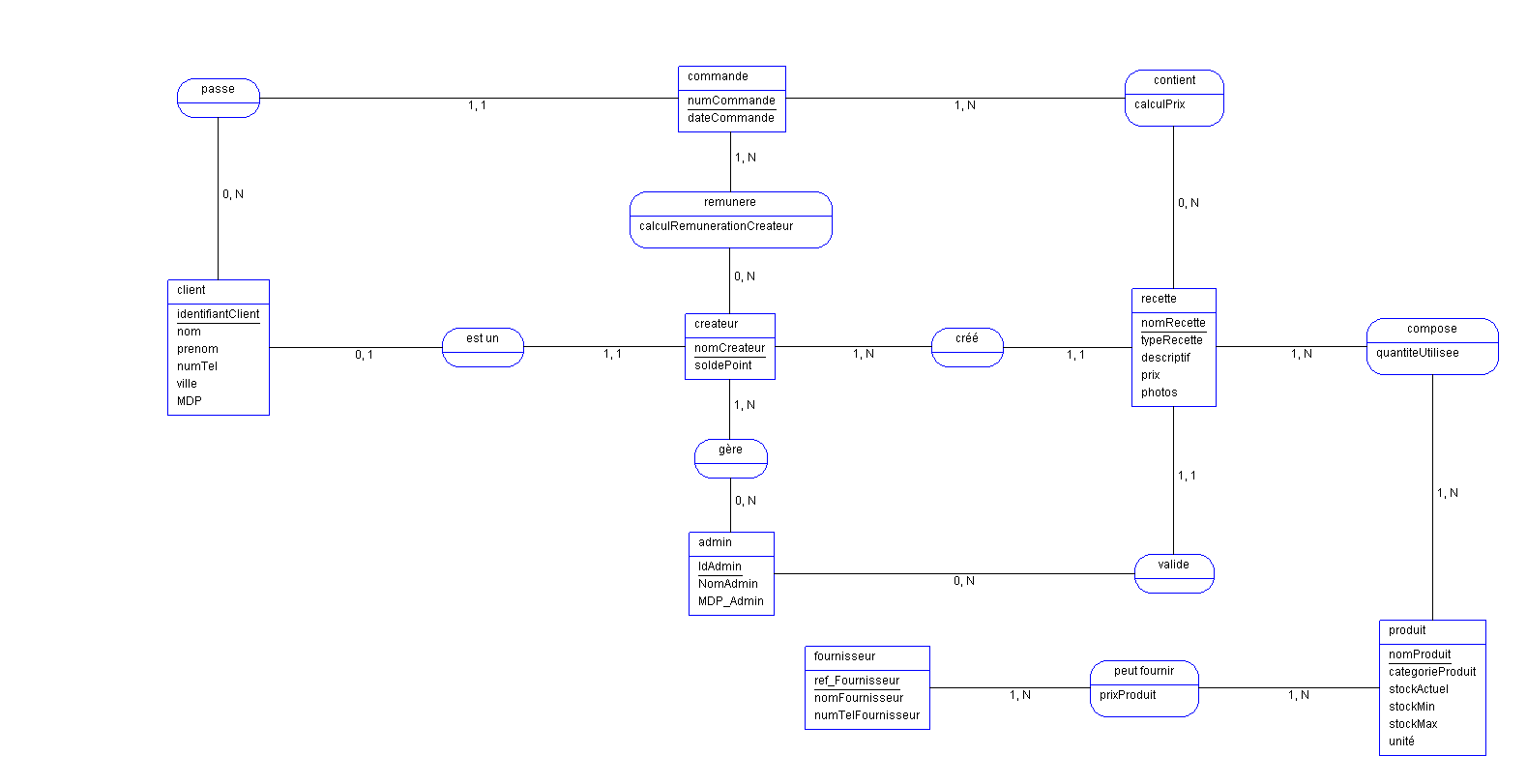
# Rapport Projet BDD Cooking – 10/05/2020

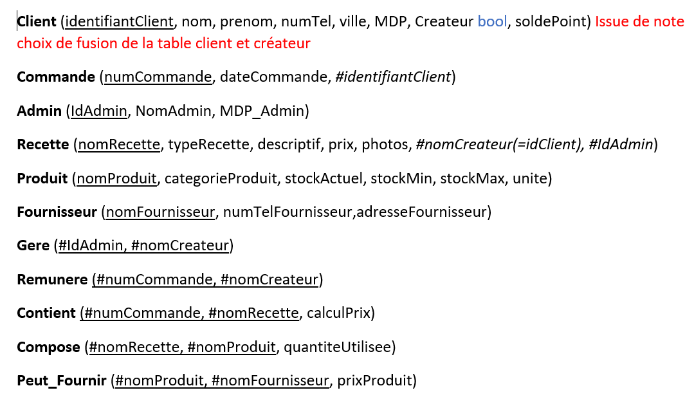
Réalisé par Victor Weill et Emilien Hahusseau ESILV A3 TDN

## Organisation de la base données

Suite à l’étude du cahier des charges et aux échanges avec notre chargé de td, nous avons décidé de faire une base de données la plus exhaustive possible afin de ne pas la surcharger et d’éviter des complications inutiles.

*Diagramme E/A BDD Cooking*

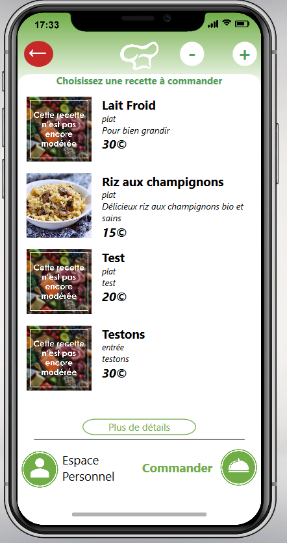
Pour le schéma relationnel, nous avons décidé de fusionner la table client et la table créateur. En effet, le créateur n’a qu’un seul attribut (soldePoint) de plus qu’un simple client. De plus on trouvait intéressant et pratique que le client est aussi un solde de points afin de pouvoir faire une sorte de cagnotte rechargeable avec des packages (10€ = 30PC), ici nous n’avions pas a nous occupé de cette partie. Pour fusionner ces deux tables nous avons juste créé l’attribut booléen Createur dans la table client pour savoir si ce dernier est un createur.



*Schéma Relationnel BDD Cooking*

## Codage de l’application Cooking en WPF sur C#

Premièrement nous avons décidé que l’affichage de notre projet Cooking serait sous la forme d’une application mobile. Nous avons donc créé l’ensemble des fenêtres avec un seul et même template. Nous avons beaucoup de fenêtres, c’est un choix, orienté UX, nous avons vraiment voulu que notre application soit le plus simple et accessible possible en nous inspirant des différentes applications de livraison à domicile.



Nous avons principalement utilisé des « ListView » dans des « Grid » pour l’affichage des listes de recettes, clients et ingrédients. Pour tout ce qui est changement de page ou action, nous avons utilisé des boutons. Afin de pouvoir utiliser les ListView, on devait forcément devoir passer par une classe afin d’afficher une liste d’élément de cette classe, une méthode plus simple sans créer doit surement exister, c’est l’un des points que l’on souhaiterait améliorer sur notre projet.

## Fonctionnalités Client :

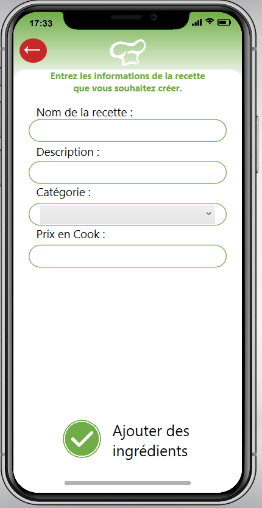
Nous avons réussi à créer et mettre en place l’ensemble des fonctionnalités demandées dans le cahier des charges. Il peut commander une ou plusieurs recettes, accéder à son espace client (petit plus il peut modifier ses informations personnelles), accéder à son espace créateur, créer une recette.

Nous avons décidé qu’un client devenait créateur à partir du moment ou il crée sa première recette. Ensuite si cette recette n’est conforme a nos attentes l’admin supprime la recette et cela entraine que le client ayant créé cette recette ne soit plus considéré comme créateur, sauf s’il avait déjà créé des auparavant bien évidemment.

Pour la partie, création de recette on a longtemps stagné sur un problème, on ne savait pas comment faire pour que le client puisse rentrer les quantités nécessaires d’ingrédients pour ça recette et récupérer ces informations. En effet, ceci doit être dû à la « ListView » qui affiche les produits, leur unité et une textbox dite template pour chaque produit dans laquelle le client devait remplir la quantité utilisée dans sa recette. Néanmoins, nous n’avons jamais réussi à avoir accès à cette TextBox.

Pour contourner ce problème, nous avons dans un premier temps, créer la recette sans les ingrédients, puis ensuite une nouvelle page s’ouvre et le client entre les ingrédients et les quantités souhaités. On récupère alors ces quantités avec un Selected, une TextBox et un Bouton. Le client valide la quantité à chaque fois pour chaque ingrédient selectionné. Lorsqu’il clique sur valide cela créer automatiquement la liaison « compose » entre le produit et la recette.

De cette solution il en ressort un problème toujours lié au fait que l’on n’a pas accès à la textBox de la quantité utilisée, le client ne voit donc pas quel produit il a ajouté, il faut donc faire attention à ne pas se tromper. C’est le deuxième point que l’on souhaitera améliorer dans notre projet.



*Ajout des ingrédients*

*Création de la recette*

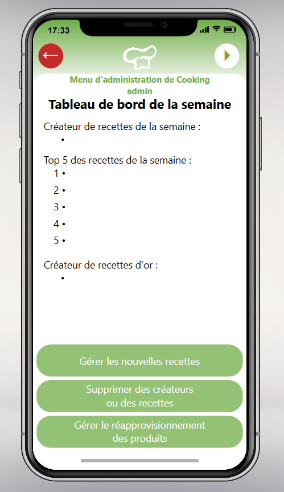
## Fonctionnalités Administrateur :

Pour répondre au cahier des charges et pour modérer l’application, nous avons dédié une partie entière du programme à l’espace Administrateur. Cet espace est séparé du reste des fonctionnalités de l’application mais passe par la même page de connexion. Elle se base sur un Id et un mot de passe stockés dans la table admin de notre base de données.

L’organisation de la page reprend les mêmes codes que le reste de l’application. Cette partie de l’application a nécessité plus de réflexion sur les requêtes SQL afin de générer le tableau de bord demandé et permettre à l’administrateur de modifier des informations ou de supprimer un client ou une recette.

Malheureusement, nous avons été débordés par le temps et nous n’avons pas pu finaliser la fonction pour réaliser un fichier de commande en .XML. Par conséquent ce bouton de la page administrateur est désactivé.

Le mode démo est disponible en haut à droite dans le bouton blanc. Il donne accès à une nouvelle fenêtre avec les infos demandées dans le cahier des charges ainsi qu’un bouton qui remplace la saisie au clavier pour avancer parmi les étapes. Nous avons préféré cette méthode d’affichage pour améliorer l’expérience utilisateur et la rendre plus naturelle à l’utilisation.



*Mode démo*

*Interface administrateur*

## Organisation de la base données

Ce projet nous a permis de considérablement approfondir nos connaissances en WPF du premier semestre. Nous sommes désormais capable de réaliser une application WPF esthétique et « user friendly » utilisant une base de données que nous avons-nous même designer et créer. Nous avons principalement rencontré des difficultés dans l’affichage et la récupération des données d’entrée en WPF. Ce projet était très enrichissant et motivant, nous avions sans cesse envie d’améliorer la qualité de notre interface graphique ainsi que de rajouter de nouvelles fonctionnalités pour l’utilisateur.