RAPPORT DE STAGE

SMD SARL

Soudeuses automatiques à soudure de fond pour fabrication de sacs et sachets polyéthylène (PEBD/PELD/PEHD)

Stage de 5 semaines

ENTREPRISE SMD

17 Zone Industrielle
61300 L'AIGLE

t. 02.32.34.51.24

Table des matières

Remerciements 4
Domaines d'activité 4
Histoire de SMD 4
Fondateurs 4
Clients 5
Leurs Offres 5
Formations 6
Base de données de l'entreprise 7
Projet 9
Environnement Debian 9
Particularité 9
Quelques logiciels 10
Modèle MVC: 11
PgAdmin III 12
Ajouter des données 13
Créer une clé primaire /étrangère 13
Avantage PgAdmin III 14
SQL / No SQL 14
Modèle sans relation : 15
Modèle Merise : 15
Diagramme de classes : 15
Présentation Perl 16
Mieux connaitre Perl avec Moose 17
Utilisation 18
Fichier CSV 18
Tester l'existence d'un fichier et l'ouvrir 19
Lire ligne par ligne et assimilé des variables 19
Additionner la variable Qté et table de hashage 19
Fermer le fichier et afficher le contenu 20

Framework Dancer2 ____ 20
Dancer2 ____ 21
Template ____ 21
Template Toolkit ____ 22
Route par défaut ____ 22
Route ____ 23
Envoyer une variable à un Template ____ 23
Résumé : ____ 23
Se connecter à une base de données ____ 24
AJAX dans Dancer2 ____ 24
Vue ____ 26
Conclusion : ____ 27

Remerciements

Je souhaite remercier toute l'équipe de SMD, qui m'a particulièrement aidé lors des cinq semaines de stage. En particulier mon maître de stage Hervé Desrues qui a pris le temps de m'expliquer certaine notion que je n'avais pas appris auparavant.

Domaines d'activité

Le domaine d'activité de l'entreprisse SMD est la construction et réparation de machines pour l'industrie du plastique et de l'emballage. Plus précisément pour la fabrication de sacs, sachets et housses en plastique de toutes dimensions pour l'agro-alimentaire, le pharmaceutique ou l'industrie en général, la logistique, le bâtiment etc. La clientèle est essentiellement composée de PME et de quelques groupes industriels, situés principalement en France, Allemagne, Royaume-Uni, et autres pays d'Europe ainsi qu'en Afrique, au Moyen-Orient, DOM-TOM etc.

Histoire de SMD

Les machines dont nous nous occupons ont été inventées en 1968 et ont été commercialisées sous la marque ARVOR. Le succès a été immédiat du fait d'une très grande facilité d'emploi et de la large palette de produits pouvant être fabriqués grâce à ces machines. Il y a eu environ 8000 machines installées de par le monde et la plupart sont encore en fonctionnement à ce jour.

La fabrication était assurée par la société SOFAMECA depuis les années 1970, chez qui cela représentait la moitié du chiffre d'affaires environ. SOFAMECA a pris le contrôle d'ARVOR en 1983 et pour finir la société ARVOR a été absorbée par SOFAMECA en 1995 lors du regroupement des activités commerciales et de fabrication sur le site de Dorceau (61).

Faute de vision d'avenir chez ses dirigeants d'alors, la SOFAMECA est ensuite passée entre les mains de différents repreneurs assez peu scrupuleux pour finir par fermer définitivement au début 2010. C'est alors que SMD a racheté les stocks, les encours, les plans et repris du personnel de SOFAMECA pour continuer à assurer le service après-vente.

Cela a fonctionné plutôt bien et nous avons très rapidement récupéré une bonne partie de la clientèle. SMD avait démarré en 1996 et avait alors pour activité principale la vente de pièces détachées à l'export, déjà dans le même domaine.

Fondateurs

Catherine Sablé, responsable du SAV chez ARVOR/SOFAMECA de 1983 à 1992 et **Hervé Desrues**, technicien bureau d'études/SAV chez ARVOR de 1990 à 1992, rejoints en 2010 par **Grégory Tiriakian**, technicien chez SOFAMECA depuis 2000.

Clients

France	**GuerinPlastiques**	http://www.guerin-plastiques.com/
Allemagne	Roundliner	http://www.roundliner.de/
Royaume-Uni	Cedo	http://www.cedo.com/
Benelux	HevelVacuum	http://www.hevel.nl/

France	**Leygatech**	http://www.leygatech.com/
Allemagne	Petroplast	http://www.petroplast.de/
Royaume-Uni	PMCPolythene	http://www.pmcpolythene.co.uk/
Benelux	Beyers	http://www.beyersplastics.be/fr/

Leurs Offres

Machines neuves

SMD continu à faire évoluer nos machines et proposons chaque année des améliorations et nouveautés. La gamme de machines a été rationalisée en poussant le plus possible la standardisation tout en augmentant la souplesse d'utilisation et l'étendue des possibilités de fabrication proposée à nos clients.

Reconstruction de machines

Certains clients confient à SMD leurs machines pour des modifications, des améliorations ou des reconstructions complètes. SMD achète également des machines d'occasion que nous reconstruisons et proposons sur le marché.

Service après-vente et pièces détachées

SMD tient en stock plusieurs centaines de références de pièces détachées pour nos clients et interviens en France et en Europe pour la maintenance et le dépannage de machines. Les pièces mécaniques ou de tôlerie que nous utilisons sont fabriquées par des sous-traitants régionaux, garantissant des fabricants les plus réputés.

Formations

SMD forme à ce jour une équipe de cinq spécialistes avec chacun de nombreuse années d'expérience dans notre domaine.

Bureau d'études mécanique et électrique
Équipé avec logiciels SolidWorks 2015 Premium + SolidWorks Electrical 3D.

Atelier de montage et d'assemblage
Assemblage des machines neuves, démontage et remontage des machines en reconstruction. Préparation de sous-ensembles pour le service après-vente. Câblage d'armoires électriques.

Atelier de fabrication
Mécanique, petite chaudronnerie, peinture. Fabrication de pièces à l'unité et réparation de carters des machines en reconstruction. Équipé avec fraiseuse et tour conventionnels, perceuses sur colonne, postes de soudure, compresseur. Gestion des commandes, expéditions et achats par logiciel spécifiquement développé pour notre activité.

Base de données de l'entreprise

L'entreprise SMD possède un système de production interne . La base de données est gérée par PgSql où est défini une Nomenclature . Une nomenclature est un tableau contenant la désignation de toutes les pièces qui composent un assemblage ou dessin industriel. Cette nomenclature est visualisée par l'interface utilisateur qui est une interface graphique (fichier HTML).

Cette interface utilisateur permet de visualiser :

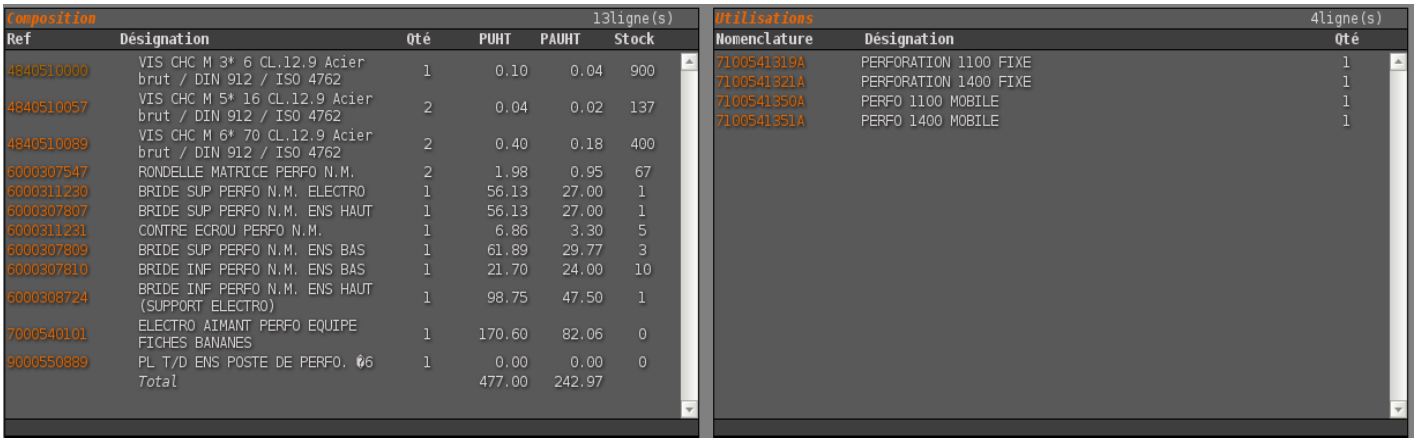
→ La composition de la nomenclature

→ la présence de la nomenclature sélectionner dans une machine

→ le nombre de nomenclatures disponibles

→Les fournisseurs, les achats , les ventes

La composition et l'utilisation de la nomenclature :



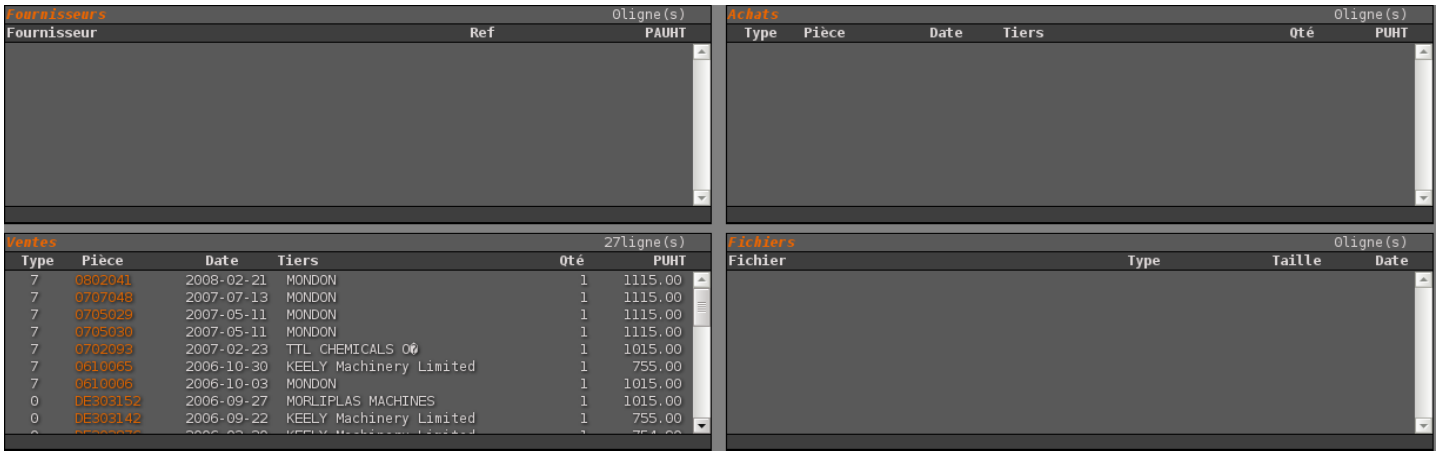
Composition — 13ligne(s)

Ref	Désignation	Qté	PUHT	PAUHT	Stock
4840510000	VIS CHC M 3* 6 CL.12.9 Acier brut / DIN 912 / ISO 4762	1	0.10	0.04	900
4840510057	VIS CHC M 5* 16 CL.12.9 Acier brut / DIN 912 / ISO 4762	2	0.04	0.02	137
4840510089	VIS CHC M 6* 70 CL.12.9 Acier brut / DIN 912 / ISO 4762	2	0.40	0.18	400
6000307547	RONDELLE MATRICE PERFO N.M.	2	1.98	0.95	67
6000311230	BRIDE SUP PERFO N.M. ELECTRO	1	56.13	27.00	1
6000307807	BRIDE SUP PERFO N.M. ENS HAUT	1	56.13	27.00	1
6000311231	CONTRE ECROU PERFO N.M.	1	6.86	3.30	5
6000307809	BRIDE SUP PERFO N.M. ENS BAS	1	61.89	29.77	3
6000307810	BRIDE INF PERFO N.M. ENS BAS	1	21.70	24.00	10
6000308724	BRIDE INF PERFO N.M. ENS HAUT (SUPPORT ELECTRO)	1	98.75	47.50	1
7000540101	ELECTRO AIMANT PERFO EQUIPE FICHES BANANES	1	170.60	82.06	0
9000550889	PL T/D ENS POSTE DE PERFO. Ø6	1	0.00	0.00	0
	Total		477.00	242.97	

Utilisations — 4ligne(s)

Nomenclature	Désignation	Qté
71005413190A	PERFORATION 1100 FIXE	1
71005413210A	PERFORATION 1400 FIXE	1
71005413500A	PERFO 1100 MOBILE	1
71005413510A	PERFO 1400 MOBILE	1

Le côté commercial de la nomenclature :



Structure :

Les pièces dans la base de données sont caractérisées par leur référence :

De 4000 000 000 à 4999 999 999 → Pièces acheter dans le commerce (vis, écrous, etc…)

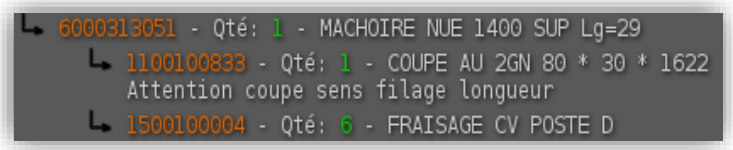
De 6000 000 000 à 6999 999 999 → Composant fabriqué

De 7000 000 000 à 7999 999 999 → Nomenclatures

Vue arbre /Vue à plat :

Vue arbre :

La vue des composants, en arbre permet de visualiser un composant qui est dans la pièce ainsi que ses sous parties. Par exemple Le **Fraisage cv** est présent que six **fois,** pour là sous partie de la mâchoire.



Vue à plat :

La vue des composants, à plat permet de visualiser le nombre de pièce au total présent sur la nomenclature. Par exemple Le Fraisage cv est présent au total **7,06 fois.**

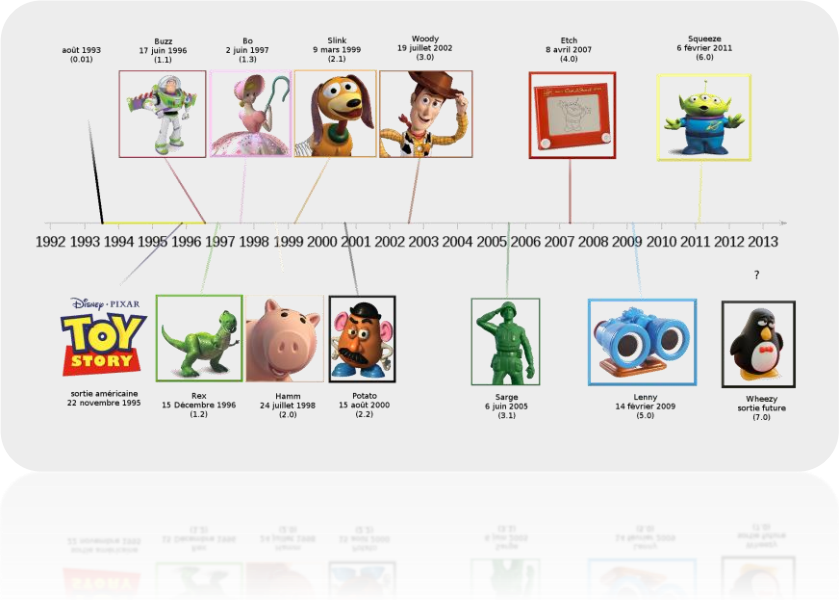
Projet

Mon projet de stage a été de réaliser une amélioration à une base de données existante et de créer une interface entre l'utilisateur et la base de données pour que celui puisse modifier/supprimer/ajouter ou imprimer un article. Pour ce faire, j'ai dû travailler sur un environnement que je connaissais peu (Debian) et surtout un langage de programmation Perl5.

Environnement Debian

Debian est une distribution Linux créer en 1996 par **Ian Mudock** . Il a travaillé chez Pixar auparavant c'est pour cela que Debian possède une particularité celle d'avoir le même noms de versions que les personnages de Toy Story (Hamm,slink,Poatato,Woody…Jessie).Il est mort dans des circonstances douteuse en 2016.

Chez Debian,il existe plusieurs « stades » dans la création de la version de Debian.



Version stable : Une distribution dite « stable »contient la dernière distribution officiellement sortie de Debian (version actuel 8 « Jessie ») 8.4 depuis le 2 Avril 2016. Cette version passe ensuite en oldstable

Version testing : Une distribution dite « testing » est une distribution qui n'a pas encore été accepté par la version stable, cependant cette version apporte la dernière version de logiciel

Version unstable : Une distribution en développement pour les dévellopeurs (actuellement Sid)

Particularité

La distribution Debian possède de nombreuses normes tel que « apt-get install » qui permet d'installer des applicatifs ou logiciels dans une distribution Debian.

Quelques logiciels

XChat est un **client IRC (Internet Relay Chat)** .**L'IRC** est avant tout un protocole de communication textuelle sur Internet . Il sert à la communication instantanée principalement sous la forme de discussions en groupe par l'intermédiaire de canaux de discussion, mais peut aussi être utilisé pour de la communication d'un à un et même échanger des fichiers. Plusieurs canaux sont accessibles via la commande #. Très utilisé pour poser des questions informatiques telles que les langages de programmation mais, aussi sur les environnement d'UNIX.

Emacs est une famille d'éditeurs de texte disposant d'un ensemble extensible de fonctionnalités et qui est très populaire parmi les programmeurs et plus généralement les personnes ayant des compétences techniques sur les ordinateurs. Il distingue la syntaxe de nombreux langages (HTML, PERL, Python …)

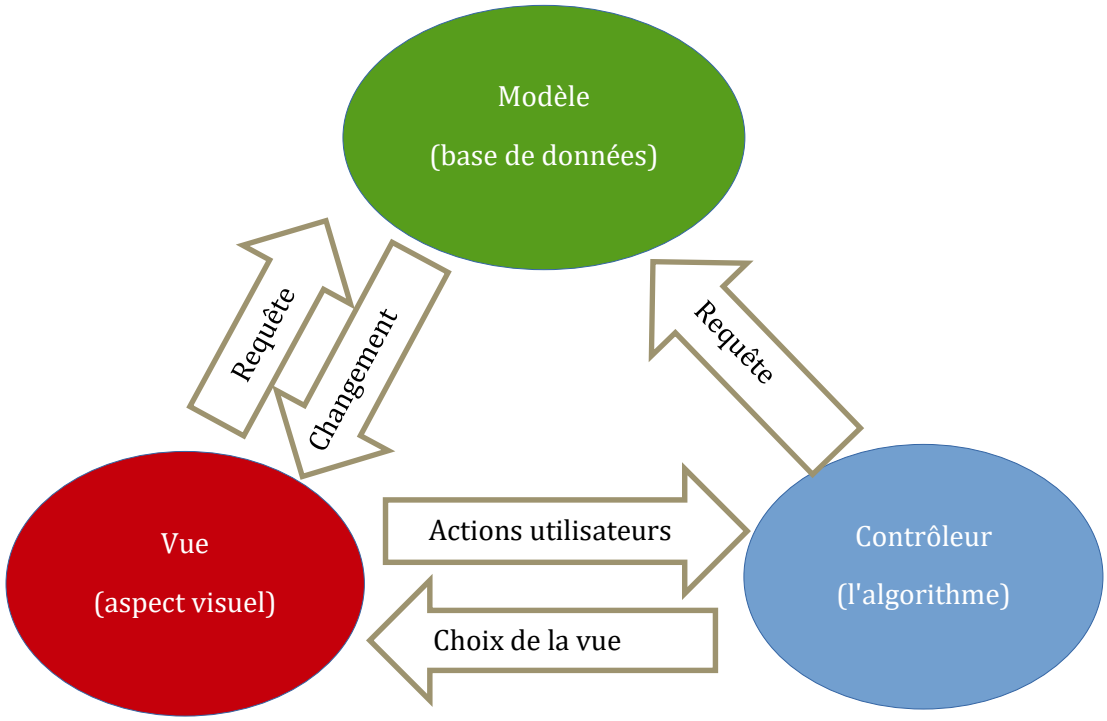
L'EMACS originel, signifiant ***E**diting **MAC**roS*** qui a été écrit en 1976 par Richard Stallman,

Iceweasel est un des navigateurs Web du projet Debian. C'est l'adaptation du célèbre navigateur **Firefox** de la fondation Mozilla au projet Debian. Il n'y a quasiment aucune différence de code entre **Iceweasel** et **Firefox. Il est généralement intégré dans une distribution Debian.**

Icedove est un logiciel de messagerie électronique du projet Debian. Il prend en charge différents comptes de messagerie (POP, IMAP, Gmail) et possède de nombreuses fonctionnalités tel que

- Dispose d'un filtre anti-spam.
- Permet la lecture des flux RSS.
- Offre une organisation facile de mails par des marqueurs et dossiers virtuels.
- Gère l'affichage des messages et des dossiers par onglets etc.

Modèle MVC:



Modèle : La couche Modèle représente la partie de l'application qui exécute la logique métier. Cela signifie qu'elle est responsable de récupérer les données.

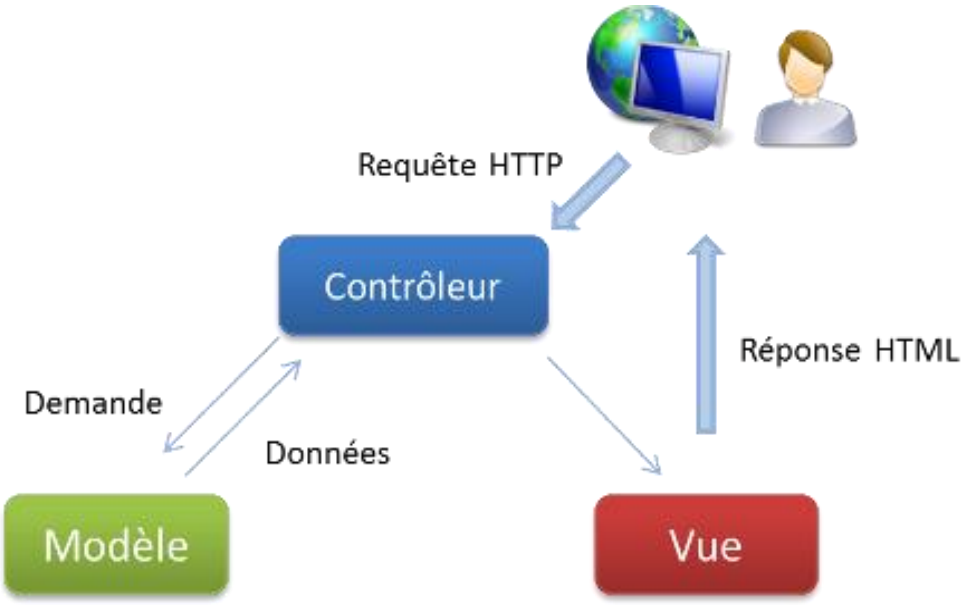
Vue : La Vue retourne une présentation des données venant du modèle. L'aspect visuel de l'application.

Contrôleur : La couche Controller gère les requêtes des utilisateurs.

Modèle Cette couche va permettre de stocker l'intégralité de la base de données de l'entreprise

Vue Cette couche permettra à l'utilisateur d'avoir une réponse html.

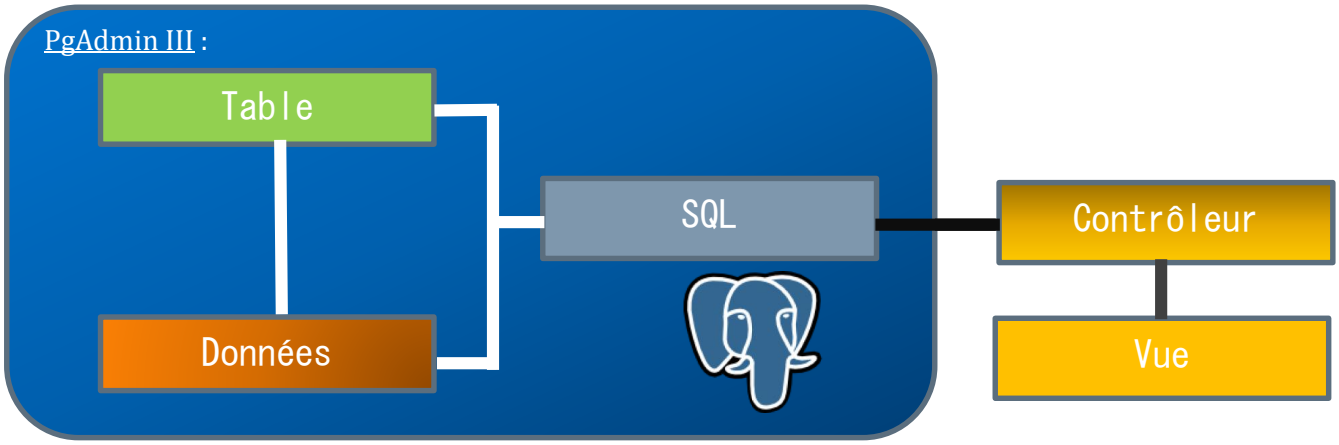
Contrôleur : cette couche permettra via l'action de l'utilisateur ajouter/supprimer une donnée dans la base de donnée



PgAdmin III

PostgreSQL est un système de base de données relationnelle et également outil libre. Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme **MySQL**), ou propriétaires (comme **Oracle**). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises. Il est souvent plus utilisé en entreprise que MySQL, de par ses nombreux avantages, il permet par exemple de visualiser l'intégralité d'une table sans requête SQL.

Voici un schéma du rôle de PgAdmin III au sein du projet :



PgAdmin III possède une arborescence assez détaillée, dans un premier temps il faut sélectionner la base de données en question ici la base de données s'appelle **testdb**.

La rubrique Schémas contient la partie Public.

Ce qui nous intéresse particulièrement est la partie publique. Cette partie contient l'ensemble des tables de la base de données pour **testdb**.

La rubrique Séquences regroupe l'ensemble des clées primaires et étrangères de toute la base de donnée (voir plus tard pour l'explication)

Ajouter des données

Une table se créer facilement sur PgAdmin III. En fessant un clic droit sur la rubrique Tables et choisir "Ajouter une table' 'Il peut y avoir plusieurs paramètres à mettre en place comme celle d'ajouter des données dans la rubrique "colonne".

Pour le projet deux tables ont été créé :

Table Nomenclature :

Table article :

Créer une clé primaire /étrangère

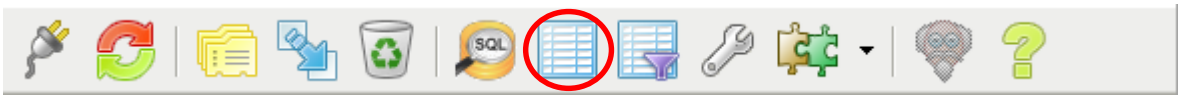
Une particularité de PgAdmin III est qu'il ne possède pas d'options "clée primaire' 'Pour créer une clée primaire ou étrangère il nous faut créer une Séquence (même principe que pour créer une table). Elle possède également de nombreux paramètres comme l'auto incrémentation et devra être assimilée à une donnée de la table et prend comme valeur par défaut :

Nextval('"nom_de_la_séquence"'::regclass)

Une clée primaire ou étrangère est réglé par la catégorie "Contraintes' 'Lors de la création d'une nouvelle contrainte nous avons plusieurs choix (clée primaire, clée étrangère etc.).
Une clée qu'elle soit primaire ou étrangère doit faire référence à une colonne.

Avantage PgAdmin III

PgAdmin III possède de nombreux avantages comme la visualisation d'intégralité d'une table sans requête.



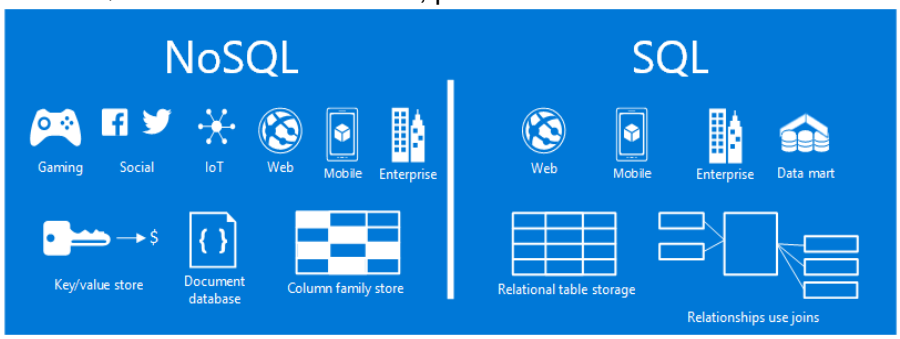
Exemple d'une table :

	id [PK] serial	date date	type integer	delai integer	id_tiers integer	id_agent integer	description character varying	ar_ref character varying(11)	date_livraison date
1	1	2011-06-01	2	90	904	2774	Machine E-1400 CANTASAC	7999212001	2015-12-31
2	2	2011-11-16	2	90	2775		Machine D-1400 COTIPLAST	7999212002	2015-12-31
3	3	2012-07-09	2	90	1150		PA 650	7999212003	2015-12-31
4	4	2012-07-10	1	90	1150		PA 650 STOCK	7999212004	2015-12-31
5	5	2012-11-02	2	20	2775		Banc soudure soufflets	7999212005	2015-12-31
6	6	2013-03-01	2	60	2775		Machine 882GGL + Perfo double et dérouleur 5 postes	7999212006	2015-12-31
7	7	2014-07-03	2	90	1212	1812	Machine C-1400 SCHMEDES	7999212007	2015-12-31
8	8	2014-12-09	2	90	987	2774	Machine E-0800 CHAPTEUIL	7999212008	2015-12-31
9	9	2015-06-18	2	90	1017	2774	Machine E-1400 REP	7999212009	
10	10	2016-03-25	2	120	1070	2774	Machine E-1100 SÜDPACK	7999212010	2016-09-15
*									

SQL / No SQL

No SQL signifie "Not Only SQL", littéralement "pas seulement SQL". Ce terme désigne l'ensemble des bases de données qui s'opposent à la notion relationnelle des SGBDR. La définition, "pas seulement SQL", En effet, No SQL ne vient pas remplacer les bases de données relationnelles mais proposer une alternative ou compléter les fonctionnalités des outils de base de données.

Les géants du Web comme Google et **Amazon** ont vu leurs besoins en termes de charge de données croître depuis plusieurs années. Et c'est pour répondre à ces besoins que le No **SQL** a vus le jour. Par la suite des entreprises comme **Facebook, Twitter** ou encore **LinkedIn** ont migré une partie de leurs données sur des bases No SQL. Les bases **clef/valeur**, permettent de stocker des informations sous forme d'un couple clef/valeur où la valeur peut être une chaîne de caractère, un entier ou un objet **sérialisé**.



Nos deux tables sont alors créées , nous allons mettre en place plusieurs modèles de base de données afin de bien comprendre leurs fonctionnements.

Modèle sans relation :

Modèle Merise :

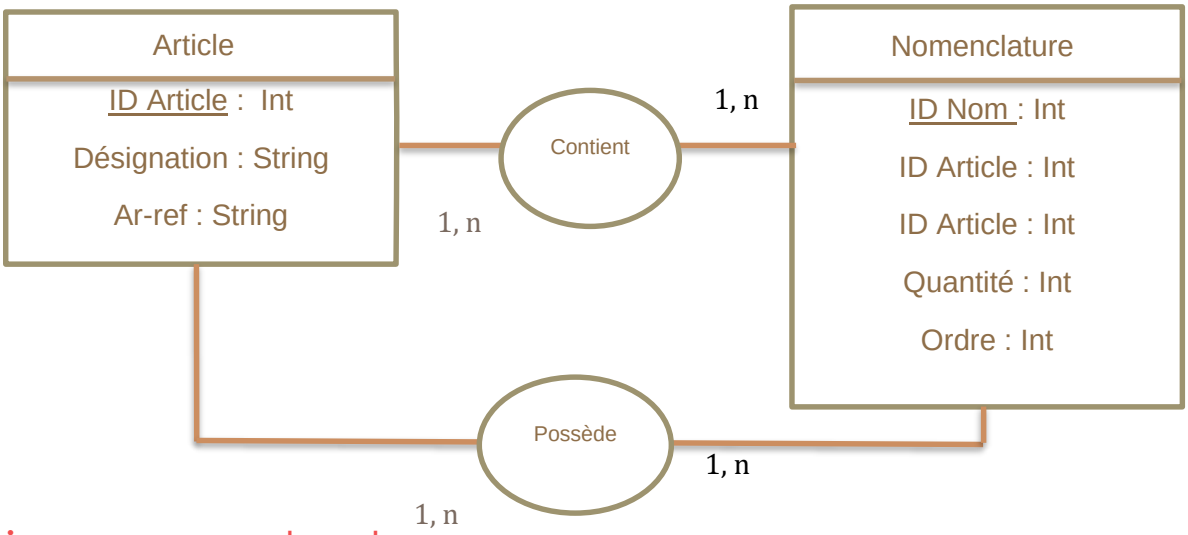
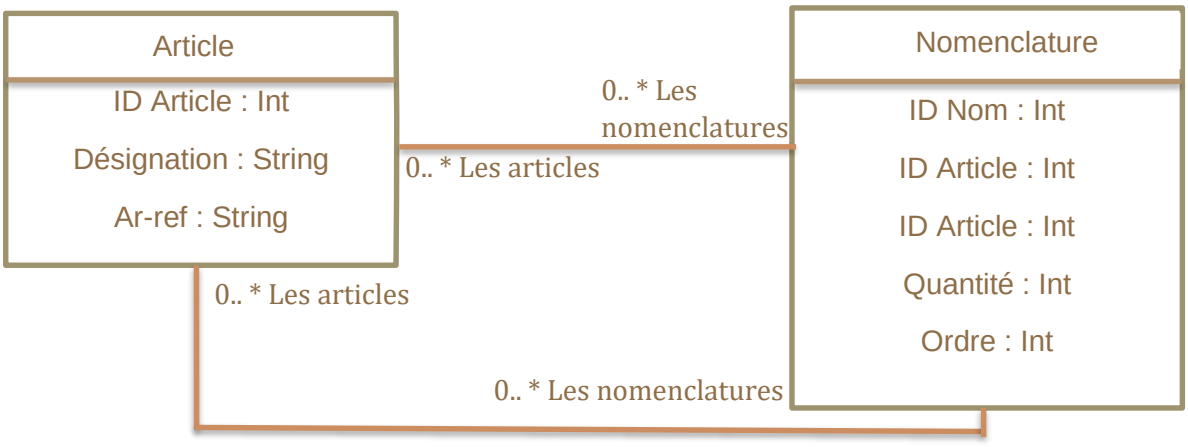


Diagramme de classes :



Présentation Perl

Le Langage Perl est un langage de programmation créé par **Larry Wall** en 1987 pour traiter facilement de l'information de type textuel.

Il prend en charge les expressions régulières dans sa syntaxe même, permettant ainsi directement des actions sur l'aspect général de séquences de texte.

Perl 5 particulièrement bien adapté aux tâches faisant intervenir, les utilitaires systèmes, les outils logiciels, les gestionnaires de tâches, l'accès aux bases de données, la programmation graphique, les réseaux et la programmation web. **Ces points forts** en font un langage particulièrement populaire auprès des administrateurs systèmes mais l'utilisent également des mathématiciens, des généticiens, des journalistes et même des managers.

Le statut du langage est celui de logiciel libre, distribué sous double License : **Artistic License** et **GPL.**

HTTP://WWW.CPAN.ORG/

CPAN est un sigle pour ***Comprehensive Perl Archive Network*** **(réseau complet d'archive Perl)**. Il s'agit d'une archive dense de logiciels de bibliothèques de fonctions utilitaires écrits en langage Perl. Grâce à sa popularité le langage Perl a vu sa communauté évoluée au fil du temps**, il n'est donc pas rare qu'une personne a** déjà réalisé un module sur ce qu'on est en train de faire.

Perl a su influencé pas mal de langages de programmation tel-que Python et PHP (qui était un ancien module de Perl) il n'est donc pas rare que sa syntaxe soit quelques fois la même. Une rumeur dit que PHP ne serait pas ***Hypertext Preprocessor*** *mais* **People Hate Perl**.

Le langage Perl est en constante évolution ce qui a permis de créer **Perl6** Il s'agit aussi d'une refonte profonde du langage, aussi bien dans sa conception que dans son implémentation il possède un meilleur support de la programmation orientée objet.

Mieux connaitre Perl avec Moose

Moose est une extension de Perl5 permettant d'aider les développeurs à mieux programmer la programmation orienté objet. Il apporte de nombreux outils, qui permettent comme en **C# de créer** des classes, des outils héritages etc…

Pour installer le module Moose plusieurs choix sont proposer :

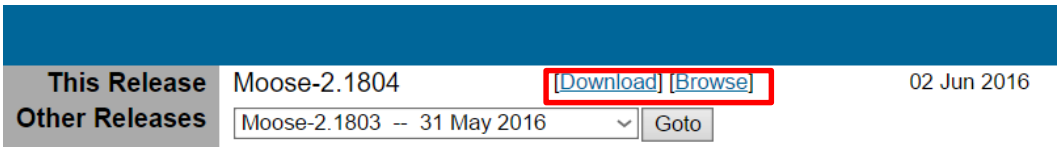
a) Télécharger grâce à **CPAN**

perl -MCPAN -e " Install Moose"

Attention cette commande fonctionne uniquement si au préalable installer le module MCPAN. (Apt-get install cpanminus)

b) En **tarball**

Il faut télécharger depuis un navigateur (http://search.cpan.org/dist/Moose)



Une fois ceci télécharger, il faut l'installer avec les commandes suivantes :

```
tar xzvvf Moose
cd Moose
perl Makefile.pl
sudo make install
```

Tar est une commande qui concatène plusieurs fichiers pour les regrouper en une seul archive (la commande Tar était anciennement utilisé pour les bande magnétique)

c)Téléchargement **direct**

Comme expliquer auparavant Debian possède

apt-get install libmoose-perl

Il faut être administrateur pour faire cette commande.

Utilisation

Pour utiliser Moose, il faut créer deux fichiers qui sont dans le même répertoire : lib/test.pm et bin/test.pl (les deux répertoires sont déjà créés dans Dancer2).

Humain.pl :

```
# debut use #
 use strict;
 use warnings;
 use v5.10;
# fin use #

 use humain; # classe Humain

my $vincent = humain->new(nom=>'vincent')
my $etudiant = humain->new(nom=>'Martin')
my $bidule = humain->new(nom => 'jean' );
```

Le programme Humain.pl définit une classe humaine à l'aide de la commande Use.
Puis nous définissons plusieurs humains.
Il nous reste plus qu'à afficher les variables à l'aide d'un print ou Say.

Humain.pm :

```
package humain;
use Moose; # on utilise Moose ( qui intégre strict et warning)

extends 'Personne'; #héritage du fichier personne

has 'nom' => (is => 'rw');
has 'age' => (isa => 'Int', is => 'rw');
has 'lieux'=> (is => 'rw');
1;
```

Le programme prend le package humain définit sur Humain.pl et utilise Moose.
Il peut également faire un héritage, là il hérite de ‘’use **Personne”.**
Il définit les variables que le programmes Humain.pl a besoin.
Moose doit toujours retourner une valeur vraie donc il y a un 1 à la fin.

Commande dos :

```
perl -Ilib bin/humain.pl
```

Cette commande va permettre de lancer le programme et de restituer l'intégralité des print ou say écrit dans Humain.pl.

Fichier CSV

Un **fichier CSV** (***Comma-separated values***) est un fichier tableur, contenant des données sur chaque ligne séparée par un caractère de séparation (généralement une virgule, un point-virgule ou une tabulation). Il peut être lu par beaucoup de logiciels de tableur mais aussi bloc-notes et autres. Le but de cette partie et de combiner deux fichiers CSV en additionnant une rubrique si celle-ci est présente dans les deux fichiers.

Pour demander à l'utilisateur de tester son fichier il faut faire ceci :

My @files = @ARGV ; et faire une boucle : foreach my $file (@files) pour tester sur tous les fichiers

Tester l'existence d'un fichier et l'ouvrir

```
#voir si le fichier existe
my $test= (-e $file && -f $file) or die "Fichier <$file> inexistant\n";

#ouverture du fichier
open(my $FICHIER, '<', $file) or die "Fichier <$file>  impossible à lire\n";
```

Lire ligne par ligne et assimilé des variables

```
#boucle ligne par ligne
while ( my $ligne = <$FICHIER> ) {

#définir des variables qui prennent les lignes du fichier
my( $PST, $ref, $design, $stock, $qte, $udv, ) = split ';', $ligne;
```

Additionner la variable Qté et table de hashage

```
#la variable itm prend la valeur de quantité si elle éxiste
$itm_qte = $h{$ref}{'Qté'} if (exists $h{$ref});

#table de hashage qui assimile a la variable le mot définit ''
my $result = { 'designation' => $design,
               'Stock'       => $stock,
               'Qté'         => $itm_qte + $qte,
               'Udv'         => $udv, };
```

Fermer le fichier et afficher le contenu

1

```
    #met la variable $result dans la variable h
    $h{$ref}=$result;

    }
    #ferme le fichier
    close $FICHIER;

affiche le tableaux de hashage h
rint Dumper %h ;
```

Framework Dancer2

Un Framework est un ensemble d'outils informatique qui constituent un logiciel ou application informatique (également pour aider et facilité la vie des programmeurs). Dancer est un **Framework** pour application web pour Perl.

Pour installer Dancer2 il y a plusieurs moyens :

- Apt-get install libdancer2 –Perl
- perl – MCPAN –e 'install Dancer2'
- curl -L http://cpanmin.us | perl - --su Dancer2

Afin que l'application web soit créer il faut au préalable l'installer dans un fichier :

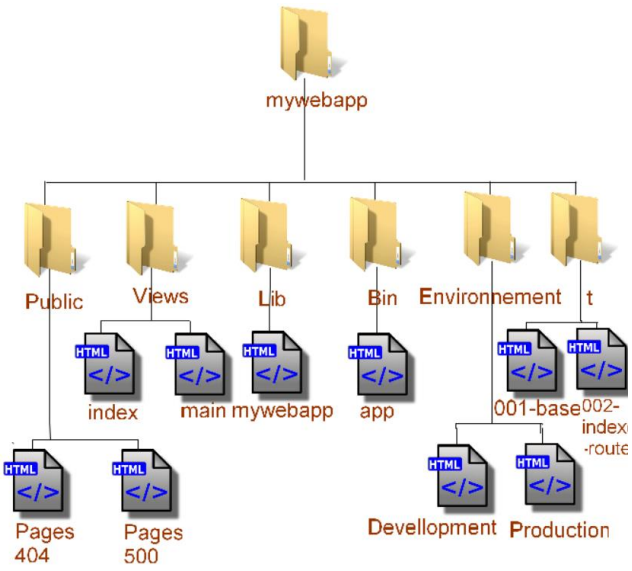
- Dancer2 –a mywebapp && cd mywebapp

Le Framework Dancer2 est un **PSGI** (Perl Web Server Gateway Interface). Le psgi représente une interface entre un serveur web et une application écrite en Perl. Pour que tout cela fonctionne correctement nous devons indiquer le **plack** qui est une implémentation de référence de PSGI

- Plackup –p 5000 bin/app.psgi

Cette commande doit se faire à chaque modification de fichier de plus il faut souvent vider le cache du navigateur. Après ceci fait nous pouvons observer notre application web en local :

http://localhost:5000 ou 0.0.0.0 :5000



Le dossier mywebapp comporte de nombreux sous dossiers qui séparent :

Le visuel :

- ➔ La vue (partie HTML et Template)
- ➔ La partie css et images

L'application :

- ➔ Les programmes
- ➔ Les routes

Configuration :

- ➔ La configuration de Dancer
- ➔ Le Lancement de Dancer

Dancer2

Template

Un template représente le squelette de la page HTML, il peut être utile pour distribuer le même CSS aux différentes pages du programme. Un Template est déjà installé de base lors de l'installation de **dancer2.** Il se trouve dans **Views/Layouts/main.tt.**

Le template prend son **CSS et** les images dans la rubriques **public/css** et **public/images** et peut bien évidement être modifiable.

Exemple :

```
<!DOCTYPE html>
</ul>
<html lang="en">
<head>
     <script type = "text/javascript"
         src = "http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

  <meta charset="<% settings.charset %>">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes">
  <title>mywebapp</title>
<link rel="stylesheet" type="text/css" href="css/style.css" />
```

Template Toolkit

Template Toolkit est un module qui permet de transférer une variable d'un programme Perl vers un template.
Pour installer le module il faut faire cette commande :

apt-get install libtemplate-perl
apt-get install libtemplate-plugin-dbi-perl

Puis modifier le fichier **config.yml** pour que template toolkit soit bien configuré

```
template: "template_toolkit"

#engines:
 #  template:
  #    template_toolkit:
```

Exemple :

```
[% BLOCK tabrow %]
   <tr><td>[% name %]</td><td>[% email %]</td></tr>
[% END %]

<table>
[% PROCESS tabrow name="tom"   email="tom@here.org"    %]
[% PROCESS tabrow name="dick"  email="disk@there.org"  %]
[% PROCESS tabrow name="larry" email="larry@where.org" %]
</table>
```

tom	tom@here.org
dick	disk@there.org
larry	larry@where.org

Route par défaut

```
package mywebapp;
use Dancer2;

our $VERSION = '0.1';

get '/' => sub {

    template 'index';
};

true;
```

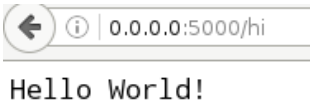
Le système de route de Dancer2 se trouve par défaut dans le fichier **bin**
Une route dans dancer2 permet d'envoyer un programme ou un template sous une réponse HTTP bien spécifique. La route par défaut est **'/'** donc 0.0.0.0**:5000** va envoyer à l'utilisateur le template nommé index (se trouvant dans le views).

Une route doit toujours se terminer par une valeur vrais (donc true) pour fonctionner convenablement.

Route

```
get '/hi' => sub {
    return 'Hello World!';
};
```

Une route est un chemin que dancer2 va prendre quand il sera lancer ici la route est **/hi.** Pour lancer cette route il faut taper **0.0.0.0:5000/hi** et retourne le texte **'Hello World !'.**



Envoyer une variable à un Template

```
get '/' => sub {
    print "*******HELLO*******\n" ;
 #Gerere le template

    my $tt = Template->new()
        || die Template->error(), "\n";

    my @heroes = ( { name => 'Vincent',
                     age => 19, },
                   { name => 'Woody',
                     age => 29, }, );

    my %data = ( var1 => 'sa marche',
                 var2 => '123',
                 heroes => \@heroes, );
    my $page;
    $tt->process( 'views/layouts/mywebapp.tt', \%data,\$page) || die $tt->error(), "\n";
    return $page;

};
```

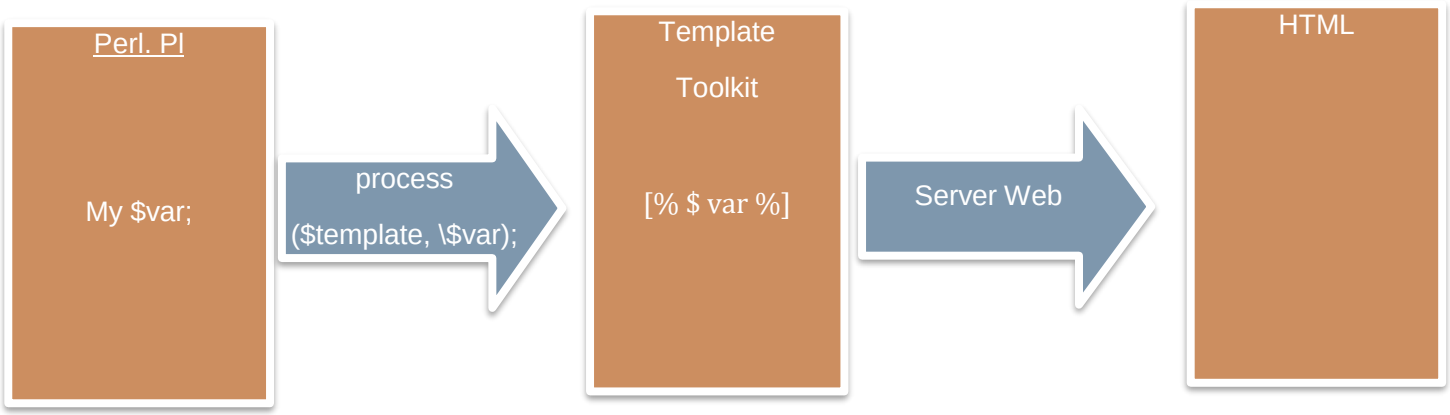
Dans un premier temps, nous allons créer un nouveau template et définir s'il y a une erreur ou non.

Puis on définit des variables de types différentes (chaine de caractère,tableaux etc…).

Puis on envoit dans le template les variables définit plus haut. On retourne la variable $page car , par défaut le code envoit le code sur le terminal et non sur la page.

Il faut maintenant faire [% var1 %] sur le template. Nous verront le contenu de la variable $var1

Résumé :

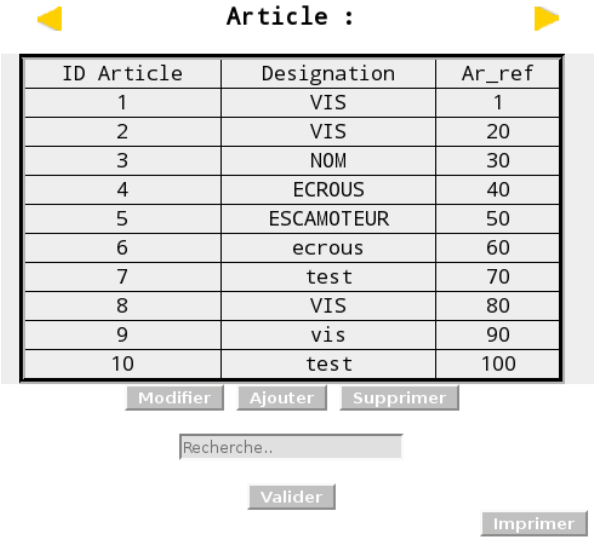


Se connecter à une base de données

Pour se connecter a une basse de donnée sous le Framework Dancer2 .Il faut installer le plugin DBI (déjà installer auparavant) puis se connecter à sa base de donnée comme ceci :

```
[% USE DBI( database = 'dbi:Pg:dbname=testdb',
    username = 'vince',
    password = 'vince' )%]
```

Pour notre tableaux nous avons fait une requête SQL suivante :



[% FOREACH article IN DBI.query('SELECT * FROM articleLIMIT 10') %]

puis :

[% article.ID_Article %]

[% article.Designation %]

[% article.Ar_ref %]

AJAX dans Dancer2

Pour installer AJAX dans dancer2 il faut créer un nouveau projet avec la commande :

Dancer2 -a D2 :: Ajax

cd D2-Ajax

git init

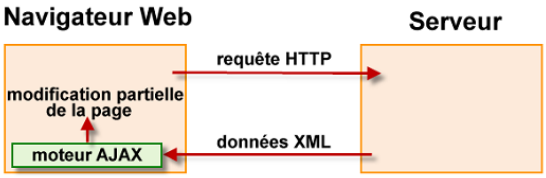
git add .

git commit –m '"initial"

AJAX

Ajax (acronyme d'***Asynchronous JavaScript and XML***) permet de construire des applications web et des sites dynamiques interactifs.Ajax combine JavaScript, les CSS, JSON, XML, le DOM ,le JQUERY DOM et JavaScript permettent de modifier l'information présentée dans le navigateur en respectant sa structure ;

J'ais voulus abordé AJAX car il fallais que l'utilisateur puisse accéder a l'ensemble de sa basse de données sousune forme de pagination (flèche jaune).

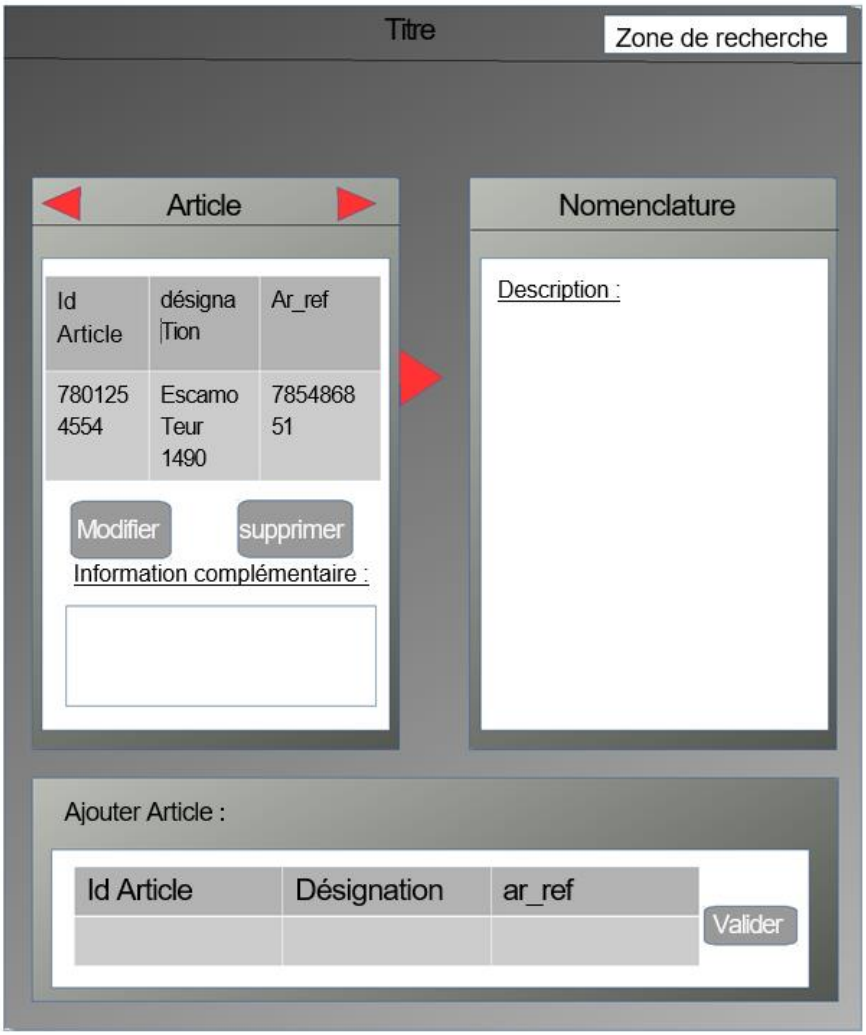


Cependant par manque de temps , je n'est pas pu apprendre le langage AJAX pour le projet c'est pour cela qu'aucune interaction avec l'utilisateur ne fonctionne convenablement .

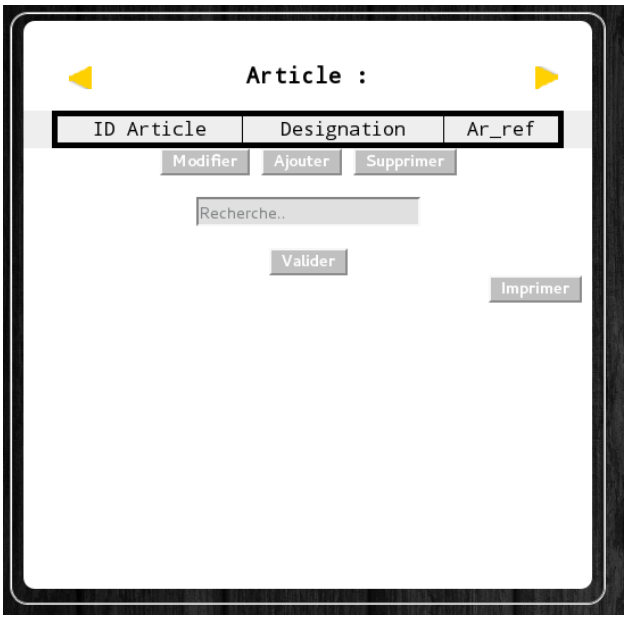
Vue

Le projet était de réaliser une interface entre l'utilisateur est la base de donnée.L'utilisateur pourra accéder aux informations d'un article.Il pourra également Modifier /Ajouter/Supprimer et même Imprimer un Article.

Pour cela nous avons fait un croquit de l'interface :



Puis le résultat final :



L'interface entre l'utilisateur est le programme est très important il permet à l'utilisateur qui ne connait rien au programme depouvoir quand même interagir avec celui-ci. L'interface a était murement réfléchit pour être a la fois simple et clir pour l'utilisateur.

Conclusion :

Au cours de mon stage de cinq semaines j'ai pu apprendre un langage de programmation le perl. Le monde professionnel est très différent du monde scolaire, les attentes ne sont pas les mêmes. Il est très important surtout en programmation d'être autonome et de mettre les idées au clair, comme disait mon maitre de stage, ce qui marche sur papier marche sur ordinateur.