!-- Extrait du code html de la page d'authentification signin.php --></p> <p>S'identifier</p> <pre><form method="GET" action="connect.php"> Adresse courriel <input type="text" name="adr" /> Mot de passe <input type="password" name="mdp" /> <input type="submit" name="btn" value="S'identifier" /> </form></pre>
---	---