

Projet STIVE



Sommaire

1. Présentation
2. Gestion du projet
3. Structure de la base de données
4. Le code
5. Problèmes rencontrés
6. Conclusion

1.Présentation

a.Fonctionnalités attendues

Le projet que nous avons à réaliser consiste à créer un site web pour une entreprise de vin.

L'objectif est de le créer en utilisant le framework Asp.Net core (MVC) en C#.

L'administrateur doit pouvoir être capable de créer des clients, des fournisseurs, des familles de vin, des articles. Il doit également pouvoir modifier les stocks (gérer la quantité, la disponibilité), les prix unitaires, et les prix par lot.

La création de compte client doit être possible afin de pouvoir créer un panier, puis passer des commandes. Les clients doivent pouvoir passer commande même si le stock est insuffisant.

Lorsqu'un produit n'est plus disponible en stock, un bon de commande pour ce produit doit être automatiquement envoyé à un fournisseur. Lorsque cette commande sera livrée, la quantité livrée doit pouvoir s'ajouter au stock.

Le réapprovisionnement automatique doit être activable et désactivable par l'administrateur.

2. Gestion du projet

a. Répartition des tâches

Le projet devra être disponible pour avril 2022, et nous serons trois pour le réaliser.

Pour travailler sur le projet, six jours nous ont été accordés durant nos semaines de cours. Les deux premiers jours, nous avons commencé par nous répartir les tâches à l'aide de Trello, nous avons ensuite créé le MCD sur JMerise, puis Nicolas a créé toutes les classes qui définissent la structure de notre projet. Le MPD a ensuite directement été créé.

Entre le 2 novembre et le 13 décembre, Mathilde a créé le système de connexion, de déconnexion et d'inscription.

Du 2 au 13 décembre, Martin a commencé le CRUD pour la gestion des produits, et nous l'avons terminé ensemble le 17 novembre.

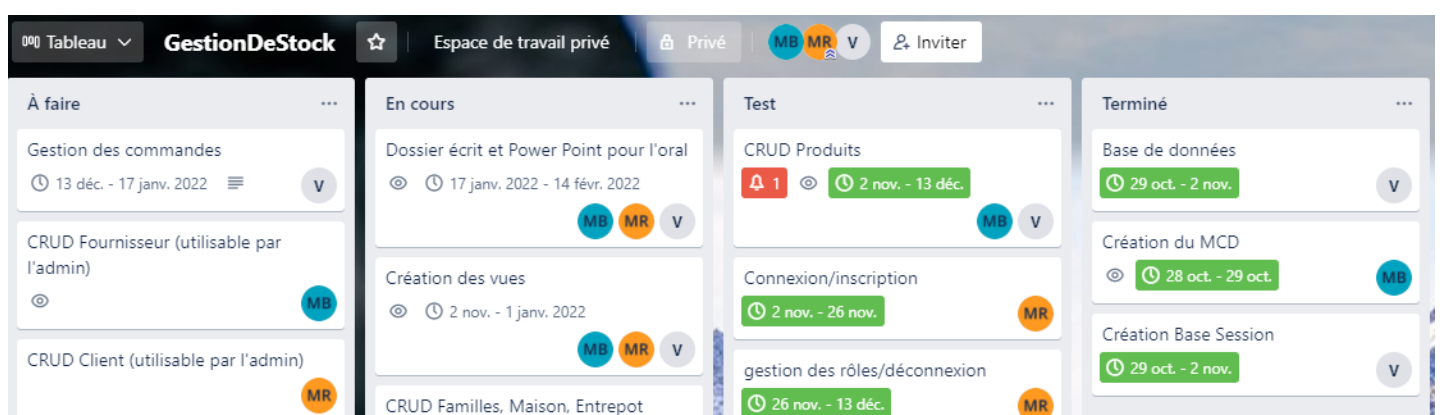
Les fonctionnalités qui seront développées dans les semaines qui suivent seront la création des vues, que nous avons déjà commencé mais que nous devons améliorer pour le 1er janvier.

Le CRUD des familles de vin, des maisons, des entrepôts, et des fournisseurs seront réalisés par Martin pour le 26 décembre.

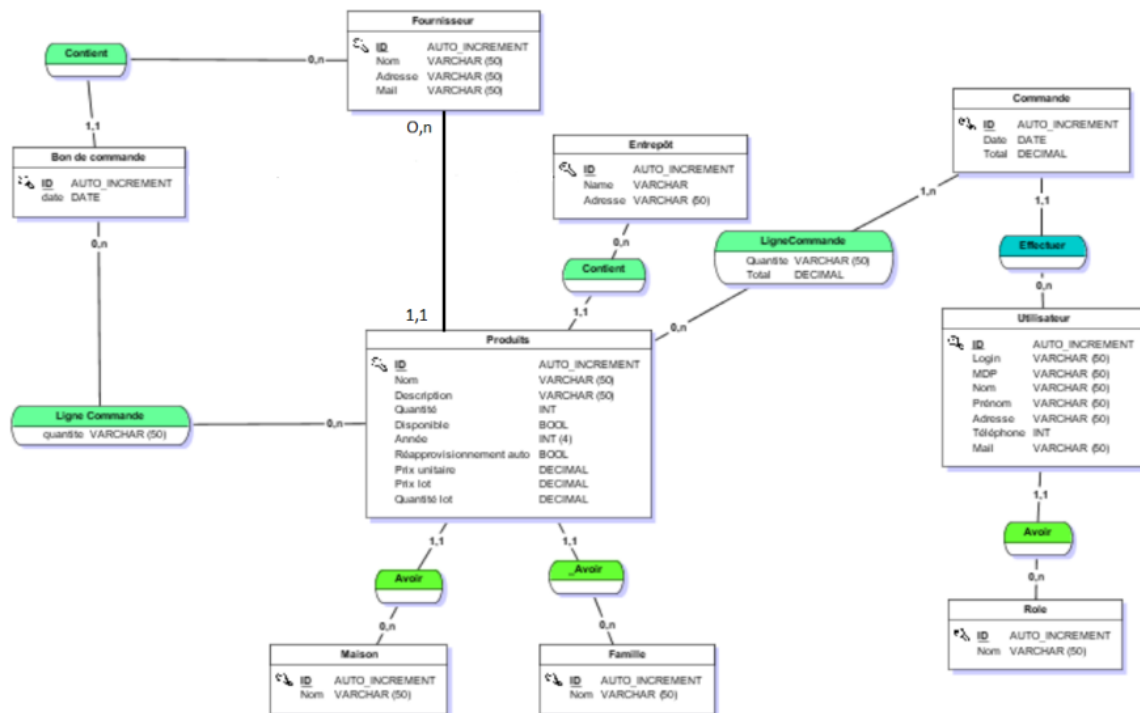
La gestion des commandes (possibilité d'achat et de création de panier) par les clients sera mise en place le 17 janvier par Nicolas.

Le CRUD des clients, et la gestion des droits (administrateur, client) sur les différentes pages seront effectués par Mathilde pour le 26 décembre.

L'envoi de bon de commande sera réalisé pour le 14 février.



3. Structure de la base de données



Vous pouvez voir ci-dessus le MCD, cela nous permet de mieux comprendre le projet et d'organiser notre base de données.

Nous pouvons voir que la table principale est la table produits, elle possède un Id, un nom, une description, sa quantité ainsi que sa disponibilité, son année de fabrication, une option de réapprovisionnement automatique, un prix unitaire ainsi que ses options de lot (prix et quantité dans un lot).

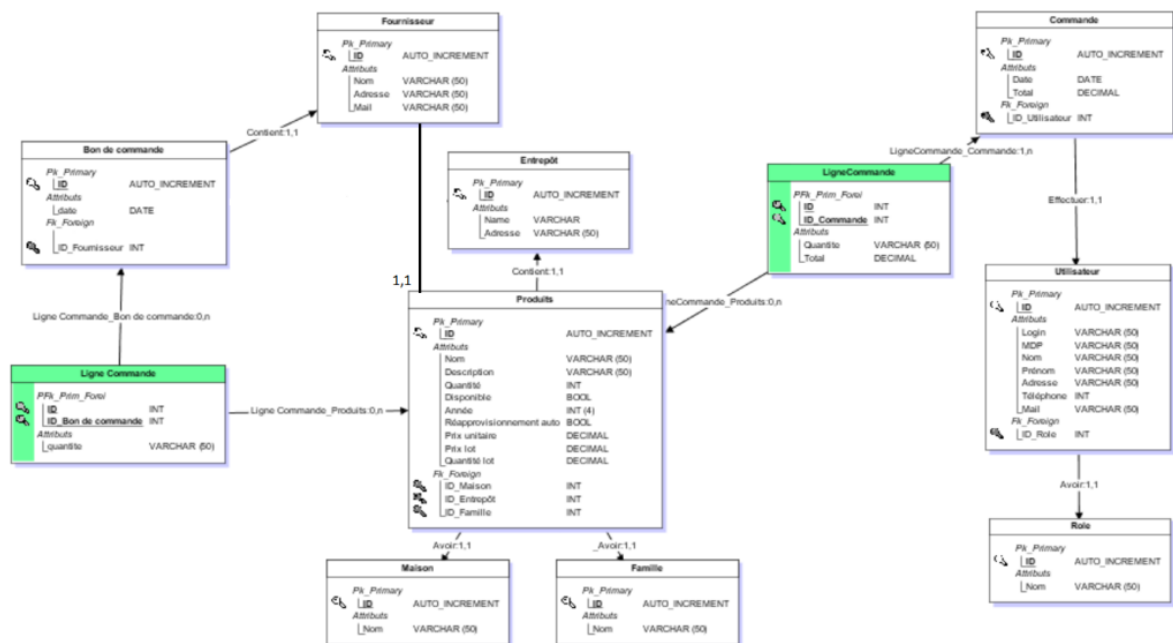
Cette table est reliée à deux autres tables, maison et famille. Ces tables possèdent toutes deux un id et un nom. Les produits peuvent avoir une famille et maison uniquement mais les maisons et familles peuvent avoir zéro ou plusieurs produits. Les produits possèdent aussi un bon de commande qui possède lui-même plusieurs lignes de commandes (chaque ligne à sa propre quantité) et les bons de commande contiennent un id et une date.

Ces mêmes bons de commande contiennent des fournisseurs constitués d'un Id, d'un Nom, d'une Adresse et d'un Mail. Ils contiennent aussi un entrepôt qui eux ont un Id, un nom et une adresse.

Les produits ont une table commande qui possèdent eux même une quantité de lignes de commande et d'un total. Ces commandes sont constituées d'un Id, d'une date et d'un total.

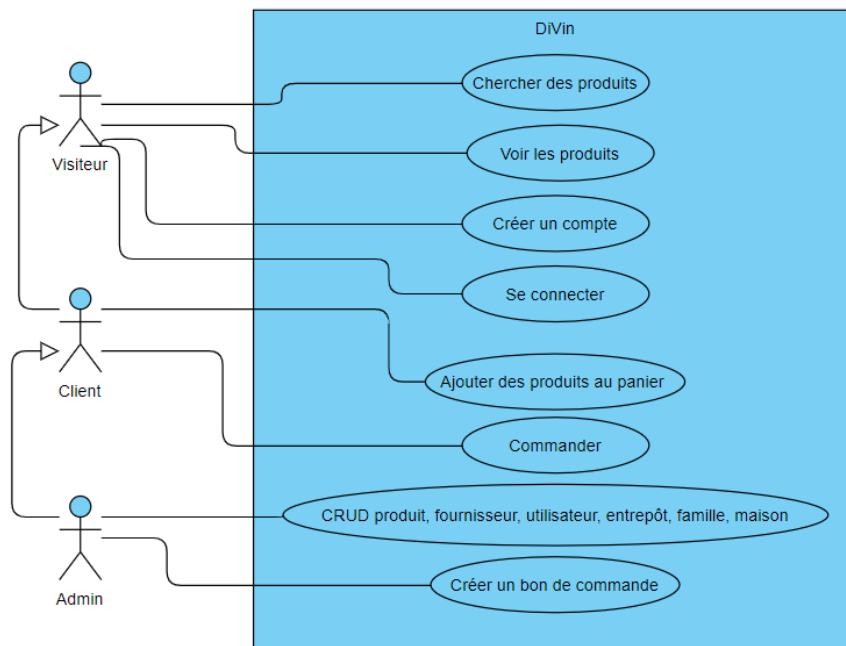
Ces commandes sont effectuées par un utilisateur qui possède lui-même un Id, un login, un mot de passe, un nom, un prénom, une adresse, un téléphone et un mail.

Ces utilisateurs possèdent un rôle qui contient un Id et un nom.

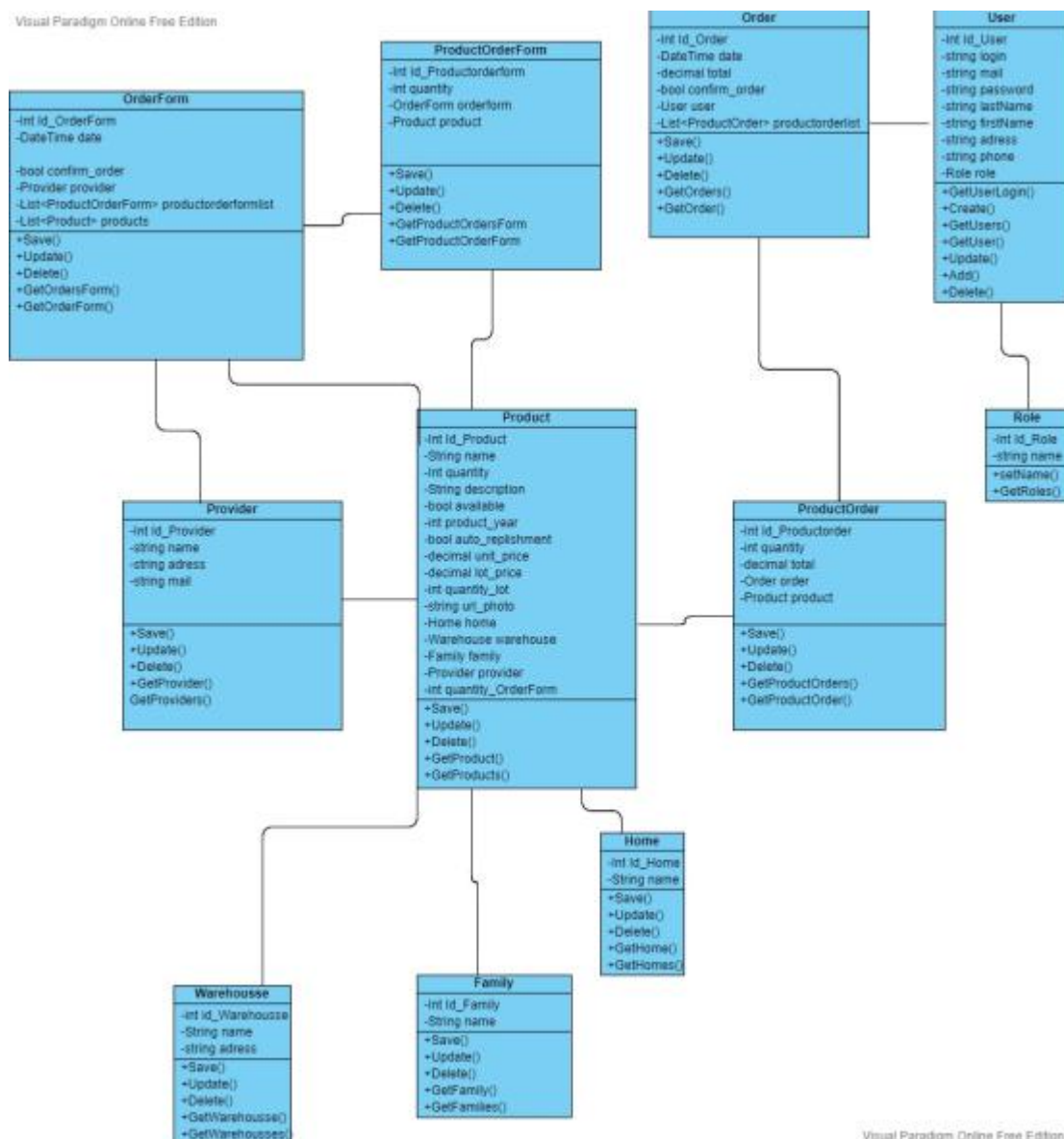


Modèle logique de données du site Annuaire

Sur ce modèle nous pouvons voir que les relations ont disparu et que selon les cardinalités définies dans le MCD, des « clés étrangères » sont apparues. La structure obtenue nous permet déjà de mieux nous rendre compte de ce à quoi va ressembler notre base de données finale. Il ne nous reste plus qu'à créer le code SQL pour mettre en place tout cela, et encore une fois Looping nous permet de générer le script SQL en un clic en choisissant le type de SGBD que nous utilisons, dans notre cas c'est une base de données MySQL.



Sur ce diagramme de cas d'utilisation nous pouvons voir qu'un visiteur aura la possibilité d'effectuer des recherches de produits et de trier ceux-ci et qu'il pourra afficher la fiche d'un produit. L'utilisateur pourra aussi se connecter, s'inscrire et une fois connecté, il pourra ajouter un produit au panier et commander ceux-ci. L'administrateur quant à lui aura la possibilité de gérer les CRUD produit, fournisseur, utilisateur, entrepôt, famille et maison. Il pourra aussi créer un bon de commande.



Sur ce diagramme nous pouvons voir les différentes classes du projet ainsi que leurs attributs et leurs opérations.

Par exemple : pour la classe « Provider », nous pouvons voir qu'elle prend l'attribut « name », « adress », « mail » qui sont des string et en opérations : Save(), Update(), Delete(), GetProvider() et GetProviders(). Ces opérations sont les fonctions qui seront utilisées par la classe « Provider »

4. Le code

Le code suit l'architecture MVC, il est versionné à l'aide de git et est bien commenté. Afin de suivre l'évolution de notre code ainsi que de le sauvegarder et le partager nous avons mis en place un dépôt GitHub :

<https://github.com/MathildeRenard/GestionDeStock>

Le code est en objet. Il est séparé en 5 dossiers. Un dossier controllers qui est composé des classes permettant de retourner des vues. Un dossier models qui contient les classes permettant de manipuler les valeurs (affichage, suppression, modification et lecture). Le dossier service est utilisé pour les logins. Un dossier viewmodel puis le dossier views qui permet d'afficher toutes les informations sur le site.

5. Problèmes rencontrés

Pour réaliser ce projet nous avons rencontré de nombreux problèmes.

Premièrement, nous avons eu du mal avec les CRUD étant liés au niveau de la base. Par exemple pour le CRUD utilisateur, il y a une colonne liée à la table role qui nous posait problème lors de la suppression. Travailler ensemble nous a aussi posé quelques problèmes mais avec une bonne organisation, l'utilisation de Trello ainsi que de nombreux messages pour faire le point, nous avons réussi à mener ce projet tous ensemble. Quelques soucis ont aussi été rencontrés au niveau de la commande. Par exemple, il nous fallait enlever du stock pour chaque commande ainsi que créer des bons de commande lorsque le produit n'était plus disponible.

6.Conclusion

Pour conclure, ce projet nous a permis de développer de nombreuses compétences dans de nombreux domaines. Que ce soit nos soft skills, en développant notre capacité à communiquer sur nos avancées, organiser des points ainsi que renseigner ce que l'on a réalisé tout comme gérer le projet, créer un planning et analyser celui-ci.

Mais nous avons aussi développé nos hard skills. Nous avons tous progressés en C# et le fait de travailler en groupe a permis aux gens qui maîtrisent le mieux cette technologie d'expliquer aux autres. Nous avons aussi progressé sur GIT en créant des branches pour chaque Features que nous faisons puis après nous la mergeons sur notre branche master. Nous avons aussi approfondi notre capacité à créer des MLD et à créer des bases de données propres qui ont du sens.

Nous avons quelques idées afin d'améliorer ce projet :

La première chose qui pourrait être améliorée dans le projet est la sécurité. Je n'ai pas à l'heure d'aujourd'hui les connaissances nécessaires pour sécuriser un site web.

La partie Front du site est à revoir, l'utilisation de Bootstrap facilite la chose mais cela reste très basique et peu esthétique.

Pour la base de donnée, nous pourrions ajouter une table adresse afin de ne pas les rentrer à chaque fois.

Pour le bon de commande, nous aurions aimé faire tout cela en une seule page ainsi que mettre un module de paiement ainsi que d'envoyer des confirmations par mail.

Avoir un profil utilisateur qui lui permettrait de modifier ses informations personnelles.