

UTBM - Département Informatique

Projet – BD40

Sujet n.3 : Cave à vin



TABLE DES MATIERES

<u>1</u>	<u>HISTORIQUE DU DOCUMENT.....</u>	<u>1</u>
<u>2</u>	<u>DESCRIPTION DU SUJET DE PROJET.....</u>	<u>2</u>
2.1	CONTEXTE DU PROJET.....	2
2.2	DOMAINE ETUDIE	2
2.3	FONCTIONNEMENT DU SYSTEME	3
2.4	ACTEURS ET FLUX	4
2.5	PERIMETRE DU SYSTEME	4
2.6	EXTENSIONS POSSIBLES.....	4
<u>3</u>	<u>DOSSIER DE CONCEPTION DU PROJET.....</u>	<u>5</u>
3.1	MODELISATION CONCEPTUELLE.....	5
3.1.1	DICIONNAIRE DE DONNEES	5
3.1.2	MODELE ENTITE ASSOCIATION.....	7
3.1.3	MODELES DE COMMUNICATION ET DE TRAITEMENTS	11
<u>4</u>	<u>LA MODELISATION LOGIQUE</u>	<u>12</u>
<u>5</u>	<u>LA MODELISATION DE L'IHM</u>	<u>15</u>
<u>6</u>	<u>DOSSIER DE DEVELOPPEMENT DU PROJET</u>	<u>20</u>
6.1	PRESENTATION DE L'INTERFACE DEVELOPEE AVEC MS/ACCESS	20
6.1.1	PAGE PRINCIPALE	20
6.1.2	PAGE DES VINS CONNUS	21
6.2	PRESENTATION DES REQUETES DEVELOPEES.....	22
6.3	PRESENTATION DU CODE VBA DEVELOPPE DANS L'APPLICATION	25
6.3.1	MODULES VBA	25
6.3.2	VBA POUR FORMULAIRES.....	27
6.3.3	CODE VIA DAO	27
6.4	PRESENTATION DES OPTIONS DE SECURITE	28
6.4.1	MISE EN PLACE RUBAN PERSONNALISE	28
6.4.2	LE VERROUILLAGE DE L'ACCES AUX DONNEES ET LA REACTUALISATION DES LIENS	31
6.4.3	CONTROLE DES SAISIES.....	34
6.4.4	CONCLUSION SUR LA SECURITE	34
<u>7</u>	<u>CONCLUSION SUR LE PROJET</u>	<u>35</u>
<u>8</u>	<u>SOURCES</u>	<u>36</u>

1 HISTORIQUE DU DOCUMENT

<i>Version</i>	<i>Date</i>	<i>Auteur</i>	<i>Modifications du document</i>
<i>Etape 1</i>	09/10/2018	Anaïs J.	Création du rapport, page de garde, table des matières et mise en page
	13/10/2018	Anaïs J.	Création d'un schéma « résumé » pour la partie explicative du projet
	21/10/2018	Weitao	Rédaction du contenu 2.3 et 2.4
	21/10/2018	Haoyun	Rédaction du contenu 2.1 et 2.2
	21/10/2018	Anaïs J.	Création du MCD (entité-relation)
	21/10/2018	Weitao	Création du modèle de communication (Diagramme de flux)
	21/10/2018	Haoyun et Anaïs	Modifications et correction des parties 2.1, 2.2
	22/10/2018	Weitao	Modification du contenu 2.3 et 2.4 et rédaction du 2.5
	22/10/2018	Anaïs J.	Ajout de la partie 2.6 et modification du MCD
	22/10/2018	Weitao	Création du MCT et modification du MCC
	22/10/2018	Haoyun	Création du dictionnaire
	26/10/2018	Anaïs J.	Correction du MCD
	05/11/2018	Weitao	Modification du MCT et MCC
	07/11/2018	Anaïs J.	Dernières modifications du MCD, du dictionnaire et des parties avant rendu
<i>Etape 2</i>	23/11/2018	Anaïs J.	Correction MCD, orientation des pages et tri du dictionnaire
	04/11/2018	Weitao	Ajout de la partie Normalisation
<i>Etape 3</i>	04/11/2018	Anaïs J.	Ajout et rédaction de la partie IHM
	28/12/2018	Anaïs J.	Rédaction des options de sécurité
	10/01/2019	Anaïs J.	Fin de la rédaction de toute la partie Dossier de développement
	11/01/19	Anaïs J.	Correction + conclusion
	12/01/19	Anaïs J.	Ajout de la partie traitant de la liaison des tables

Anaïs JARNO

Haoyun LIAO

Weitao YAN

2 DESCRIPTION DU SUJET DE PROJET

2.1 CONTEXTE DU PROJET

Nous sommes des étudiants en première année d'informatique à l'UTBM. Dans le cadre de l'UV de BD40 concernant la conception de systèmes d'informations, nous avons à réaliser un projet pour mettre en application les connaissances acquises durant les cours et les séances de travaux pratiques. Les membres de notre groupe étaient très intéressés et curieux de découvrir le sujet du vin. Le groupe étant composé de deux chinois, nous pensions que ce serait une bonne occasion pour approfondir un sujet très lié à la culture française.

C'est ainsi que nous avons donc choisi le sujet de la cave à vin.

2.2 DOMAINE ETUDIE

Nous avons choisi de restreindre le domaine à une utilisation personnelle.

Le principe est de créer une application de base de données permettant la gestion de cave à vin personnelle. Elle devra ainsi permettre à son utilisateur de visualiser le contenu de sa cave, les informations concernant ses vins et bouteilles, et d'ajouter ou de retirer des bouteilles de sa cave.

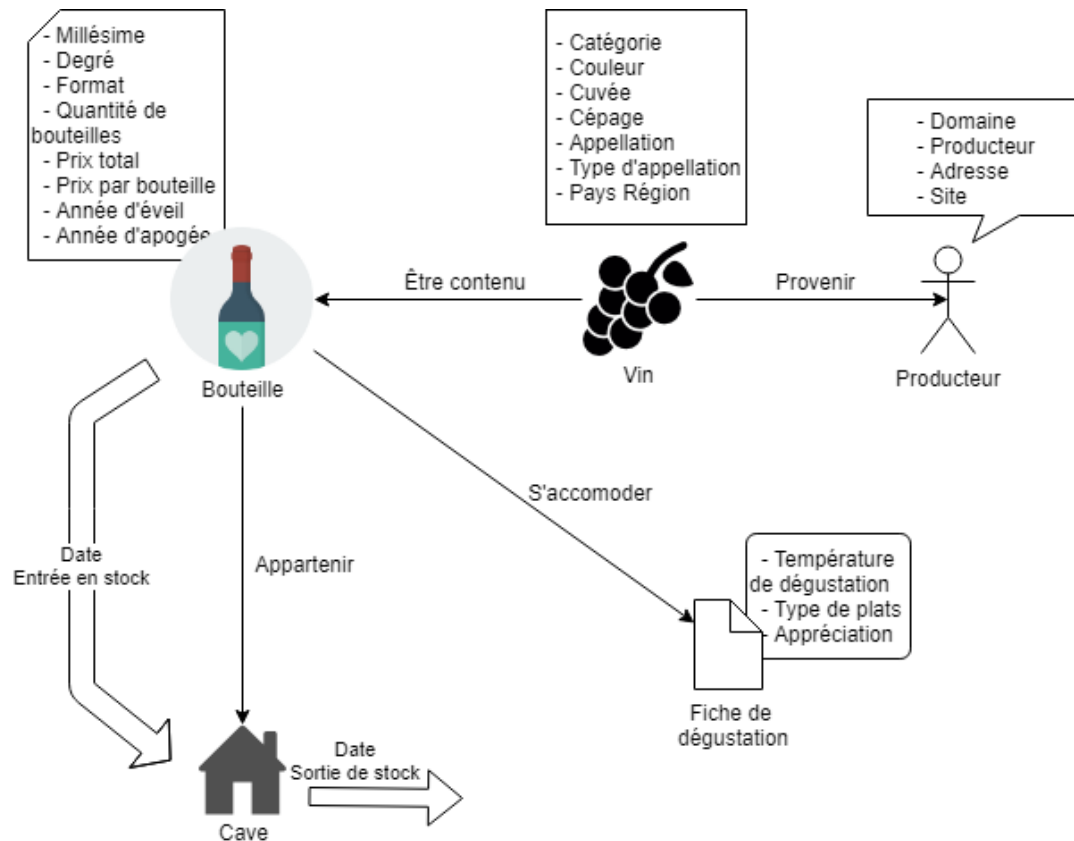
Nous étudions donc les diverses fonctions nécessaires à l'application, telles que la conservation du vin, l'enregistrement de la date d'achat, de la meilleure date de consommation (l'apogée) etc.

Nous avons alors réalisé un schéma servant à résumer les caractéristiques principales de notre gestion de cave :

Anaïs JARNO

Haoyun LIAO

Weitao YAN



2.3 FONCTIONNEMENT DU SYSTEME

Ce système d'information s'adresse à la gestion d'une cave à vin personnelle. On distingue le vin et la bouteille. Les termes '*fournisseur*' et '*producteur*' indiqueront le même acteur.

Des fournisseurs vont produire des vins de différentes régions et ayant d'autres caractéristiques comme la couleur, le cépage, l'appellation, etc.

Le client peut acheter plusieurs bouteilles à la fois, ou une seule. La cave à vin sert à stocker des bouteilles achetées par le consommateur. Les bouteilles sont stockées dans la cave jusqu'au jour où le consommateur veut la déguster.

Dès que la bouteille est sortie de la cave pour être dégustée, celle-ci est retirée de la liste des bouteilles, ou sa quantité est réduite. Une date d'entrée est également notée dès que la bouteille entre en stock.

On a également une fiche de dégustation qui permet d'enregistrer toutes les informations concernant la dégustation pour que le moment venu le consommateur puisse choisir le vin le plus adapté à son désir, à son repas...

Anaïs JARNO

Haoyun LIAO

Weitao YAN

2.4 ACTEURS ET FLUX

Les acteurs externes	Les acteurs internes	Le flux primaire	Les flux secondaires
Producteur de vin	<ul style="list-style-type: none"> ▪ Le consommateur ▪ La cave 	Les entrées-sorties de bouteilles dans la cave	<ul style="list-style-type: none"> ▪ Le stockage du vin (information enregistrée) ▪ La dégustation du vin ▪ L'achat du vin

2.5 PERIMETRE DU SYSTEME

Notre système est très simple. C'est une cave à vin personnelle. C'est-à-dire la cave sert à stocker des bouteilles pour un foyer ou pour une personne.

De temps en temps, l'utilisateur achètera une bouteille ou plusieurs bouteilles, qu'il fera alors entrer dans sa cave.

Puis, il invitera des amis, des parents ou des collègues chez lui et ils pourront alors déguster l'un des vins contenus dans la cave. Par la suite, il pourra alors remplir une fiche de dégustation. Cela permet de noter les avis après dégustation pour que l'utilisateur puisse choisir la prochaine fois son vin préféré, ou le plus adapté à la situation.

2.6 EXTENSIONS POSSIBLES

Les extensions possibles du projet, c'est-à-dire les options que l'on ferait dans un contexte plus professionnel et non scolaire, pourrait être du type :

- Système de facture lié à l'enregistrement d'une arrivée de stock
- Revente de bouteilles
- Commande directe depuis l'application pour un vin spécifique vers le producteur concerné et gestion de la livraison

Anaïs JARNO

Haoyun LIAO

Weitao YAN

3 DOSSIER DE CONCEPTION DU PROJET

3.1 MODELISATION CONCEPTUELLE

3.1.1 Dictionnaire de données

Nom Conceptuel	Nom Logique ou Alias	Type (E, Ca, Co)	Nature + Longueur	Type Win'Design	Identifiant
Adresse du domaine	ADR_DOMAINE	Co	AN234	A(234)	
— Numéro de rue	NUM_RUE_DOMAINE	E	N4	AV(4)	
— Type de rue	TYPE_RUE_DOMAINE	E	A20	AV(20)	
— Nom de rue	NOM_RUE_DOMAINE	E	A40	AV(40)	
— Complément d'adresse	COMPL_ADR_DOMAINE	E	A40	AV(40)	
— Ville	VILLE_DOMAINE	E	A50	AV(50)	
— Code postal	CP_DOMAINE	E	A7	AV(7)	
— Pays	PAYS_DOMAINE	E	A50	AV(50)	
— Continent	CONTINENT_DOMAINE	E	A20	AV(20)	
Année	ANNEE_MILLESIME	E	N4	NS(4)	OUI
Année apogée	ANNEE_APOGEE	E	N4	NS(4)	
Année d'éveil	ANNEE_EVEIL	E	N4	NS(4)	
Catégorie	CATEG_VIN	E	A25	AV(25)	
Catégorie de plats	CATEGORIE_PLATS	E	A50	AV(50)	
Code pays	CODE_PAYS	E	A2	AV(2)	OUI
Couleur du vin	COULEUR_VIN	E	A20	AV(20)	
Cuvée	CUVEE_VIN	E	A30	AV(30)	
Date commande	DATE_COMMANDE	E	AN8	D(8)	
Degré	DEGRE_BOUTEILLE	E	N2	NS(2)	
Nom appellation	NOM_APPELLATION	E	A30	AV(30)	
Nom cave	NOM_CAVE	E	A25	AV(25)	OUI
Nom de région	NOM_REGION	E	A50	AV(50)	OUI
Nom du cépage	NOM_CEPAGE	E	A30	AV(30)	
Nom du domaine	NOM_DOMAINE	E	A50	AV(50)	
Nom du format	NOM_FORMAT	E	A20	AV(20)	
Nom du type	NOM_TYPEAPP	E	A20	AV(20)	
Nom pays	NOM_PAYS	E	A50	AV(50)	
Nom producteur	NOM_PROD	E	A30	AV(30)	
Nom vin	NOM_VIN	E	A30	AV(30)	
Note d'appréciation	NOTE_DEGUST	E	A8	AV(8)	
Numéro bouteille	NUM_BOUTEILLE	E	N4	NS(4)	OUI
Numéro cépage	NUM_CEPAGE	E	N4	N(4)	OUI
Numéro commande	NUM_COMMANDE	E	N4	NS(4)	OUI

Anaïs JARNO

Haoyun LIAO

Weitao YAN

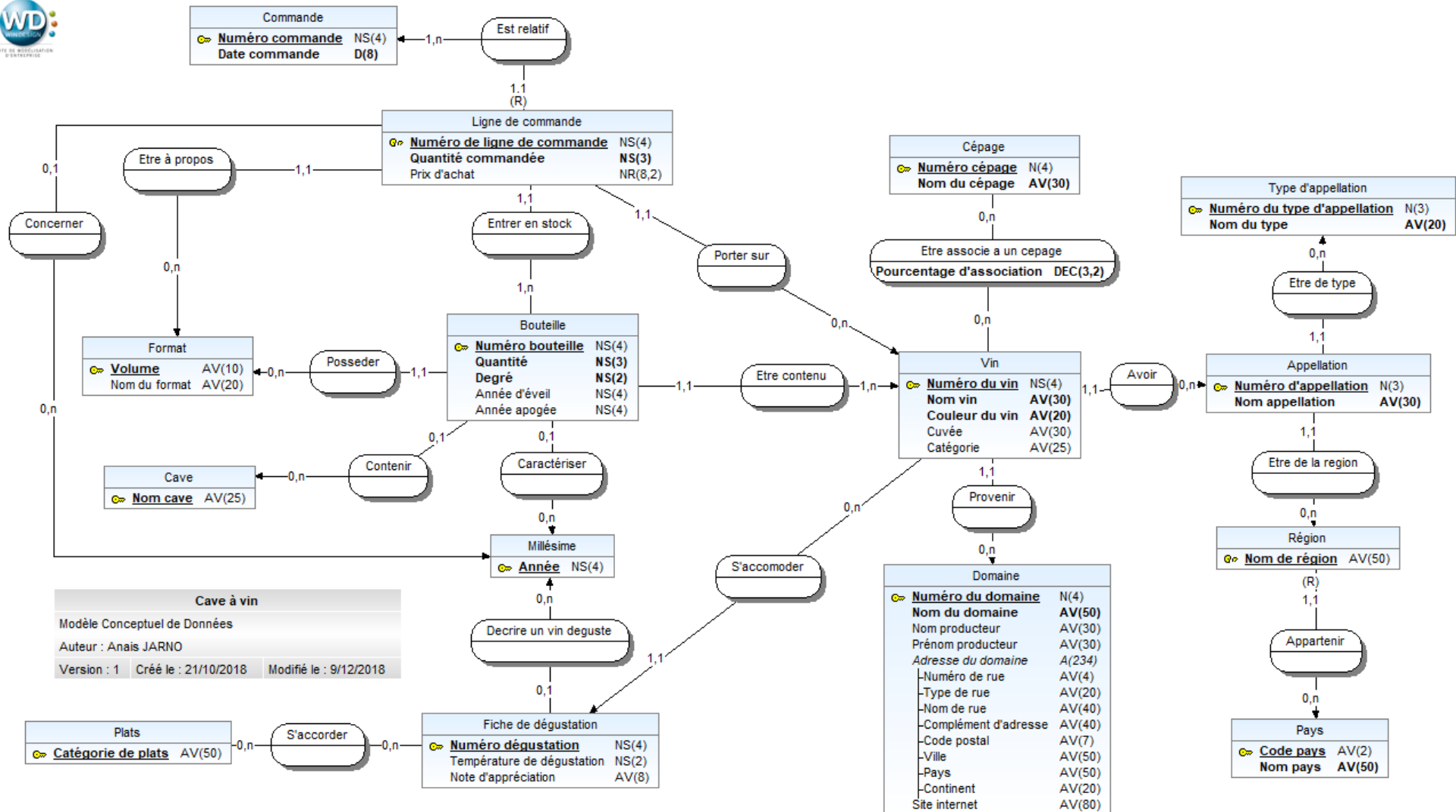
Numéro d'appellation	NUM_APPELLATION	E	N3	N(3)	OUI
Numéro dégustation	NUM_DEGUST	E	N4	NS(4)	OUI
Numéro de ligne de commande	NUMERO_LIGNE_DE_COMMANDE	E	N4	NS(4)	OUI
Numéro du domaine	NUM_DOMAINE	E	N4	N(4)	OUI
Numéro du type d'appellation	NUM_TYPEAPP	E	N3	N(3)	OUI
Numéro du vin	NUM_VIN	E	N4	NS(4)	OUI
Pourcentage d'association	POURCENTAGE_CEPAGE	E	AN(3,2)	DEC(3,2)	
Prénom producteur	PRENOM_PROD	E	A30	AV(30)	
Prix d'achat	PRIX_ACHAT	E	AN(8,2)	DEC(8,2)	
Quantité	QUANTITE_BOUTEILLE	E	N3	NS(3)	
Quantité commandée	QUANTITE_COMMANDE	E	N3	NS(3)	
Site internet	SITE_PROD	E	A80	AV(80)	
Température de dégustation	TEMP_DEGUST	E	N2	NS(2)	
Volume	VOLUME_FORMAT	E	A10	AV(10)	OUI

Anaïs JARNO

Haoyun LIAO

Weitao YAN

3.1.2 Modèle entité association

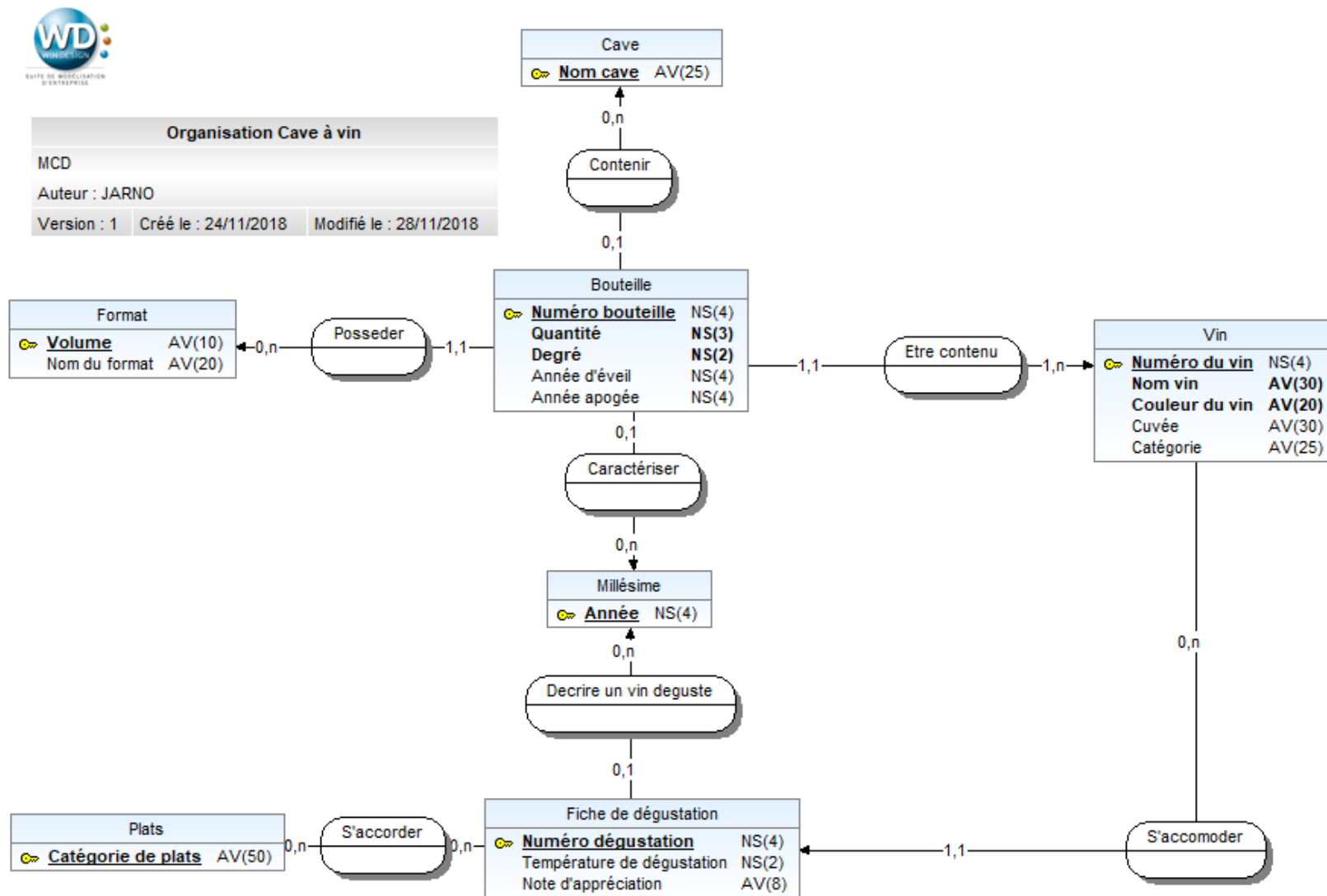


Anaïs JARNO

Haoyun LIAO

Weitao YAN

Décomposition du MCD global en sous-modèles :



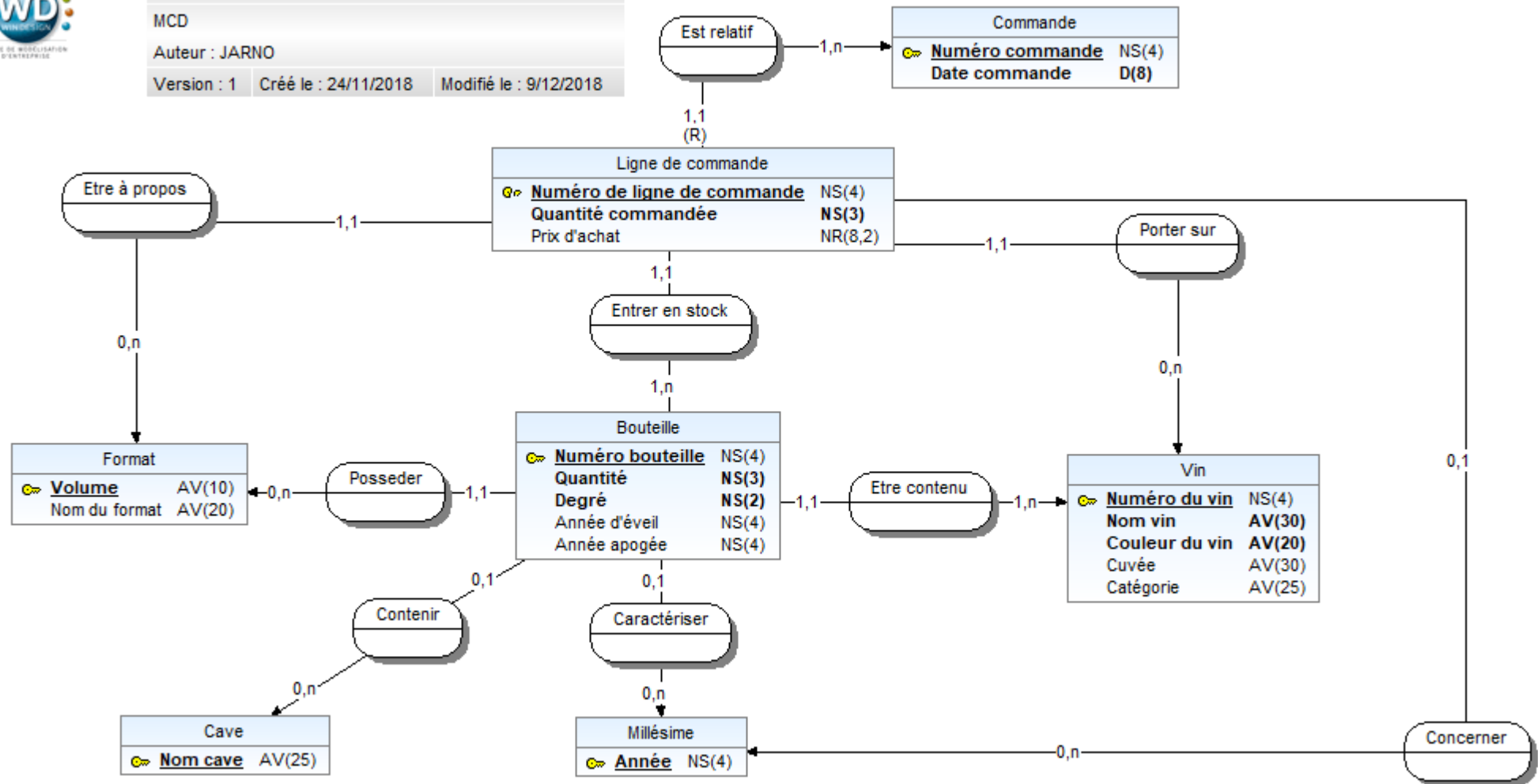
Anaïs JARNO

Haoyun LIAO

Weitao YAN



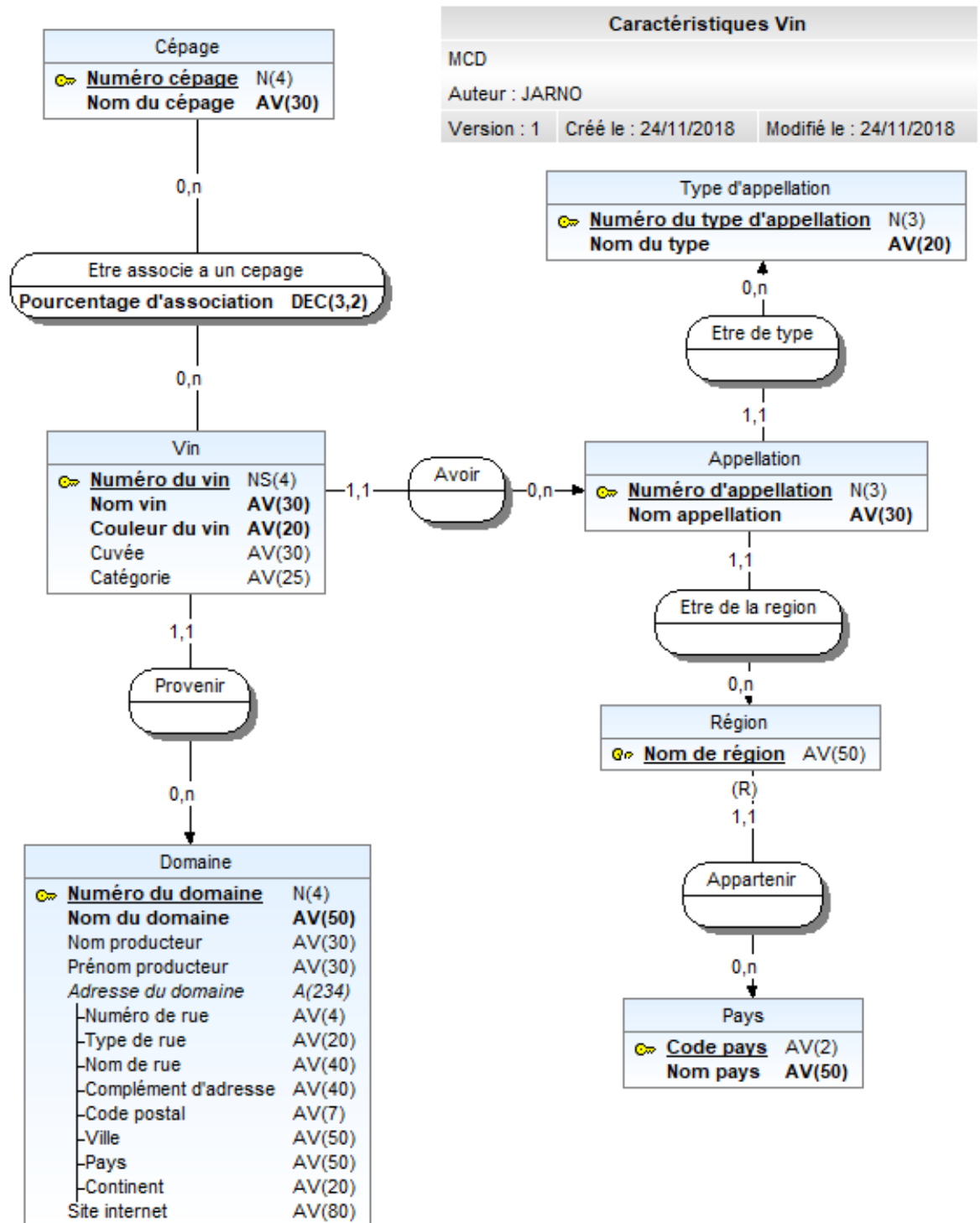
Commande Cave à vin		
MCD		
Auteur : JARNO		
Version : 1	Créé le : 24/11/2018	Modifié le : 9/12/2018



Anaïs JARNO

Haoyun LIAO

Weitao YAN

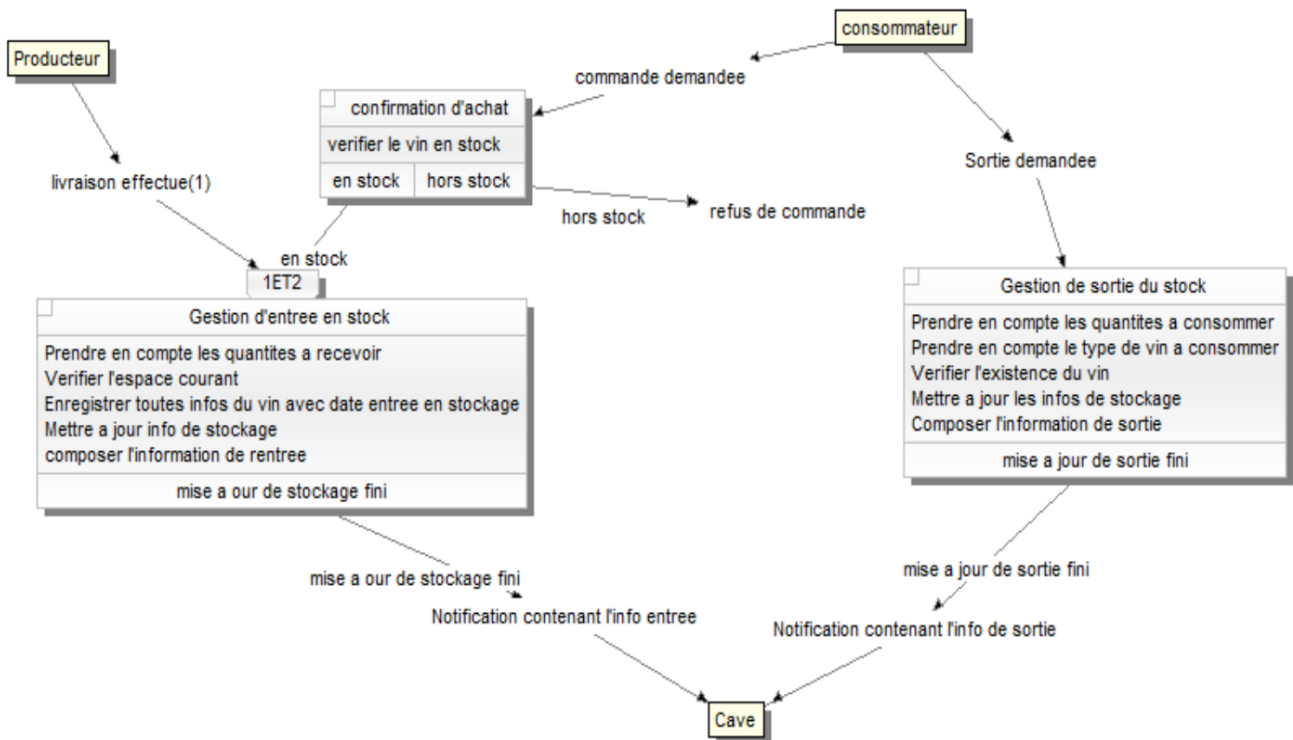
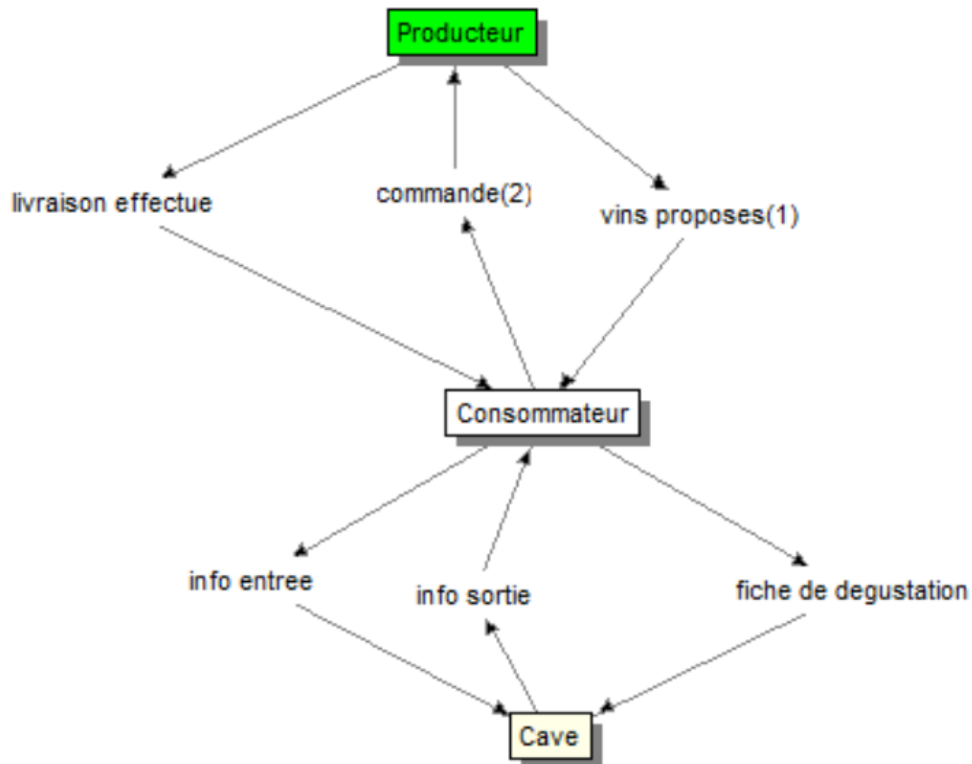


Anaïs JARNO

Haoyun LIAO

Weitao YAN

3.1.3 Modèles de communication et de traitements



Anaïs JARNO

Haoyun LIAO

Weitao YAN

4 LA MODELISATION LOGIQUE

MLR-Sous modèle-Organisation

1FN : Oui

Justification : Tous les attributs sont fonctionnellement dépendants de la clé.

2FN : Oui

Justification : Toutes les relations sauf 'S'ACCORDER' n'ont qu'une seule clé. Donc naturellement tout attribut non clé ne dépend pas d'une partie de cette clé. C'est-à-dire qu'ils sont en dépendance fonctionnelle élémentaire par rapport à la clé. Quant à 'S'ACCORDER', elle est en 2FN étant donné qu'elle n'a pas d'attribut non clé.

3FN : Oui

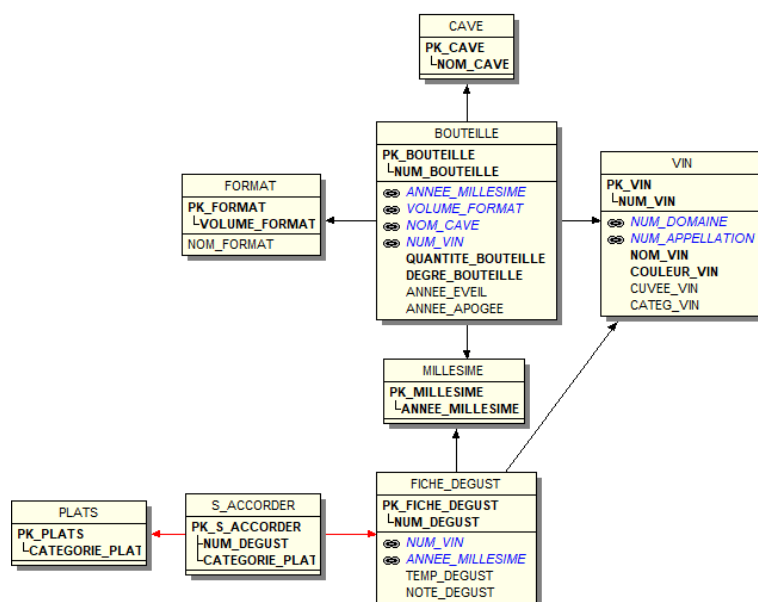
Justification :

BOUEILLE : Tous les attributs non clé sont indépendants des autres.

VIN : Tous les attributs non clé sont indépendants des autres. Pour **NUM_DOMAINE** et **NUM_APPELLATION**, vu qu'ils se réfèrent à des entités différentes n'ayant pas d'associations entre elles. Ils sont ainsi indépendants.

FICHE_DEGUST : Tous les attributs non clé sont indépendants des autres. Pour **NUM_VIN** et **ANNEE_MILLESIME**, vu que **MILLESIME** caractérise la bouteille, ils sont ainsi indépendants.

Conclusion : Ce sous-modèle est en 3FN, il est ainsi normalisé.



Anaïs JARNO

Haoyun LIAO

Weitao YAN

MLR-Sous modèle-Commande

1FN : Oui

Justification : Tous les attributs sont fonctionnellement dépendants de la clé.

2FN : Oui

Justification : Toutes les relations sauf 'LIGNE_DE_COMMANDE' et 'ENTRER EN STOCK' n'ont qu'une seule clé. Donc naturellement tout attribut non clé ne dépend pas d'une partie de cette clé. C'est-à-dire qu'ils sont en dépendance fonctionnelle élémentaire par rapport à la clé. Quant à 'ENTRER EN STOCK', elle est en 2FN étant donné qu'elle n'a pas d'attribut non clé. Quant à 'LIGNE_DE_COMMANDE', vu que l'identifiant de l'entité LIGNE_DE_COMMANDE est relatif, autrement dit, tout attribut non clé de LIGNE_DE_COMMANDE sont en dépendance fonctionnelle élémentaire par rapport à ces deux clés. Elle est ainsi en 2FN.

3FN : Oui

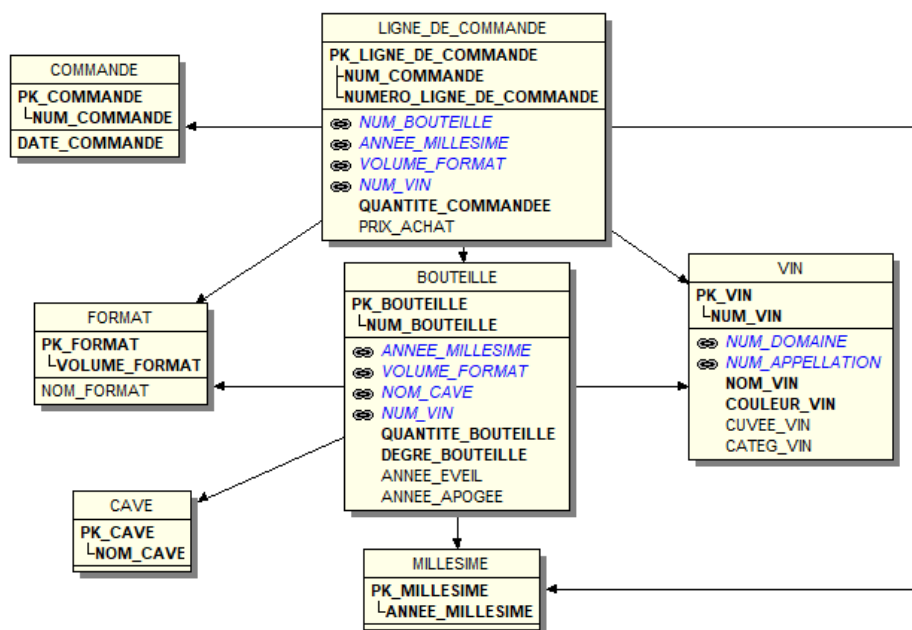
Justification :

BOUTEILLE : Tous les attributs non clé sont indépendants des autres.

VIN : Tous les attributs non clé sont indépendants des autres. Pour **NUM_DOMAINE** et **NUM_APPELLATION**, vu qu'ils se réfèrent à des entités différentes n'ayant pas d'associations entre elles. Ils sont ainsi indépendants.

LIGNE_DE_COMMANDE : Tous les attributs non clé sont indépendants des autres.

Conclusion : Ce sous-modèle est en 3FN, il est ainsi normalisé.



Anaïs JARNO

Haoyun LIAO

Weitao YAN

MLR-Sous modèle-Commande

1FN : Oui

Justification : Tous les attributs sont fonctionnellement dépendants de la clé. Dans *DOMAINE*, même s'il y a des propriétés composées, *ADR_DOMAINE* ne fait pas partie des attributs non clé. De plus, chaque valeur de la clé (*NUM_DOMAINE*) ne correspond qu'à une seule valeur des attributs composés. Elle est ainsi en 1FN.

2FN : Oui

Justification : Toutes les relations sauf '*ETRE_ASSOCIE_A_UN_CEPAGE*' et '*REGION*' n'ont qu'une seule clé. Donc naturellement tout attribut non clé ne dépend pas d'une partie de cette clé. C'est-à-dire qu'ils sont en dépendance fonctionnelle élémentaire par rapport à la clé. Quant à '*REGION*', elle est en 2FN étant donné qu'elle n'a pas d'attribut non clé. Quant à '*ETRE_ASSOCIE_A_UN_CEPAGE*', vu que *POURCENTAGE_CEPAGE* pour un vin donné est déterminé en même temps par *NUM_CEPAGE* et *NUM_VIN*, il est en dépendance fonctionnelle élémentaire par rapport à ces deux clés. La relation est ainsi en 2FN.

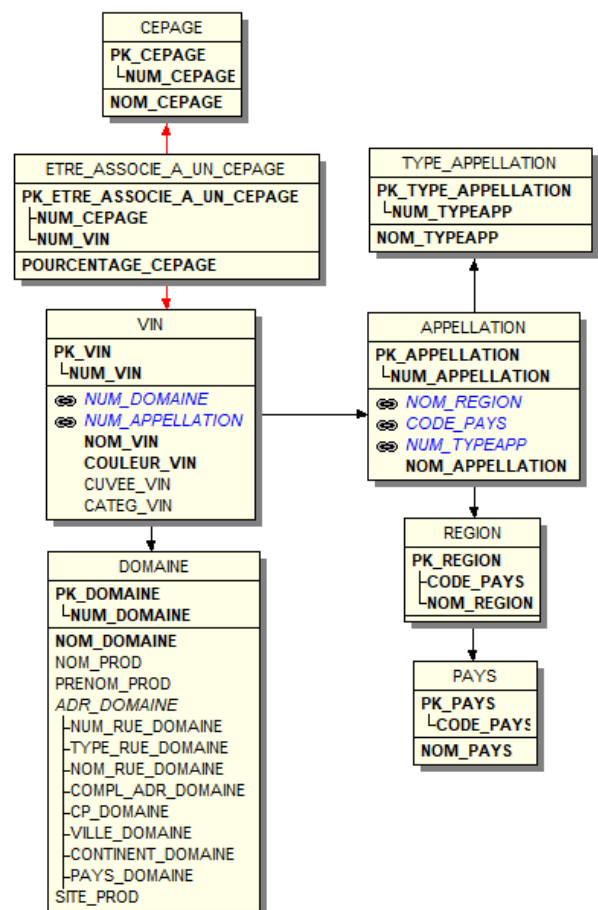
3FN : Oui

Justification :

VIN : Tous les attributs non clé sont indépendants des autres. Pour *NUM_DOMAINE* et *NUM_APPELLATION*, vu qu'ils se réfèrent à des entités différentes n'ayant pas d'associations entre elles. Ils sont ainsi indépendants.

APPELLATION : Tous les attributs non clé sont indépendants des autres.

DOMAINE : Tous les attributs non clé sont indépendants des autres. Pour l'adresse, il existe des pays qui se situent dans deux continents. Donc *PAYS_DOMAINE* et *CONTINENT_DOMAINE* sont indépendants. Même dans un pays, il se peut qu'une ville a un homonyme, donc *VILLE_DOMAINE* et *CP_DOMAINE* sont aussi indépendants.



Conclusion : Toutes les relations dans MLR sont en 3FN, donc le MLR global est normalisé.

Anaïs JARNO






Haoyun LIAO

Weitao YAN

5 LA MODELISATION DE L'IHM

En réfléchissant à l'organisation orientée métier de notre application, dans le but d'être utilisée par une personne lambda, nous avons alors découpé notre IHM en quelques formulaires qui nous semblaient être les plus pertinents.

Les informations ont été triées en différentes catégories :

-  Informations non modifiables
-  Informations pouvant être modifiées
-  Informations obligatoires
-  Informations optionnelles
-  Informations calculées

En ouvrant son application, l'utilisateur peut voir les informations de base de tous les vins qu'il connaît déjà. Il peut sélectionner le vin dans le formulaire pour afficher des informations spécifiques à celui-ci. Il peut également trouver des informations sur certains vins en effectuant une recherche selon le nom du vin et en filtrant selon sa couleur.

De même, il peut modifier certaines informations spécifiques au vin en question en accédant au bouton « *Modifier* ». Ce qui nous donne la maquette suivante :

Vin | Bouteille | Commande |

Rechercher un vin VIN.NOM_VIN

Filtrer Couleur du vin [] Numero d'appellation []

Liste des vins

Nom du vin [] Couleur du vin [] Cuvee [] Categorie []

Origine | Appellation_Cepage | Degustation |

Nom du domaine []

Nom producteur

Nom [] prenom [] Modifier

Adresse du domaine

Numero de rue [] Type de rue [] Nom de rue []

Complement d'adresse []

Continent [] Pays [] Ville [] Code postal []

Site internet []

Nom du Vin	Nombre de bouteilles	Volume	Millesime

Retour

Figure 5-1 Affichage principal - Liste des vins

Origine | Appellation_Cepage | Degustation |

Nom du domaine []

Nom producteur

Nom [] prenom [] Modifier

Adresse du domaine

Numero de rue [] Type de rue [] Nom de rue []

Complement d'adresse []

Continent [] Pays [] Ville [] Code postal []

Site internet []

Figure 5-2 Affichage principal - L'onglet de l'origine du vin

Origine | Appellation_Cepage | Degustation |

Appellation

Nom []

Type []

Region []

Pays [] Modifier

Cepage

Nom []

Pourcentage []

Figure 5-3 Affichage principal - L'onglet de l'appellation et le cépage du vin

Anaïs JARNO

Haoyun LIAO

Weitao YAN

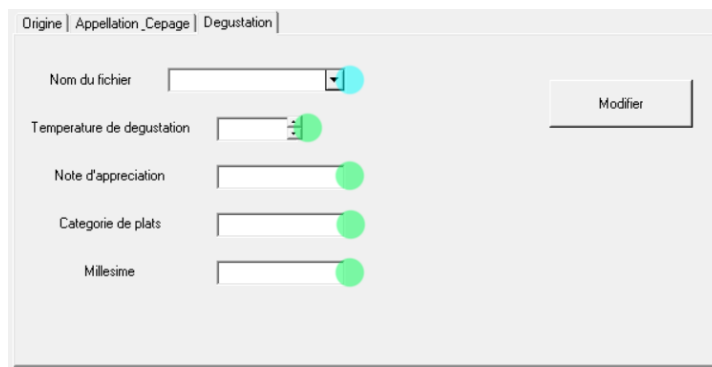
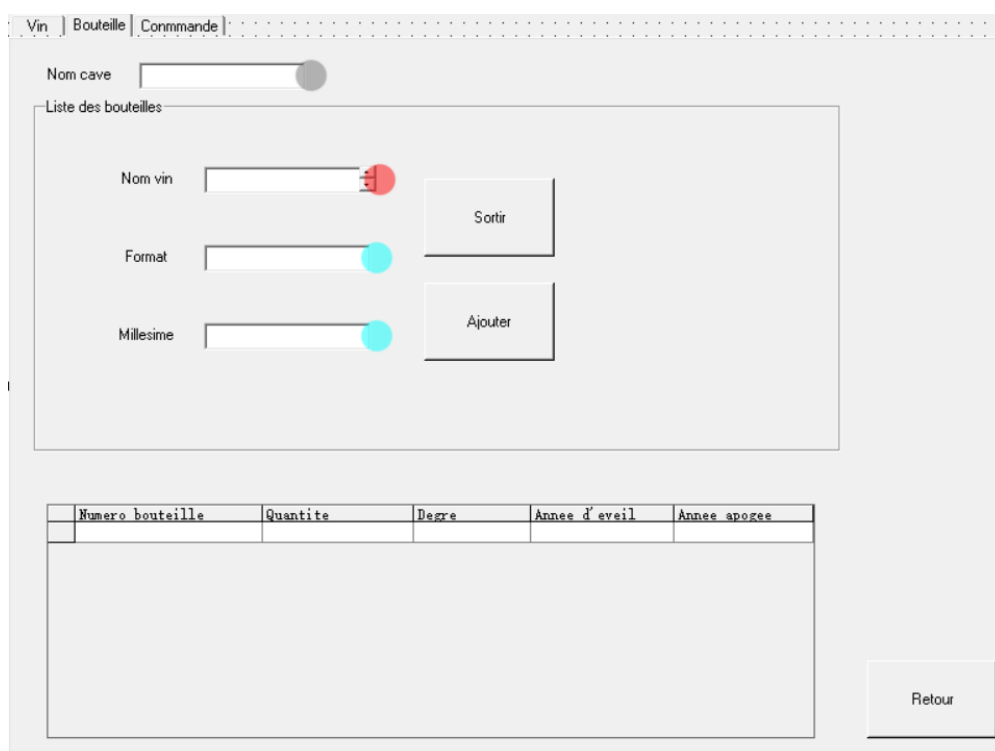


Figure 5-4 Affichage principal - L'onglet de la dégustation du vin

Considérons la sortie d'une bouteille. L'utilisateur peut renseigner les noms des vins dont il souhaite retirer des bouteilles, toutes les bouteilles du même nom de vin seront affichées au bas de la page. L'utilisateur retire une unité en validant la sélection de la bouteille.

Ce même utilisateur peut également ajouter une nouvelle bouteille en choisissant le nom du vin puis en renseignant les informations de base.



Numero bouteille	Quantite	Degre	Annee d' eveil	Annee apogee

Figure 5-5 Sortir et ajouter bouteille

Anaïs JARNO

Haoyun LIAO

Weitao YAN

De même, nous avons une page pour voir toutes les informations concernant les commandes.

Selon la date entrée, le tableau ci-dessous affiche les informations de commande. En sélectionnant une commande, toutes les informations concernant les bouteilles comprises dans la commande sélectionnée seront affichées en bas de la page.

Vin

Bouteille

Commande

Date commande

Date commande	Nom vin	Format	Quantite	Prix

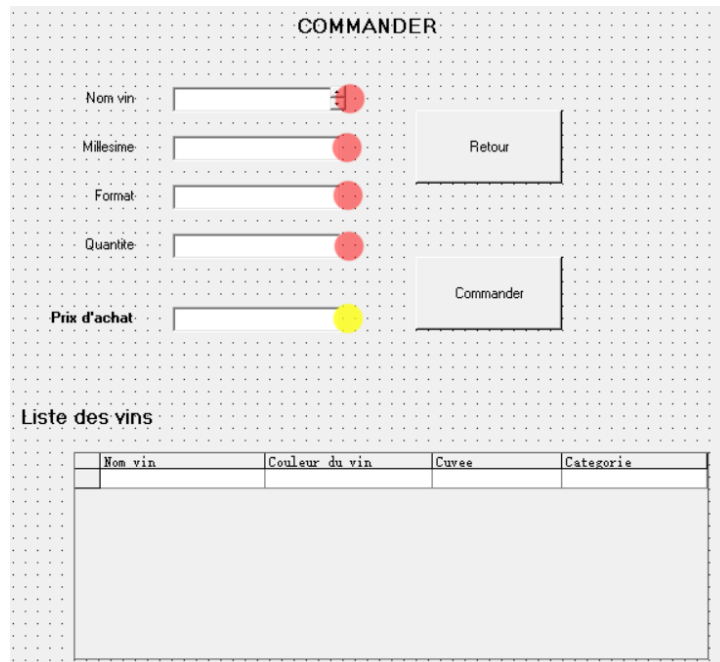
Bouteilles en stock

Numero bouteille	Millesime	Degre	Annee d'evail	Annee apogee

Retour

Figure 5-6 Voir les informations de commande

L'utilisateur peut entrer une nouvelle commande de vin via notre application. Les informations sur le vin pouvant être commandé sont affichées en bas de la page. L'utilisateur peut choisir le vin ou entrer le nom du vin pour déterminer le vin qu'il souhaite acheter. L'utilisateur entre ensuite les informations de base. L'utilisateur peut alors valider l'entrée de sa commande en appuyant sur le bouton « Commander ».



COMMANDER

Nom vin:

Millesime:

Format:

Quantite:

Prix d'achat:

Retour

Commander

Liste des vins

Nom vin	Couleur du vin	Cuvee	Categorie

Figure 5-2 Visuel de commande de bouteilles

6 DOSSIER DE DEVELOPPEMENT DU PROJET

6.1 PRESENTATION DE L'INTERFACE DEVELOPPEE AVEC MS/ACCESS

6.1.1 Page principale

Tout d'abord, il nous a semblé important que le programme s'ouvre directement sur la page permettant l'accès aux bouteilles, car c'est la fonctionnalité qui devrait intéresser le plus l'utilisateur. Ainsi notre page d'accueil et page principale correspond à la suivante :

Vos bouteilles en stock

Cave :

Millésime :

Volume (L) :

Année d'éveil :

Nom du vin :

Quantité en stock :

Degré :

Année d'apogée :

Nom du vin	Millésime	Volume (L)	NOM_CAVE	QUANTITE_E	DEGRE_BOU	ANNEE_EVEI	ANNEE_APO
Pinot Blanc Clas	2016	0,75	Cave	2	12		
Pinot Gris Classi	2016	0,375	Cave	1	13		
Riesling Classic	2016	1,5	Cave	3	12		
Côtes du Jura R	2011	0,75	Cave	2	12		
Vin Jaune	2001	0,375	Grenier	3	14		
Pinot Noir Class	2015	1,5	Placard	5	13		
✱							

Anaïs JARNO

Haoyun LIAO

Weitao YAN

6.1.2 Page des vins connus

Ensuite, une autre fonctionnalité importante pour l'utilisateur était d'avoir accès aux vins qu'il connaît déjà, et bien sûr de pouvoir ajouter de nouveaux vins.

C'est ainsi que sur cette page, nous pouvons voir tous les vins connus de l'utilisateur.

Tous les vins connus

Nom du vin :

Domaine :

Appellation :

Couleur :

Catégorie :


Cuvée :

Nom du vin	Domaine	Appellation	Couleur	CATEG_VIN	CUVEE_VIN
Côtes du Jura R	Rolet Père & Fil	Côtes du Jura	Rouge	AOP	
Pinot Blanc Clas	Maison Hugel	Alsace ou Vin d'	Blanc	AOC	
Pinot Gris Classi	Maison Hugel	Alsace ou Vin d'	Blanc	AOC	
Pinot Noir Class	Maison Hugel	Alsace ou Vin d'	Rouge	AOC	
Riesling Classic	Maison Hugel	Alsace Grand Cr	Blanc	AOC	
Supp	Domaine de la T	Alsace Grand Cr	C	AOC	
Vin Jaune	Domaine de la T	Arbois	Jaune	AOP	

6.2 PRESENTATION DES REQUETES DEVELOPPEES

Nous avons deux requêtes principales pour les formulaires et les autres requêtes sont utilisées pour remplir les Combobox.

La requête suivante est celle utilisée pour notre formulaire principal, Bouteilles en stock

 REQ - Bouteilles en stock

Cave : ▾	Nom du vin : ▾	Millésime : ▾	Volume (L) : ▾	Quantité en : ▾	Degré : ▾	Année d'éveï ▾	Année d'apoï ▾
Cave	1	2016	0,75	2	12		
Cave	6	2016	0,375	1	13		
Cave	8	2016	1,5	3	12		
Cave	12	2011	0,75	2	12		
Grenier	10	2001	0,375	3	14		
Placard	7	2015	1,5	5	13		
*							

Le code SQL associé est le suivant :

```
SELECT BOUTEILLE.NOM_CAVE
      , BOUTEILLE.NUM_VIN
      , BOUTEILLE.ANNEE_MILLESIME
      , BOUTEILLE.VOLUME_FORMAT
      , BOUTEILLE.QUANTITE_BOUTEILLE
      , BOUTEILLE.DEGRE_BOUTEILLE
      , BOUTEILLE.ANNEE_EVEIL
      , BOUTEILLE.ANNEE_APOGEE

FROM BOUTEILLE

ORDER BY BOUTEILLE.NOM_CAVE
      , BOUTEILLE.NUM_VIN
      , BOUTEILLE.ANNEE_MILLESIME
      , BOUTEILLE.QUANTITE_BOUTEILLE DESC;
```

Il nous permet de sélectionner les champs voulus tout en exerçant un tri sur certaines valeurs.

Ensuite, nous avons besoin du nom du vin plutôt que de son numéro pour l’affichage. Car en effet, on affiche le numéro du vin car c’est le champ présent dans bouteille. Cependant celui-ci ne signifie rien pour l’utilisateur, il lui faut donc plutôt son nom, présent dans la Combobox remplie grâce à la requête suivante :

Anaïs JARNO

Haoyun LIAO

Weitao YAN

REQ - Vin

Nom du vin :	Nom du vin :
12	Côtes du Jura R
1	Pinot Blanc Clas
6	Pinot Gris Classi
7	Pinot Noir Class
8	Riesling Classic
10	Vin Jaune
*	(Nouv.)

De la même façon nous avons besoin des noms de cave, et nous avons donc rempli la Combobox par les valeurs de la table Cave étant donné que son Identifiant est aussi son nom. Nous avons également besoin du Volume, de la capacité de contenance de la bouteille. Nous avons traité cette Combobox de la même manière que celle pour les caves, à la différence près que nous avons ajouté l'affichage du nom de format avec le volume associé.

Puis, pour ce qui est du formulaire des vins nous avons la requête suivante :

REQ - Tous les vins connus

Nom du vin :	Domaine :	Appellation :	Couleur :	Catégorie :	Cuvée :	Nom du vin :
Côtes du Jura R	3	6 Rouge	AOP			12
Pinot Blanc Clas	1	3 Blanc	AOC			1
Pinot Gris Classi	1	3 Blanc	AOC			6
Pinot Noir Class	1	3 Rouge	AOC			7
Riesling Classic	1	4 Blanc	AOC			8
Vin Jaune	2	9 Jaune	AOP			10
*						(Nouv.)

Nous avons fonctionné de la même manière que pour le formulaire sur les bouteilles.

C'est ainsi que nous avons eu besoin d'une requête pour le domaine:

REQ - Domaine

Domaine :	Nom du domaine :
2	Domaine de la Touraize
1	Maison Hugel
3	Rolet Père & Fils
*	(Nouv.)

Anaïs JARNO

Haoyun LIAO

Weitao YAN

De la même façon pour les appellations:

REQ - Appellations

Appellation :	Appellation :
4	Alsace Grand Cr
3	Alsace ou Vin d'
9	Arbois
11	Château-Chalor
6	Côtes du Jura
5	Crémant d'Alsac
7	Crémant du Jura
10	Étoile
8	Macvin du Jura
*	(Nouv.)

Par la suite, nous avons voulu proposer à l'utilisateur de choisir la couleur de son vin et sa catégorie via une autre Combobox. Cependant ces valeurs ne sont que des attributs de la table VIN et non des tables à part entière.

C'est donc ainsi que nous avons formulés les requêtes suivantes :

REQ - Catégories de vin

Catégorie :
AOC
AOP

REQ - Couleurs de vin

Couleur :
Blanc
Jaune
Rouge

Dont les codes SQL associés sont les suivants :

```
SELECT DISTINCT VIN.CATEG_VIN
FROM VIN
WHERE (((VIN.CATEG_VIN) <> ""))
ORDER BY VIN.CATEG_VIN;
```

```
SELECT DISTINCT VIN.COULEUR_VIN
FROM VIN
ORDER BY VIN.COULEUR_VIN;
```

Ceci nous permet de sélectionner toutes les catégories/couleurs de vin et de les trier. De plus le mot de clef DISTINCT au niveau du SELECT permet de sélectionner de manière unique les éléments. C'est-à-dire que s'il y a plusieurs vins avec l'attribut *jaune* pour la couleur, *jaune* n'apparaîtra qu'une seule fois dans la requête.

Anaïs JARNO

Haoyun LIAO

Weitao YAN

6.3 PRESENTATION DU CODE VBA DEVELOPPE DANS L'APPLICATION

6.3.1 Modules VBA

Nous avons 4 principaux modules :

- Désactivation de la Touche *MAJ*
- Vérification des liens
- Chargement du ruban
- Ruban contrôle

La désactivation de la touche *MAJ* est décrite un peu plus tard dans la partie sur les options de sécurité.

Le module de vérification des liens permet d'appeler le formulaire de sécurité exécutant la protection de la base de données et la réactualisation des liens si ceux-ci ne sont plus valables. Autrement, il ouvre juste le formulaire d'accueil.

Il est ainsi codé :

```
Public Function VerifierLiens()

Dim Dbs As DAO.Database
Dim rst As DAO.Recordset
Set Dbs = CurrentDb()

On Error Resume Next

Dim nbTbl As Integer
nbTbl = Dbs.TableDefs.Count

Dim idx As Integer
Dim TblDef As TableDef
For idx = 0 To nbTbl - 1
    Set TblDef = Dbs.TableDefs(idx)
    If TblDef.Attributes = dbAttachedTable Then
        Set rst = Dbs.OpenRecordset(TblDef.Name)
    End If
Next idx

If Err <> 0 Then
    'Les tables ont été déplacées, il faut rétablir les liens
    DoCmd.ShowToolbar "Ribbon", acToolbarNo
    DoCmd.OpenForm "Form - Securite", acNormal
Else
    DoCmd.OpenForm "Form - Bouteilles en stock", acNormal
End If

rst.Close
Dbs.Close
Set rst = Nothing
Set Dbs = Nothing

End Function
```

Anaïs JARNO

Haoyun LIAO

Weitao YAN

Le module permettant le chargement du ruban personnalisé en *XML* a été codé de la façon suivante :

```
Option Compare Database

Public Function LoadRibbon()

Dim strXML As String
Dim oFso As New FileSystemObject
Dim oFtxt As TextStream

'Charge le fichier XML en mémoire
Set oFtxt = oFso.OpenTextFile(CurrentProject.Path & _
    "\caveAVin_ribbon.XML", ForReading)

'Récupère le contenu
strXML = oFtxt.ReadAll

'Charge le ruban personnalisé correspondant
Application.LoadCustomUI "rubanperso", strXML

End Function
```

Le module Ruban contrôle configure les boutons **Mes bouteilles en stock** et **Mes vins**. En effet, on définit la réaction de ces deux boutons. Lorsque **Mes vins** est pressé, on ouvre le formulaire concernant les vins. Tandis que lorsque l'on clique sur **Mes bouteilles**, en plus d'ouvrir le formulaire correspondant, cela permet de réactualiser les enregistrements (et donc les Combobox présentes dans le formulaire).

```
Public Sub btnBouteilles_action(ByVal control As IRibbonControl)

    DoCmd.OpenForm "Form - Bouteilles en stock", acNormal, "", "", , acNormal
    DoCmd.RefreshRecord

End Sub

Public Sub btnVins_action(ByVal control As IRibbonControl)

    DoCmd.OpenForm "Form - Tous les vins connus", acNormal, "", "", , acNormal

End Sub
```

6.3.2 VBA pour formulaires

Une méthode est appelée pour les différents formulaires, celle-ci est la suivante :

```
Private Sub Form_AfterUpdate()  
  
Me.Requery  
Me.Refresh  
  
End Sub
```

Elle permet de réactualiser le formulaire courant et ses enregistrements, ce qui met à jour les Combobox présentes dans le formulaire.

6.3.3 Code via DAO

Les seuls codes comportant du DAO sont ceux s'occupant des options de sécurité détaillés dans la partie suivante.

6.4 PRESENTATION DES OPTIONS DE SECURITE

Nous avons sécurisé l'application de diverses manières.

Tout d'abord, il nous a semblé important que l'utilisateur n'ait pas accès à toutes les options proposées par Access. Autrement, il aurait pu créer de nouvelles tables, requêtes...

Pour cela, nous avons mis en place un ruban personnalisé.

6.4.1 Mise en place ruban personnalisé

Celui-ci est implémenté grâce au fichier suivant : **caveAVin ribbon.XML**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">
  <ribbon startFromScratch="true">
    <tabs>
      <tab id="tabGestion" label="Gestion de votre cave" visible="true">
        <group id="groupCave" label="Ma cave">
          <gallery id="galCave" label="Mes outils" columns="1" rows="3" imageMso="BookmarkInsert" size="large">
            <button id="itBouteille" label="Mes bouteilles en stock" screentip="Voir et ajouter des bouteilles dans la cave" onAction="btnBouteilles_action"/>
            <button id="itVin" label="Tous mes vins" screentip="Pour consulter et ajouter des vins" onAction="btnVins_action"/>
            <button id="itCommande" label="Commander" screentip="Commander des bouteilles ou consulter les commandes" onAction="btnVins_action"/>
          </gallery>
        </group>
        <group id="groupFile" label="Fichier">
          <button idMso="FileSave" label="Enregistrer" size="large"/>
          <control idMso="Undo" imageMso="Undo" size="large" label="Undo"/>
          <control idMso="Redo" imageMso="Redo" size="large" label="Redo"/>
        </group>
        <group idMso="GroupClipboard" label="Presse-papier"/>
        <group id="groupEnregistrements" label="Enregistrements">
          <button idMso="GoToNewRecord" label="Nouveau" size="large"/>
          <button idMso="RecordsDeleteRecord" label="Supprimer" size="normal"/>
          <button idMso="DataRefreshAll" label="Reactualiser" size="normal"/>
        </group>
        <group id="GroupSortAndFilter" label="Filtrer et trier">
          <toggleButton idMso="FiltersMenu" label="Filtrer" size="large"/>
          <toggleButton idMso="SortUp" label="Croissant" size="normal"/>
          <toggleButton idMso="SortDown" label="Décroissant" size="normal"/>
          <button idMso="SortRemoveAllSorts" label="Effacer tous les tris" size="normal"/>
        </group>
        <group idMso="GroupFindAccess"/>
        <group id="groupExport" label="Exporter">
          <control idMso="ExportExcel" size="large"/>
        </group>
        <group idMso="GroupPrintPreviewPrintAccess" label="Imprimer"/>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

Il est ensuite chargé grâce à la macro/fonction VBA suivante : **LoadRibbon**

```
Public Function LoadRibbon()

Dim strXML As String
Dim oFso As New FileSystemObject
Dim oFtxt As TextStream

'Charge le fichier XML en mémoire
Set oFtxt = oFso.OpenTextFile(CurrentProject.Path & _
    "\caveAVin_ribbon.XML", ForReading)

'Récupère le contenu
strXML = oFtxt.ReadAll

'Charge le ruban personnalisé correspondant
Application.LoadCustomUI "rubanperso", strXML

End Function
```

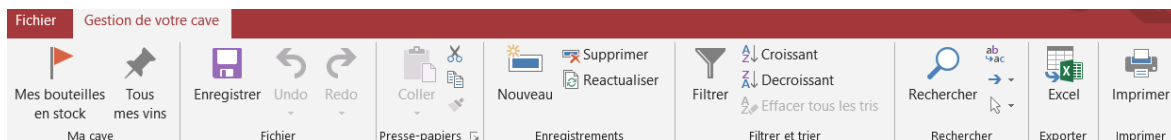
Anaïs JARNO

Haoyun LIAO

Weitao YAN

Celle-ci est à son tour appelée par la macro **AutoExec**. Elle charge alors le ruban et comme nous avons défini dans les *options de la base de données actuelle* que le ruban principal était le ruban personnalisé, alors celui-ci est tout de suite lancé et remplace celui de la base Access. En effet, toute macro nommée **AutoExec** est exécutée à chaque fois au démarrage de l'application.

Dans notre application, nous avons alors le ruban suivant :



Les différents onglets choisis et mis en place :

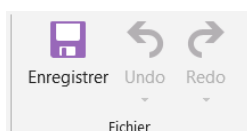
○ **Ma cave**



Cet onglet comporte une galerie *Mes outils* avec les trois boutons (*Mes bouteilles en stock – Tous mes vins – Commander*) qui renvoient les formulaires principaux de gestion de ceux-ci.

Le code XML de ces boutons a une fonction *onAction* faisant appel à une macro VBA implémentée par nos soins, servant à ouvrir les formulaires correspondants.

○ **Fichier**



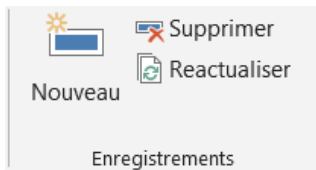
Onglet personnalisé permettant *la sauvegarde du fichier, le retour arrière et le retour avant*, boutons XML pré-codés par la bibliothèque office.

○ **Presse-papiers**



Onglet pré-codé par la bibliothèque office.

○ Enregistrements



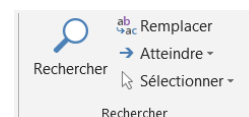
Onglet personnalisé utilisant les boutons XML pré-codés par la bibliothèque office permettant *la création d'un nouvel enregistrement, la suppression d'un enregistrement et la réactualiser de tout le document.*

○ Filtrer et trier



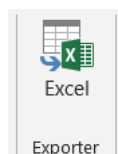
Onglet personnalisé limitant les filtres, et les tris aux *filtres généraux* et au *tri croissant/décroissant*, contrôles XML pré-codés par la bibliothèque office.

○ Rechercher



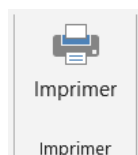
Onglet pré-codé par la bibliothèque office, permettant la *recherche* et le *déplacement parmi les enregistrements.*

○ Exporter



Onglet personnalisé permettant *l'exportation de la page courantes vers un fichier Excel*, bouton XML pré-codé par la bibliothèque office.

○ Imprimer



Onglet pré-codé par la bibliothèque office, permettant *l'aperçu et l'impression.*

Aussi, la macro **AutoExec** ouvre le formulaire principal en tant que page d'accueil pour l'utilisateur.

6.4.2 Le verrouillage de l'accès aux données et la réactualisation des liens

6.4.2.1 Pour l'application

Nous avons également mis en place une application pour verrouiller l'accès aux données. Nous avons choisi de restreindre l'accès de l'utilisateur aux Tables, Requêtes, Formulaires, Macros et Modules habituellement accessibles au travers du volet de navigation.

- 1 - Ainsi, via les **options de la base de données**, nous avons choisi de **masquer le Volet de Navigation**.

Cependant, cette sécurité n'était pas suffisante. Il suffisait d'appuyer sur la touche *MAJ* au démarrage de l'application pour accéder aux données, formulaires, etc.

- 2 - Grâce à un **code VBA**, nous avons donc la possibilité de **désactiver la touche MAJ**, et ainsi de verrouiller l'accès aux tables.

```
' ---
' DESACTIVATION DE LA TOUCHE MAJ
' ---
Sub DésactiverMaj()
    Dim blnAutoriserMaj As Boolean

    ' Si blnAutoriserMaj vaut False,
    ' la touche [Maj] sera désactivée.
    ' Si blnAutoriserMaj vaut True,
    ' la touche [Maj] sera activée.
    blnAutoriserMaj = False

    ModifiePropriete "AllowBypassKey", dbBoolean, blnAutoriserMaj
    If blnAutoriserMaj Then
        MsgBox "La touche [Maj] est activée. Fermez la base et réouvrez-la pour tester."
    Else
        MsgBox "La touche [Maj] est désactivée. Fermez la base et réouvrez-la pour tester."
    End If
End Sub
```

Cette macro appelle une **fonction VBA DAO** pour modifier directement la propriété *AllowBypassKey* qui définit si la touche *MAJ* peut être utilisée ou non pour l'ouverture du centre de navigation.

```

' ---
' PARAMETRAGE D'UNE PROPRIETE DE BASE
' ---
Function ModifiePropr( _
    chNomPropriété As String, _
    varTypeProp As Variant, _
    varValeurProp As Variant) As Integer

    Dim bds As DAO.Database, prp As DAO.Property
    Const conErreurPropNonTrouvée = 3270

    Set bds = CurrentDb
    On Error GoTo Change_Err
    bds.Properties(chNomPropriété) = varValeurProp
    ModifiePropr = True

Change_Sortie:
Exit Function

Change_Err:
    If err = conErreurPropNonTrouvée Then ' Propriété non trouvée.
        Set prp = bds.CreateProperty(chNomPropriété, varTypeProp, varValeurProp)
        bds.Properties.Append prp
        Resume Next
    Else
        ' Erreur inconnue.
        ModifiePropr = False
        Resume Change_Sortie
    End If
End Function

```

Une fois la macro lancée, il a alors suffi de fermer, puis de rouvrir la base de données pour que la désactivation soit effective.

6.4.2.2 Pour la base de données

Nous avons séparé l'application de la base de données en deux fichiers. Cependant lorsque la base de données est déplacée ou renommée, celle-ci n'est plus trouvée par l'application, il fallait donc une manière de réactualiser les liens entre les tables et l'application. Pour cela nous avons utilisé le code VBA suivant :

```

Sub RfshLinkedTables(FullDBpathName As String)

Dim db As DAO.Database, tdef As DAO.TableDef
Dim strSceDB As String, strConn As String
Dim ConnParams() As String, varParam As Variant

strSceDB = FullDBpathName

Set db = CurrentDb
For Each tdef In db.TableDefs
    If (tdef.Attributes And dbAttachedTable) = dbAttachedTable Then
        ' Nouvelle chaîne de connexion
        strConn = ";"
        ' Récupérer mot de passe si existe
        ConnParams() = Split(tdef.Connect, ";")
        For Each varParam In ConnParams
            If varParam Like "PWD=*" Then
                strConn = strConn & varParam & ";"
            End If
        Next
        ' Ajout chemin vers la base dorsale
        strConn = strConn & "DATABASE=" & strSceDB
        ' Rafraichir lien
        tdef.Connect = strConn
        tdef.RefreshLink
    End If
Next

```

Ce code fait partie des options appelées par le clic sur le bouton *Protéger et Réactualiser* du formulaire de sécurité. En effet, voici le code de ce bouton :

```

Private Sub btnProtéger_Click()

If IsNull(Me.txtBase) Then
    MsgBox "Sélectionnez une base de données au préalable !", vbInformation
    Exit Sub
End If
If Dir(Me.txtBase) = "" Then
    MsgBox "Fichier introuvable !", vbExclamation
    Exit Sub
End If

RfshLinkedTables (Me.txtBase)

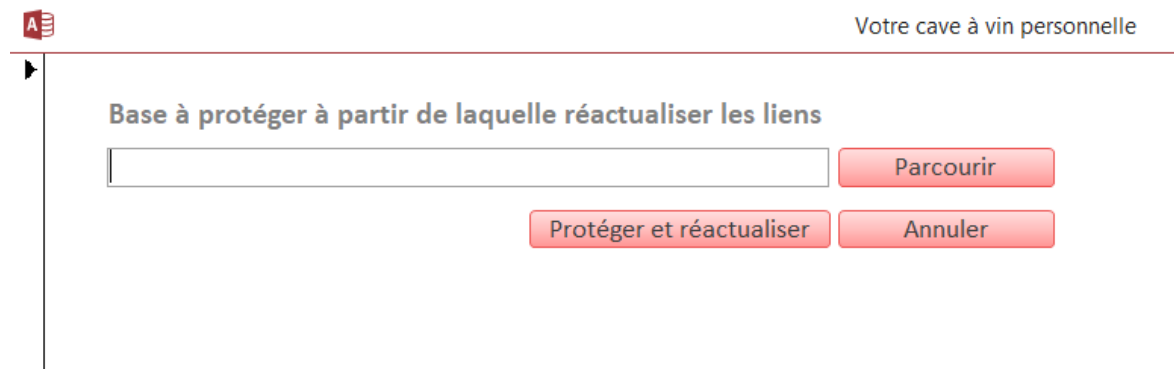
DefinirToucheMAJ Me.txtBase, False
MsgBox "Touche Majuscule désactivée et liens réactualisés", vbInformation

DoCmd.Close
DoCmd.OpenForm "Form - Bouteilles en stock", acNormal
DoCmd.ShowToolbar "Ribbon", acToolbarYes

End Sub

```

Comme précisé dans la fin du code, on utilise *ShowToolbar* qui permet d’afficher ou non le ruban. Pour des mesures de sécurité, nous avons désactivé l’affichage du ruban lorsque le formulaire de sécurité est appelé. Voici donc l’affichage produit :



Si les liens ont bien été réactualisés, on ouvre alors le formulaire principal et on affiche le ruban.

6.4.3 Contrôle des saisies

Aussi des contrôles concernant les saisies ont été réalisés.

Par exemple, le millésime doit être supérieur à 1800 lors de la saisie, autrement elle est refusée et un message d’erreur averti l’utilisateur.

Aussi, certains champs parmi les *Combobox* refusent toute nouvelle saisie car il est nécessaire de faire un nouvel enregistrement dans la base de données pour le champ donné. Ainsi, l’option refusant la saisie d’un nouvel élément dans la *Combobox* permet à l’utilisateur d’être encadré car il ne lui sera alors pas possible d’essayer de rentrer cet élément, et donc il ne recevra pas d’erreur lorsqu’il voudra valider la saisie de sa nouvelle bouteille, ou de son nouveau vin.

6.4.4 Conclusion sur la sécurité

Ainsi lorsque l’application est lancée, même en maintenant la touche MAJ lors du démarrage de l’application, l’utilisateur ne peut accéder au volet de navigation. De même il visualise le formulaire de base de l’application et n’a accès qu’au ruban personnalisé et non à celui de base. L’utilisateur est donc contraint d’utiliser les fonctionnalités à sa portée, fonctionnalités n’ayant pas d’influence sur la sécurité de l’application.

Anaïs JARNO

Haoyun LIAO

Weitao YAN

7 CONCLUSION SUR LE PROJET

Nous avons choisi ce thème pour permettre aux chinois présents dans le groupe de découvrir un thème fortement lié à la culture française. Cela s'est résulté en une grande difficulté ; freinant considérablement notre avancée, faute d'entente entre les membres du groupe ne comprenant pas le projet de la même façon.

Toutefois, nous remarquerons que nous avons tout de même pu mener à bien la partie conceptuelle du projet grâce un gros investissement...

Malgré tout, la partie concernant le développement de l'appliquatif fut difficile à achever. En effet, il manque nombre de fonctionnalités en comparaison du modèle fourni. On notera néanmoins que les parties fournies sont fonctionnelles et peuvent être parfaitement utilisées. Une réelle utilisation de l'application se révélerait cependant complexe du fait du manque de fonctionnalités présentes.

Les buts concernant le développement de l'appliquatif ne sont donc que partiellement atteints.

Ainsi, l'application telle qu'elle est, est fonctionnelle mais ne se prête pas à l'utilisation.

8 SOURCES

Sites internet

<https://www.linternaute.com/homme/art-de-vivre/1011553-appellation-cepage-domaine-quelles-differences/>

<https://www.le-vin-pas-a-pas.com/4-niveaux-a-connaître-pour-bien-classer-les-vins/>

<http://www.vinsvignesvignerons.com/Regions/Alsace/LA-REGION-ALSACE/Les-appellations>

Consultation de diverses personnes ayant des connaissances en vitiviniculture et même en œnologie

Toutes les images présentes dans notre rapport, sont citées dans nos sources, libres de droits ou alors réalisées par nos propres soins.