

Rapport mini-projet Base de Données

Tristan LARGUIER

14/03/2024

1 Hypothèses de travail

Pour réaliser ce projet, j'ai fait les hypothèses suivantes :

- Les recettes sont des plats uniques, et ne sont pas des menus.
- Les commandes contiennent une seule recette.
- Les commentaires sont liés à une commande, et non à une recette.

2 Réalisation des objectifs du projet

2.1 Écriture du schéma XML

Le schéma XML est disponible dans le fichier `model.xsd`. Il est composé de 7 éléments principaux :

- **recettes** : contient la liste des recettes, avec les informations de santé, les ingrédients, et les instructions de préparation.
- **commandes** : contient la liste des commandes, avec les informations sur les clients, les livreurs, les recettes commandées, les dates de livraison, et les détails de commande.
- **commentaires** : contient la liste des commentaires des clients, avec les informations sur les clients, les recettes, les notes, et les commentaires.
- **clients** : contient la liste des clients, avec les informations sur les clients.
- **livreurs** : contient la liste des livreurs, avec les informations sur les livreurs.
- **ingredients** : contient la liste des ingrédients, avec les informations sur les ingrédients.
- **unites** : contient la liste des unités de mesure, avec les informations sur les unités de mesure.

Chaque élément est identifié par un attribut `id` unique, sous forme d'une chaîne de caractères. Le premier caractère de l'`id` est une lettre, et les caractères suivants sont des chiffres. La lettre permet de distinguer les différents types d'éléments. Les recettes ont un `id` commençant par `R`, les commandes par `C`, les commentaires par `M`, les clients par `P`, les livreurs par `L`, les ingrédients par `I`, et les unités de mesure par `U`.

Les documents XML sont tous les mêmes pour l'ensemble des scénarios du projet, seuls les feuilles de style XSLT changent pour adapter l'affichage des données.

2.2 Scénario 1 : Afficher la liste des recettes avec leurs détails

Ce premier scénario permet d'afficher la liste des recettes avec les images, informations de santé, ingrédients, et instructions de préparation, comme un utilisateur pourrait le voir sur un site web de recettes. Cette page contient une feuille de style CSS pour améliorer la présentation.

2.3 Scénario 2 : Afficher la liste des commandes avec les détails des clients et des livreurs

Ce scénario permet d'afficher la liste des commandes en cours, avec les informations sur les clients, les livreurs, les recettes commandées, les dates de livraison, et les détails de commande. Cette page étant destinée à un utilisateur interne, la présentation est plus sobre. Lorsque l'on clique sur le nom d'une recette, on est redirigé vers la page des recettes dédiée à la recette concernée.

2.4 Scénario 3 : Afficher la liste des commentaires associés aux commandes

Ce scénario permet d’afficher la liste des commentaires des clients, avec les informations sur les clients, les recettes, les notes, et les commentaires. La note est affichée récursivement avec des petites étoiles. Malgré sa destination à des utilisateurs externes, la présentation est sobre. Lorsque l’on clique sur le nom d’une recette, on est redirigé vers la page des recettes dédiée à la recette concernée, comme dans le scénario précédent.

2.5 Scénario 4 : Afficher la liste des recettes avec la moyenne des notes des clients

Ce scénario a pour objectif de compléter le scénario 1 en affichant la moyenne des notes des clients pour chaque recette. Les informations sont renvoyées au format XML. Les données de sortie suivent le schéma suivant :

```
<RECETTES>
  <RECETTE>
    <NOM>...</NOM>
    <NOTE>...</NOTE>
  </RECETTE>
</RECETTES>
```

Ce format minimaliste permet d’accéder directement à l’information recherchée, le nom de la recette pourra facilement être remplacé par l’id de la recette si besoin pour lier à la page du scénario 1.

2.6 Scénario 5 : Afficher la liste des recettes en JSON

Ce scénario a pour objectif d’exporter en JSON la liste des recettes avec les images, informations de santé, ingrédients, et instructions de préparation. Ce scénario pouvant correspondre à une API REST, tous les ID internes sont remplacés par les noms des produits et des quantités. Les informations sont renvoyées au format JSON.

2.7 Scénario 6 : Implémenter un programme Python pour obtenir les recettes que l’on peut faire avec les ingrédients que l’on a dans le frigo

Ce scénario a pour objectif d’implémenter un programme Python pour obtenir les recettes que l’on peut faire avec les ingrédients que l’on a dans le frigo. On peut l’utiliser en lui fournissant une liste d’ingrédients (les noms au format string), et il nous renvoie la liste des noms des recettes que l’on peut faire avec ces ingrédients. Si aucune recette n’est possible, la liste retournée est vide. On peut aussi lui donner le nom d’une recette pour qu’il nous renvoie les ingrédients nécessaires pour la réaliser.

3 Détails d’implémentation

3.1 Utilisation de la récursivité

Pour afficher les notes des recettes dans le scénario 3, j’ai utilisé la récursivité pour afficher les étoiles. Cela est permis par le code suivant :

```
<!-- Template for generating stars based on the note -->
<xsl:template name="generateStars">
  <xsl:param name="note"/>
  <xsl:param name="counter" select="1"/>
  <!-- Condition to check if counter is less than or equal to note -->
```

```

<xsl:if test="$counter <= $note">
  <!-- Output a star -->
  <xsl:text disable-output-escaping="yes">&#9733;</xsl:text>
  <!-- Recursive call to generate next star -->
  <xsl:call-template name="generateStars">
    <xsl:with-param name="note" select="$note"/>
    <xsl:with-param name="counter" select="$counter + 1"/>
  </xsl:call-template>
</xsl:if>
</xsl:template>

```

3.2 Calcul de la moyenne des notes

Pour calculer la moyenne des notes des recettes dans le scénario 4, j'ai dans un premier temps sélectionné les avis des clients pour chaque recette (un avis étant lié à une livraison et une livraison étant liée à une recette), puis j'ai calculé la moyenne de ces notes.

```

<xsl:template match="RECETTE">
  <recette>
    <xsl:attribute name="id">
      <xsl:value-of select="@id"/>
    </xsl:attribute>
    <nom>
      <xsl:value-of select="NOM"/>
    </nom>
    <xsl:variable name="related_comments"
      select="//COMMENTAIRE[COMMANDEID=//COMMANDE[RECETTEID=current()/@id]/@id]"/>
    <moyenne>
      <xsl:value-of select="sum($related_comments/NOTE) div count($related_comments)"/>
    </moyenne>
  </recette>
</xsl:template>

```