**Rapport du Mini-Projet**

XMHelloFresh

Andres lucie – Froment Lorenzo

Table des matières

[I. XML et XML schéma 1](#_Toc161305445)

[Construction de la base de données 1](#_Toc161305446)

[Avantages et inconvénients 2](#_Toc161305447)

[II. Scénarios 2](#_Toc161305448)

[Scénario 1 2](#_Toc161305449)

[Scenario 2 3](#_Toc161305450)

[Scénario 3 3](#_Toc161305451)

[Scénario 4 3](#_Toc161305452)

[Scenario 6 4](#_Toc161305453)

# I. XML et XML schéma

## Construction de la base de données

Pour ce projet nous nous sommes d’abord penchés sur la création d’un XML schéma ayant une structure cohérente et complète pour que celle-ci soit en accord avec les besoins du projet.  
Nous avons commencé par noter les éléments nécessaires au projet, en voici une liste : Utilisateurs, Livreurs, Commandes, Commentaires, Menus, Recettes, Ingrédients  
  
Nous avons donc commencé par mettre ces éléments dans notre document xsd et ensuite nous les avons pris chacun un par un tout en nous demandant de quoi ils devraient être composés.  
  
Par exemple, pour les ingrédients, nous avons pris la décision de les composer bien évidemment de plusieurs « ingrédient » individuels dotés d’un id (pour éviter la redondance). Cet élément Ingrédient est un xs :complexType, il est donc composé de plusieurs éléments, et également c’est un xs :sequence, ce qui signifie qu’on veut garder l’ordre spécifié des éléments de « ingrédient ».  
Ces éléments sont, un nom (sous forme d’un string), une quantité (sous forme d’un integer), une unité (sous forme de string).

Voici sa structure :

Une image contenant texte, capture d’écran, Police

Description générée automatiquementAprès avoir fait cela avec chaque élément, notre xsd était terminé et nous avons pu construire notre XML. Nous avons essayé de le remplir de quelques données pour avoir une base de données permettant de travailler sur la suite du projet en construisant des visualisation grâce aux xslt.

## Avantages et inconvénients

Un des avantages de notre schéma xml est qu’il est bien détaillé, avec des id attribués à la majorité des éléments permettant d’éviter beaucoup de redondance, que ce soit pour les ingrédients dans les recettes ou bien les utilisateurs auxquels on fait référence dans les commandes ou bien dans les commentaires.

Cette structuration bien ordonnée permet également des modifications simples et rapides. Par exemple au cours du projet, nous nous sommes rendu compte que l’élément recette manquait d’étapes à suivre. En quelques minutes nous avons pu les rajouter sans aucune difficulté.

Une chose qui pourrait être un inconvénient mais que nous trouvons pertinente est le fait que pour les ingrédients, il est possible d’ajouter par exemple 2 ingrédients « sucre » avec 2 quantités différentes :

Une image contenant texte, Police, capture d’écran, conception

Description générée automatiquementUne image contenant texte, Police, capture d’écran

Description générée automatiquement

Ceci nous parait pertinent car nous jouons le rôle d'une entreprise de "boîtes à recettes", ces boîtes vont donc être garnis d’ingrédients avec les bonnes quantités. Donc si une recette a besoin de 50g de sucre, il n’y aura pas besoin de se compliquer la tâche, il suffira de prendre l’ingrédient avec l’id 16 qui correspond à un sachet de 50g de sucre.

# II. Scénarios

## Scénario 1

« Un utilisateur se connecte à notre site et souhaite voir les recettes qu’il pourrait faire/aimer en voyant les images et les ingrédients »

Pour ce scénario nous avons choisi de faire un simple affichage des recettes avec leurs ingrédients pour que les utilisateurs puissent avoir un aperçu des plats auxquels ils peuvent avoir accès en commandant chez nous. Ce scénario (comme les suivants) utilise évidemment xs :key pour créer des index utilisant les @id. Dans ce scénario, cela permet par exemple de faire référence à des ingrédients qui ne sont pas directement écrits dans les recettes à l’aide de l’index ingredientKey.

## Scenario 2

« Un utilisateur plutôt récent sur notre site, souhaite avoir une idée des menus proposés et de leur qualité. Il regarde donc les commentaires des utilisateurs en fonction de leur notation et du menu commandé »

Pour ce scénario, nous affichons les commentaires. Pour rendre cela un peu plus ordonné nous utilisons des for each dans le but d’afficher tout d’abord tous les commentaires ayant eu une note de 5⭐ puis puis ceux ayant eu 4⭐ etc…  
On utilise ici 2 xs :key pour avoir le nom du menu associé au commentaire ainsi que l’id de la commande. Nous affichons également le nombre total de commentaires postés via l’utilisation de la méthode count().

## Scénario 3

« Un gestionnaire de livraison souhaite accéder facilement aux détails des commandes par livreur. Pour répondre à ce besoin, une feuille de style XSLT a été développée pour extraire et organiser ces informations à partir d'un fichier XML. »

Cette feuille de style utilise des index XSLT pour référencer les différentes entités telles que les commandes, les menus, les recettes, les utilisateurs et les livreurs. Lorsque la feuille de style est appliquée au fichier XML, elle génère un fichier HTML présentant les détails des commandes regroupées par livreur. Chaque livreur est identifié, suivi de la liste des commandes qui lui sont attribuées, avec les informations telles que l'identifiant de la commande, le nom de l'utilisateur, la date, l'heure, l'adresse de livraison, le statut de la commande et le prix.

Cette interface facilite la gestion des livraisons en permettant aux gestionnaires de visualiser rapidement les informations nécessaires.

## Scénario 4

« Un gestionnaire de livraison souhaite accéder facilement aux adresses où les livreurs ont déjà effectué au moins une livraison. Pour répondre à ce besoin, une transformation XSLT a été mise en place pour extraire et organiser ces informations à partir d'un fichier XML. »

La feuille de style XSLT prend comme entrée le fichier XML contenant les données sur les livreurs et les commandes effectuées. Lorsque cette feuille de style est appliquée au fichier XML, elle génère un nouveau XML structuré pour afficher les informations sur les livreurs et les adresses où ils ont déjà livré.

Chaque livreur est représenté dans le nouveau XML avec son nom, prénom et une liste des adresses de livraison effectuées. Pour chaque livraison, la date, l'heure et l'adresse sont incluses.

Cette transformation permet aux gestionnaires de visualiser rapidement les adresses où chaque livreur a déjà effectué des livraisons, facilitant ainsi la gestion logistique des livraisons.

Scénario 5

« Un administrateur souhaite accéder aux données des recettes et des menus disponibles sur la plateforme XMHelloFresh. Pour répondre à cette exigence, une transformation XSLT a été développée pour extraire et organiser ces informations à partir d'un fichier XML. »

La feuille de style XSLT prend comme entrée le fichier XML contenant les données sur les recettes et les menus. Lorsque cette feuille de style est appliquée au fichier XML, elle génère un nouveau format JSON structuré pour afficher les informations sur les recettes et les menus disponibles.

Chaque recette est représentée dans le JSON avec son nom, le nombre de personnes pour lequel elle est destinée, une liste des ingrédients nécessaires avec leur quantité et unité, les étapes de la recette, et l'URL de l'image associée. De même, chaque menu est représenté avec son nom, son prix, et une liste des recettes incluses.

Cette transformation permet à l'administrateur d'accéder facilement aux données des recettes et des menus dans un format JSON, facilitant ainsi la gestion et l'utilisation de ces informations dans d'autres applications ou services.Haut du formulaire

## Scenario 6

« Un administrateur souhaite vérifier les données d’un utilisateur se plaignant d’une livraison à la mauvaise adresse. Il a donc créé un programme python lui affichant ces données et peut résoudre ce mystère »

Pour ce dernier scénario, nous avons pris la décision de programmer en python. Nous avons codé ce programme python pour qu’il utilise les données xml et les transforme en un fichier html. Plus précisément, l’output est un fichier html affichant les utilisateurs de XMHelloFresh ainsi que leurs informations. Cette page est bien évidemment réservée aux administrateurs du site.

Ce programme est simple, il import xml.etree.ElementTree qui permet de traiter les xml en python.  
Ensuite, depuis la racine, nous nous rendons dans la branche des utilisateurs ce qui nous permet ensuite de récupérer tous les éléments d’un utilisateur.   
Grâce à cela on peut ensuite insérer ces informations dans le format html et générer l’output.