

Rapport de Projet

Pour un maraicher:

“ archiver tout son stock à l'aide d'un dataweb XML ”

Par : Abdoulaye Oumar LY & Moussa Tine
Professeur : Mouhamadou Thiam

Sommaire:

1/ Présentation de la problématique

2/ Elaboration du projet

3/ Résultats

1/ Présentation de la problématique

Le but de ce projet est de mettre en pratique les connaissances théoriques acquises aux cours de web sementique. Dans ce sens donc, il est proposé de réaliser un dataweb xml pour un maraicher qui voudrait passer à l'informatique pour archiver son stock.

À partir de ce constat, nous allons donc essayer de réaliser une application fonctionnelle, remplissant les conditions imposées dans l'énoncé.

2/ Elaboration du projet

2.1/ Outils utilisés

Pour mettre en œuvre ce projet, nous avons utilisés différents technologies même si généralement tout tourne autour du langage de balisage XML.

- Des documents XML pour archiver le stock du maraicher et les commandes des clients.
- Des entités DTD (Document Type Définition) pour gérer la conformité des documents XML
- Du HTML/CSS/Bootstrap via des feuilles de style XSLT pour présenter les données dans un navigateur web.
- Du PHP à travers DOM pour traiter les formulaires générés avec XSLT et écrire sur les documents XML.
- Netbeans comme IDE de développement
- Apache2 comme serveur web pour exécuter le code PHP.

2.2/ Organisation du répertoire du projet

Dans le répertoire principale du projet, on retrouve 7 dossiers:

- ➔ css: contient les fichiers css
- ➔ dtd: on n y retrouve les fichiers:
 - « commandes.dtd » : défini le document XML (xml/commandes.xml) contenant les commandes.
 - « stock.dtd » : défini le document XML (xml/stock.xml) contenant les produits du maraicher.
- ➔ img: contient les images utilisés dans le projet
- ➔ lib: contient les librairies utilisées dans ce projet à savoir bootstrap
- ➔ php: contenant les scripts php permettant le traitement des formulaires
 - « ajouterProduit.php » : traite le formulaire d'ajout de produit généré par « xslt/ajouterProduit.xsl ». Elle est destiné donc aux maraicher.
 - « commander.php » : traite les commandes effectuées par les clients.
 - « modifier_produit.php » : traite les modifications effectués sur le stock par le maraicher.

- ➔ xml : contient les documents xml destinés à contenir des informations sur le stock (stock.xml) et sur les commandes (commandes.xml) ainsi que ajouterProduit.xml qui ne contient pas de données mais permet d'ouvrir le formulaire contenu dans ajouterProduit.xsl.
- ➔ xslt : contient les feuilles de styles xsl transformant les fichiers xml en html.

2.3/ Diagramme

- Diagramme de cas d'utilisation

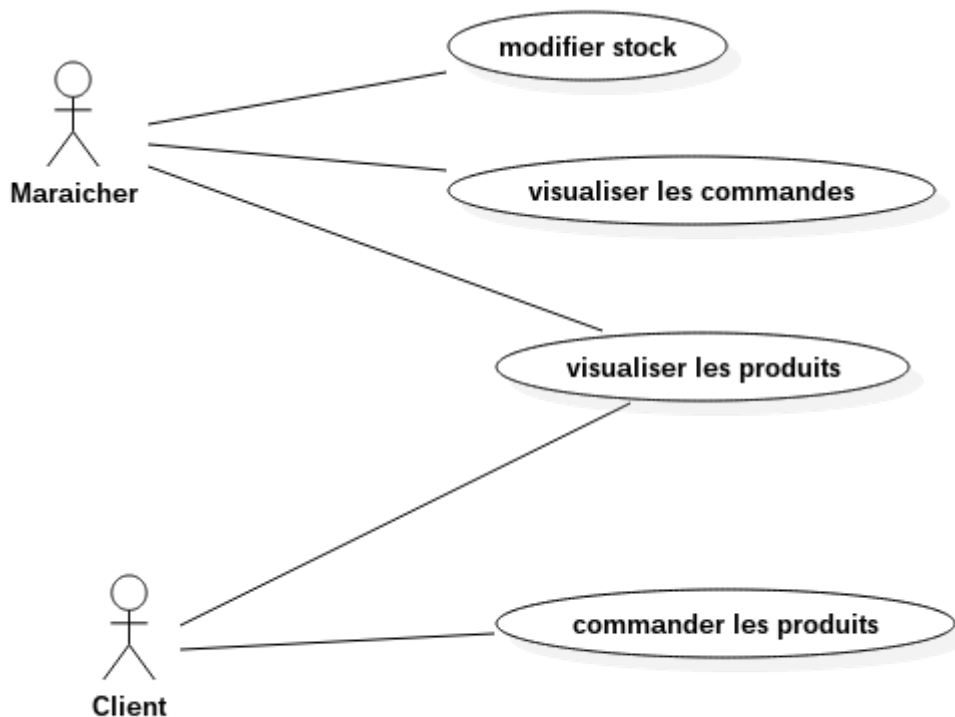


Illustration 1: diagramme de cas d'utilisation

Commentaire :

Le dataweb a deux utilisateurs que sont :

- Le maraîcher qui peut principalement effectué des opérations tels que modifier le stock, visualiser les commandes et visualiser les produits.
- Le client qui pourra visualiser les produits pour effectuer des commandes.
- Lien entre les fichiers xml, xst et php

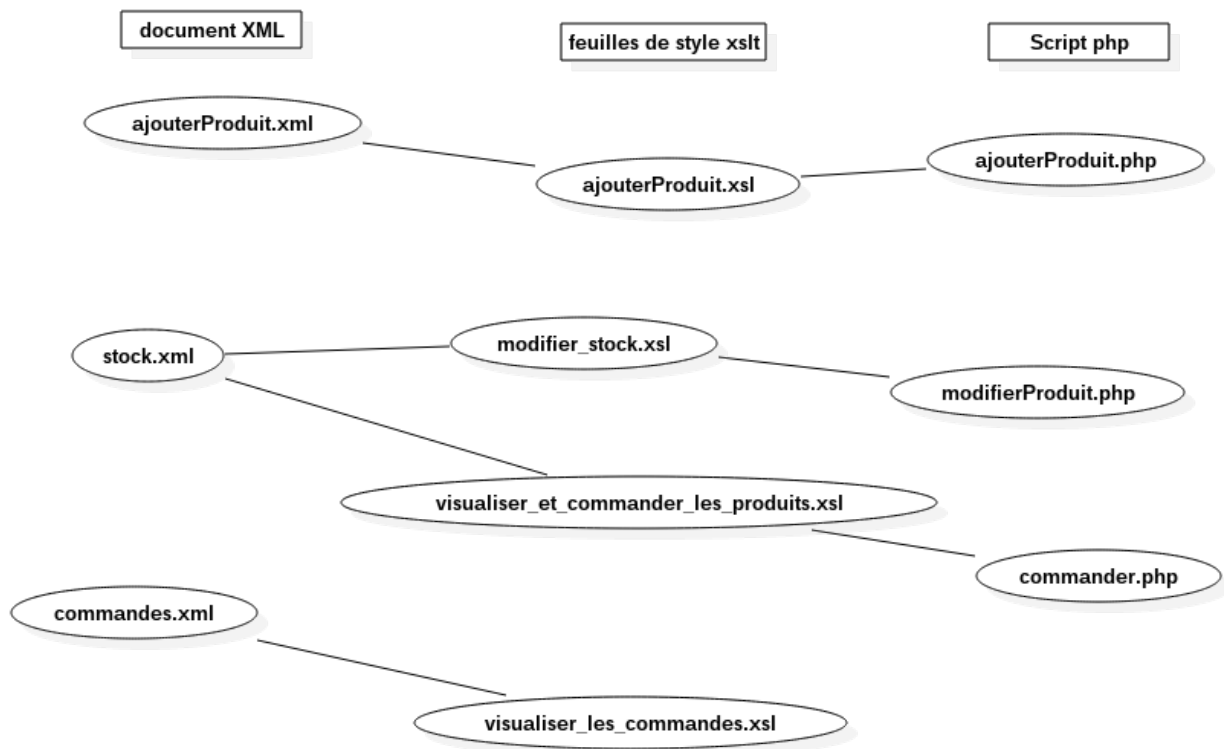


Illustration 2: Lien qui existe entre les fichiers

Commentaire :

Ce diagramme montre les fichiers qui ont des liens entre elles.

NB : Voyez qu'il le fichier stock.xml peut avoir un lien avec modifier_stock.xsl (pour modifier le stock) ou avoir un lien avec visualiser_et_commander_les_produits.xsl. Par conséquent, il faudra changer l'entête du fichier stock.xml en fonction de la tâche que l'on souhaite faire.

3/ Résultats

A la fin donc de ce projet, nous avons réussi à élaborer une petite application d'un dataweb fonctionnel malgré quelques difficultés qui ont survenues. Les commandes des clients sont bien stockées, les modifications et alimentations de stock sont aussi bien effectuées et en plus le maraîcher peut bien visualiser les commandes ...

Veillez trouver ci-dessous quelques captures d'écran de l'application.

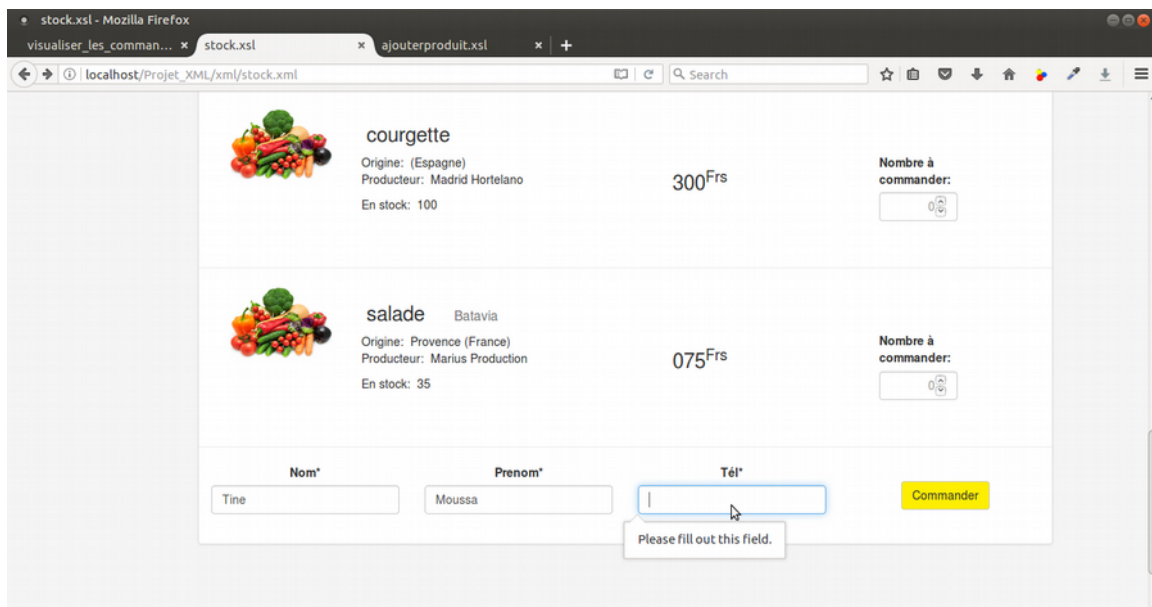


Illustration 3: afficher liste des produits et commander

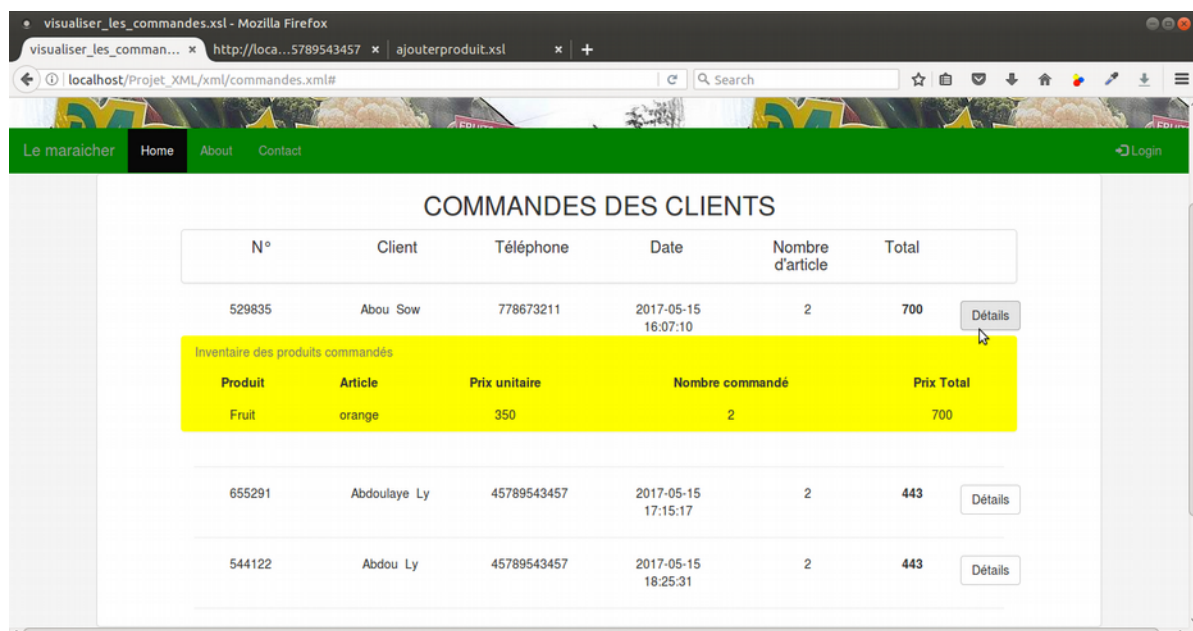


Illustration 4: Afficher la liste des commandes

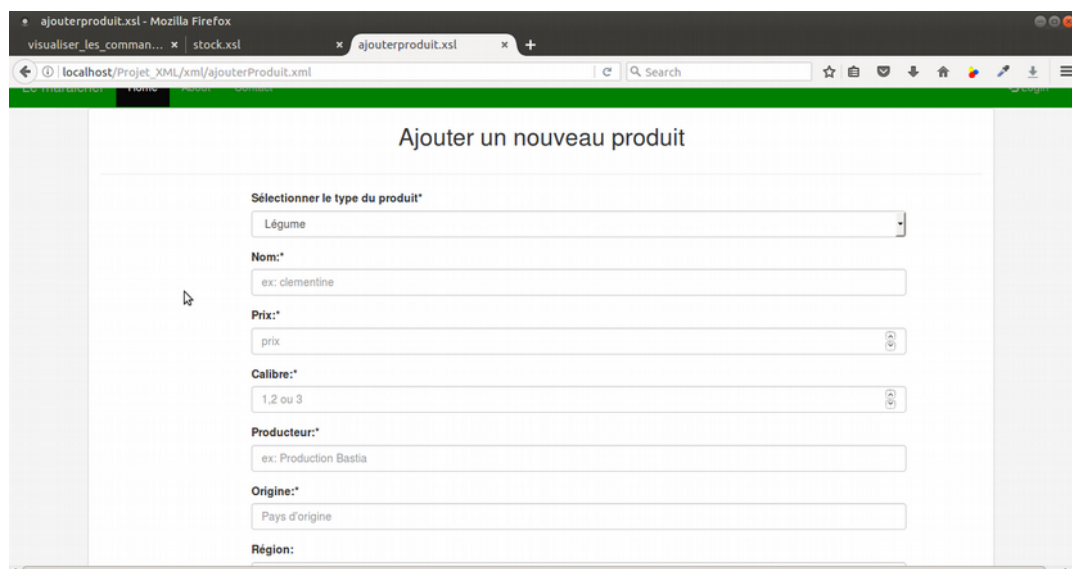


Illustration 5: Ajouter un produit

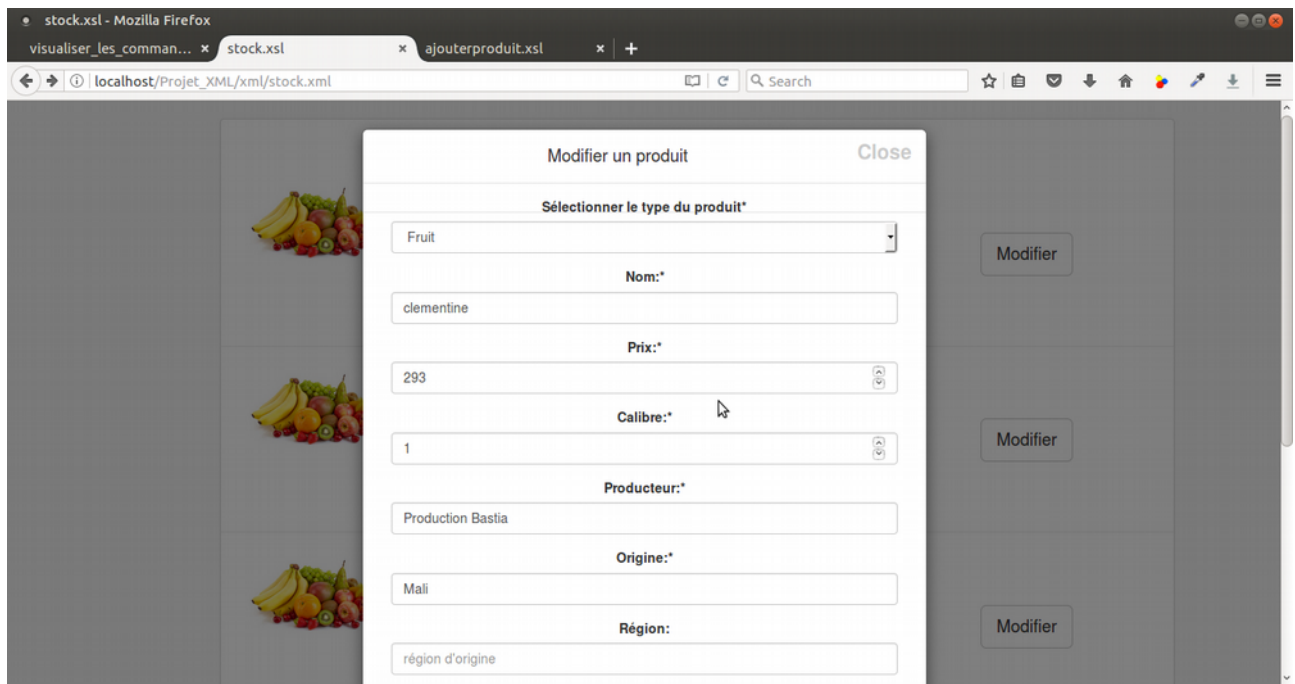


Illustration 6: Modifier un produit