

Introduction

Gestion de stock – facturation : openstock 1.02 – juin 2006

Le rapport de stage de Laurent POUCHOULOU décrivant son travail sur la période d Avril à Juin 2006 a été transformé en documentation pour openStock pour servir

- *de support aux développeurs qui souhaitent utiliser ou compléter openstock – facture.*
- *De documentation en direction des utilisateurs (3- description fonctionnelle)*



OpenStock a pour objectif de répondre aux exigences de la comptabilité publique lqui oblige les collectivités à tenir un suivi de ces stocks.

La version 1,02 propose un module de facturation pour certains services.

Cette gestion de stock est différente selon les services.

Pour le service des musée ou informatique , les articles en stock sont gérés par des régies ou des services et ne font pas l'objet de facturation.

Dans la gestion de stock des consommables du service informatique, un service a besoin d'une cartouche, une personne vient à la DSIT et il est effectué une sortie de l'article pour ce service.

Dans la gestion de stock des musées, un client choisi un article dans le magasin. Il est encaissé par un régisseur qui reverse l ensemble des encaissements à la recette municipale.

Par contre le service des pompes funèbres effectue des sorties d'articles ou de prestations pour des clients. Il a donc fallu adapter un module de facturation afin d'éditer les factures aux clients.

Il y a donc 3 cas d'utilisation d'openstock: qui se paramètre dans dyn/base.php

1 - CAS SERVICE

```
$conn[1]=array('OpenStockService', 'mysql', '', 'root', '', '', 'localhost', '', '', 'openstockservice', 'AAAA-MM-JJ', 'service');
```

2 - CAS REGIE

```
$conn[2]=array('OpenStockRegie', 'mysql', '', 'root', '', '', 'localhost', '', '', 'openstockregie', 'AAAA-MM-JJ', 'regie');
```

3 - CAS FACTURE

```
$conn[3]=array('OpenStockFacture', 'mysql', '', 'root', '', '', 'localhost', '', '', 'openstockfacture', 'AAAA-MM-JJ', 'facture');
```

SOMMAIRE

1- Phase d analyse (uml)

2- la solution envisagée

3- la description fonctionnelle

- 3.1 connexion à l application
- 3.2 menu utilisateur
- 3.3 ajouter un article
- 3.4 ajouter et livrer une commande
- 3.5 effectuer une sortie
 - 3.5.1. dans le cas d'une regie
 - 3.5.2. dans le cas d'un service
 - 3.5.3. dans le cas d'une facture
- 3.6 Editer les états

Annexe

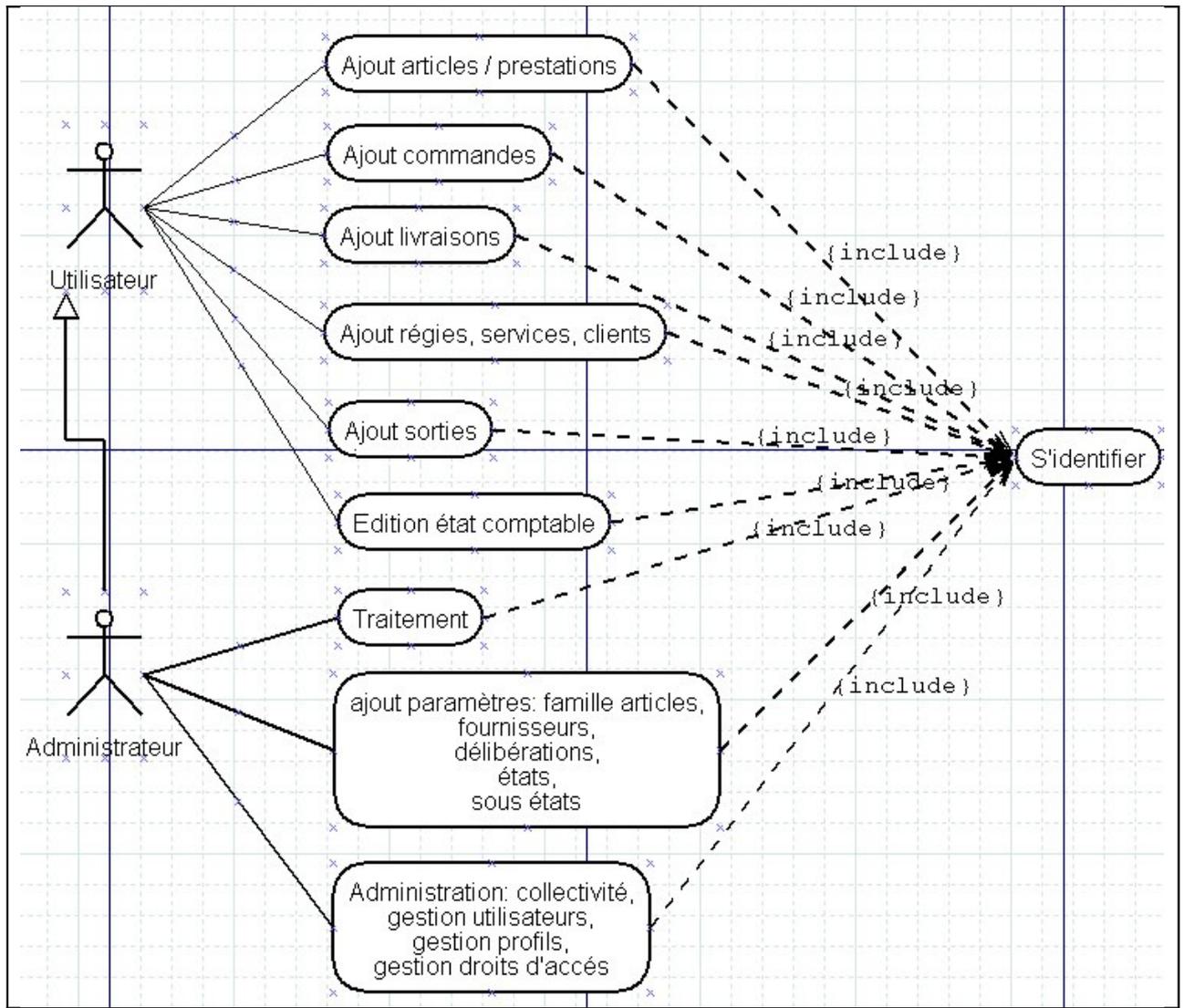
- 1 – Description textuelles des cas d utilisation
- 2 – Gestion des utilisateurs
- 3 – quelques états

1- Analyse

Il est proposé ci desous une analyse UML complétant celle de la version 1,01 d'openStock complétées par les fonctionnalités demandées.

1.1 Cas d'utilisation

1.1.1 Diagramme des cas d'utilisation



Cas d'utilisation

1.1.2 Descriptions textuelles des cas d'utilisation

Voir annexe 1

1.2 Diagramme des classes persistantes

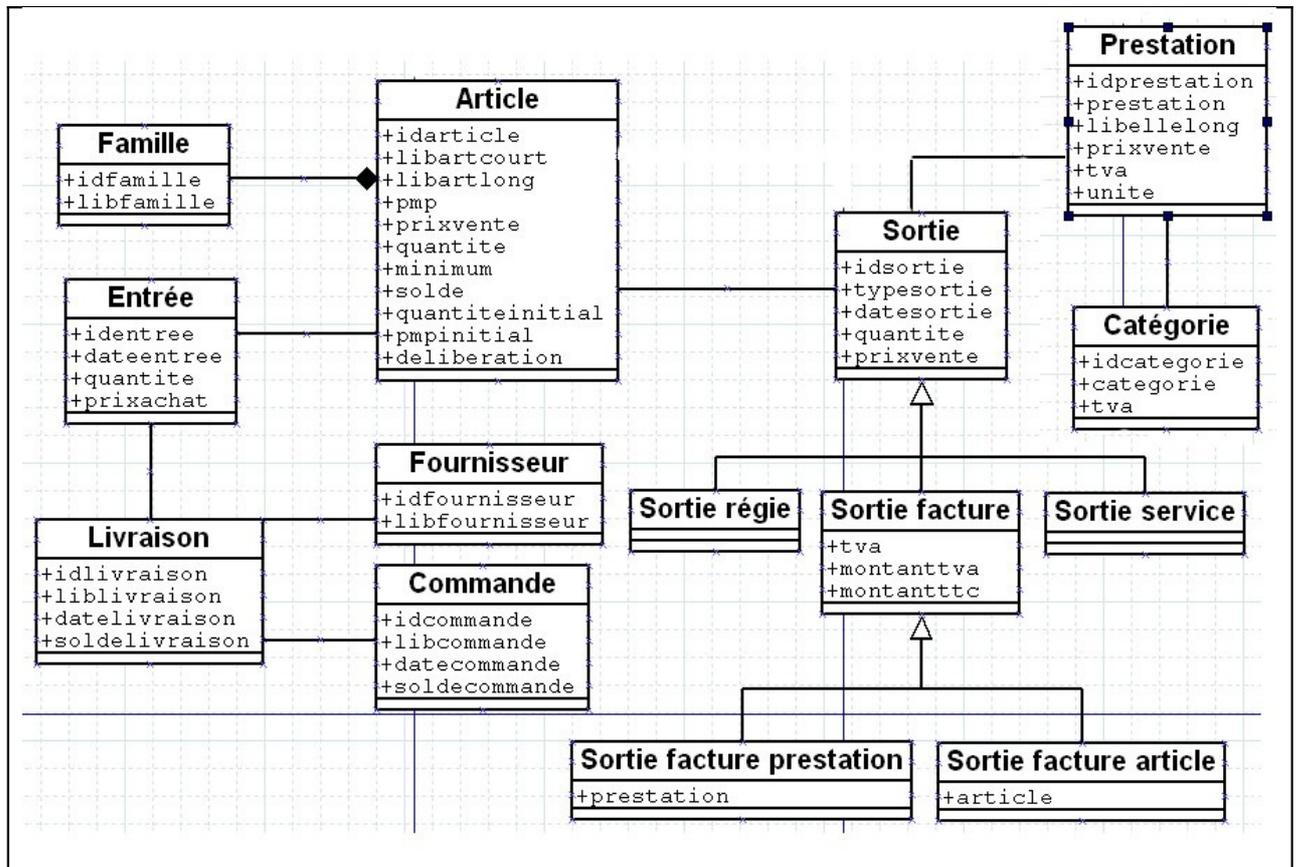


Diagramme des classes persistantes

Remarques :

Les utilisateurs possèdent des profils ainsi que des droits d'accès. Le diagramme qui gère ces classes est mis en annexe 2.

2 - Solution technique

Openstock est développé en langage PHP objet.

L'application se doit d'être adaptée à tout type de Système de Gestion de Base de Donnée (SGBD).

2.1 Serveur web et hébergement

Le logiciel a été testé sous EASY PHP 1,7 et WAMP5.

2.2 Langage utilisé

L'application a été programmée en PHP objet fonctionne sous la version 4 de PHP.

L'application est basée sur le composant : OpenMairie version égale ou supérieure à 1,11.

Ce composant écrit en PHP objet, permet la création d'applications basées sur des formulaires, sous formulaires, d'affichage sous forme tabulaire et l'édition au format PDF du contenu de la base de données. Afin d'adapter les possibilités du composant à une application spécifique, la surcharge de méthodes est nécessaire.

A partir de ces composants, plusieurs applications ont été développées ou sont encore en cours de développement comme par exemple openElec qui gère des listes électorales, openCimetière utilisé comme son nom l'indique dans la gestion des cimetières ainsi que des applications de gestion de parcs informatiques, de courriers, de demandes de logements,...

Les composants OpenMairie et les différents logiciels développés à partir de ceux-ci sont « open source » et sont téléchargeables sur le **GFORGE** de l'**ADULLACT**.

2.3 Système de gestion de base de données

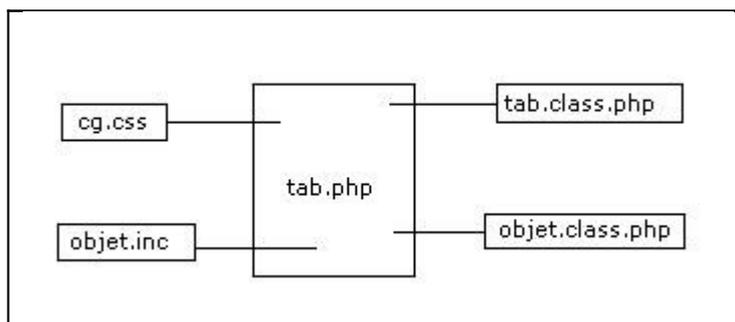
Le SGBD choisit est MySQL en ce qui concerne ce stage, mais la possibilité de changer de base reste possible.

Pour la gestion des différents SGBD pouvant être utilisés avec les applications dérivées d'OpenMairie, l'utilisation du connecteur « open source » DBPear était imposée.

2.4 Structure des pages de l'application

Les données sont affichées dans des pages sous forme de tableau et font appel à plusieurs fichiers :

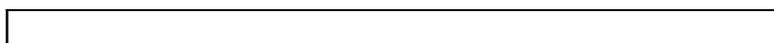
- La page principale est la page tab.php qui inclus les autres fichiers pour l'affichage des données,
- La classe tab.class.php qui est une classe d'OpenMairie qui gère l'affichage du formulaire,
- Le fichier objet.inc qui contient toutes les informations relatives à l'interaction avec la base de données telles que le paramétrage des requêtes, les liens correspondants aux icônes,
- Le fichier objet.class.php qui contient les surcharges des fonctions des classes dbForm.class et formulaire.class d'OpenMairie; c'est ce fichier qui définit les types de champs qui vont apparaître, la possibilité d'insérer ou non des informations,
- Une feuille de style cg.css pour l'apparence de la page.

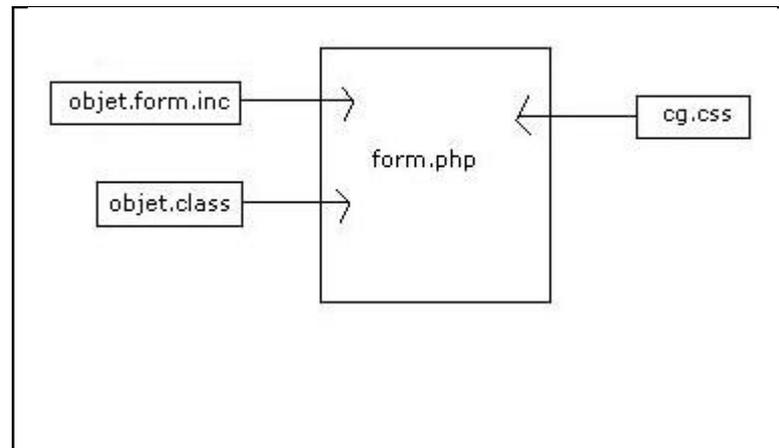


Structure d'un page affichant les données dans un tableau

Lorsqu'on souhaite modifier une entrée, le formulaire appelé est la page form.php qui contient aussi d'autres fichiers :

- La feuille de style cg.css,
- Le fichier objet.class.php,
- Le fichier objet.form.inc qui permet le lien avec la base de données qui donne les champs à modifier ou à ajouter, et qui permet l'affichage d'un sous formulaire.





Description d'un formulaire

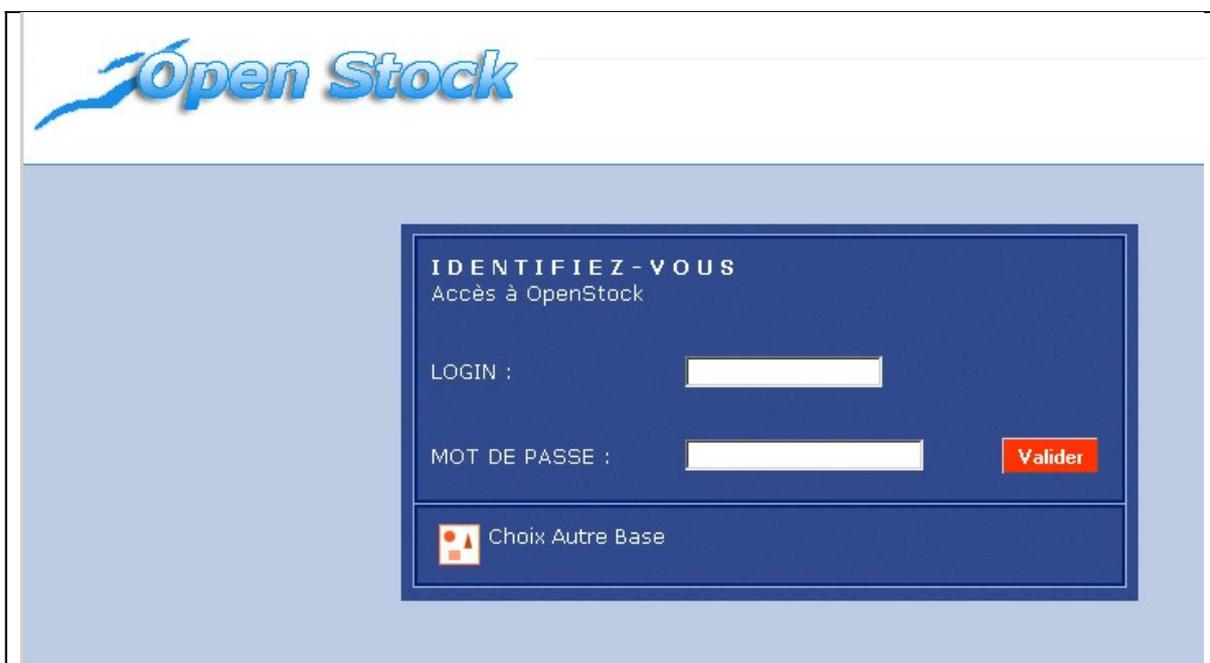
L'affichage d'un sous formulaire est aussi possible sur le même principe que le formulaire avec l'appel du fichier sousform.php à la place du fichier form.php.

3 – Description fonctionnelle

3.1 Connexion à l'application

L'administrateur de l'application donne à chaque utilisateur un code et un login afin qu'il puisse se connecter.

Une fois l'adresse du site saisi dans la barre d'adresse de leur navigateur ou en entrant sur l'intranet, les utilisateurs sont invités à s'identifier.



Login.php

Une fois les informations de connexion saisies, le formulaire renvoie l'utilisateur vers une page de traitement afin de vérifier la validité des informations. En cas d'erreur, l'utilisateur est invité à ressaisir ses identifiants. Sinon, il est redirigé vers un menu lui proposant les fonctionnalités qui lui sont offertes.

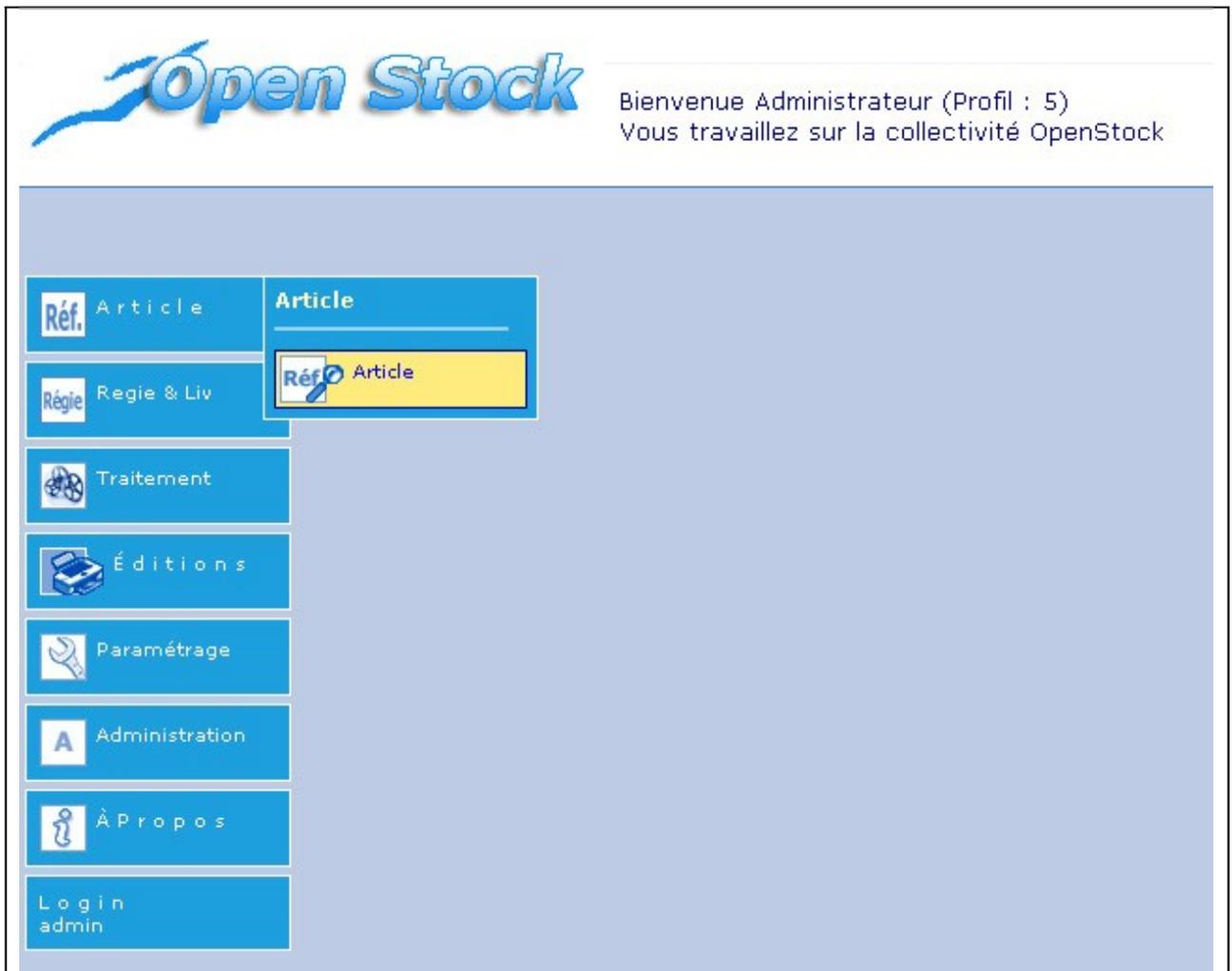
```
if ( $step==1 ){
    $pwd = md5($_POST['pwd']);
    //$pwd = ($_POST['pwd']);
    if($DEBUG==1)
        echo "La base ".$dsn['database']." est connectée.<br>";
    $sql = "SELECT * ",
           "from utilisateur ",
           "WHERE login like '".$_POST['login']."' ",
           "AND pwd like '".$pwd."' ";
    $res = $db->query($sql);
    if (DB :: isError($res))
        die($res->getMessage());
    else{
        while ($row=& $res->fetchRow()){
            $profil = $row[4];
            $nom = $row[1];
            $login=$row[2];
        }
        if ( $profil==0 ) {
            $error= "Login ou Mot de passe incorrect, recommencez ...";
            echo "<div id='msglogin'>";
            echo "<fieldset class=cadremsg>".$error."</fieldset>";
            echo "<div>";
            $step=0;
            //unset($step);
        }
        else {
            $_SESSION['profil'] = $profil;
            $_SESSION['nom'] = $nom;
            $_SESSION['login'] = $login;

            $step=1;
            echo "<div id='msgloginok'>";
            echo "Bienvenue ".ucwords($nom)." (Profil : ".$profil.")<br>";
            echo "Vous travaillez sur la collectivité ".ucwords($ville);
            echo "<div>";
        }
        $res->free();
    }
}
```

Login.php

3.2 Menu utilisateur

Le menu utilisateur est le même pour tous, seul l'accès à certains paramètres est bloqué (administration et paramétrage) pour les utilisateurs simples. Ainsi l'utilisateur ne pourra pas ajouter de droits ou d'autres utilisateurs.



Menu.inc

Ce menu est obtenu avec le fichier menu.js incluant le fichier menu.inc qui gère l'affichage. Menu.js permet d'afficher ou de cacher les sous menu au survol du curseur de la souris.

3.3 Ajouter un article

Si l'utilisateur souhaite insérer un article, il lui faut cliquer dans le menu correspondant afin de faire afficher le tableau contenant tous les articles en stock.

The screenshot shows the 'Open Stock' web application interface. At the top, there is a search bar with a 'Rechercher' button and a 'Réf.' icon. Below the search bar, the page title is 'Article Entrées/Sorties'. A notification indicates '1 - 3 enregistrement(s) sur 3'. The main content area displays a table of stock items with the following data:

	Idarticle	Libartcourt	Libartlong	Libfamille	Pmp	Prixvente	Quantité
	6	articletest	article test	TEST	12.59	20.00	189
	9	hp500	hp500	IMPRIMANTE	862.50	1000.00	4
	10	Ordinateur a	Ordinateur a	ORDINATEUR	1066.67	1200.00	2

The left sidebar contains navigation buttons: 'Réf. Article', 'Règle Régie & Liv', 'Traitement', 'Éditions', 'Paramétrage', 'Administration', 'À Propos', and 'Login admin'.

tab.php?obj=article

L'utilisateur peut donc modifier, supprimer ou ajouter un article. S'il choisit de modifier l'article, toutes les informations sont reprises dans le formulaire et sont alors modifiable. En revanche, s'il souhaite supprimer un article, les informations de ce dernier sont reprises mais sont bloquées.

Pour ce qui est de l'ajout, tous les champs sont vides et doivent être complétés. Des vérifications sont effectuées afin que l'utilisateur n'oublie pas de remplir un champ obligatoire.

The screenshot shows the 'Open Stock' web application interface. At the top left is the 'Open Stock' logo. At the top right is a 'Réf.' icon with a question mark. Below the logo is a navigation bar with 'Article' and 'Entrées/Sorties'. On the left side, there is a vertical menu with buttons for 'Réf. Article', 'Régie Regie & Liv', 'Traitement', 'Éditions', 'Paramétrage', 'Administration', 'À Propos', and 'Login admin'. The main content area is titled 'Article' and contains three sections: 'Article', 'Prix', and 'Solde'. The 'Article' section has fields for 'article :', 'Libelle :', and 'Famille :'. The 'Prix' section has fields for 'Pmp [Prix de vente] :', '[Quantite] :', and '[minimum] :'. The 'Solde' section has a 'Solde' dropdown menu set to 'Non', a 'Deliberation' label, and a 'Votre Choix' dropdown menu. At the bottom of the form is a red 'Ajouter Article' button with a back arrow icon.

form.php?obj=article

L'affichage général est donc géré par le fichier form.php mais la surcharge des méthode de dbform.php d'OpenMairie dans le fichier objet.class.php permet de modifier des champs avec par exemple la méthode setType() ou encore de remplir des listes déroulantes avec la méthode setSelect().

```
function setSelect(&$form,$maj,$db,$debug) {  
  
include ("../dyn/connexion.php");  
include ("../sql/".$dsn['phptype']."/".$this->table.".form.inc");  
if($maj<2) {  
    // requete qui recupere le libelle de la famille famille  
    $sql_famille="select idfamille, libfamille from famille order by libfamille";  
    $contenu="";  
    $res = $db->query($sql_famille);  
    if (DB :: isError($res))  
        die($res->getMessage());  
    else{  
        if ($debug == 1)  
            echo " la requete ".$sql_famille." est exécutée<br>";  
        $contenu[0][0]="";  
        $contenu[1][0]="Votre choix";  
        $k=1;  
        while ($row=& $res->fetchRow()){  
            $contenu[0][$k]=$row[0];  
            $contenu[1][$k]=$row[1];  
            $k++;  
        }  
        $form->setSelect("idfamille",$contenu);  
    }  
}  
}
```

Méthode setSelect()

On exécute la requête pour récupérer le libellé de la famille afin de l'afficher dans la liste déroulante.

3.4 Ajouter et livrer une commande

Lorsqu'on souhaite renouveler le stock d'un article, il faut passer une commande pour cet article et ensuite entrer sa livraison lorsqu'on la reçoit.

L'utilisateur choisit de passer une commande, il lui suffit pour cela d'ajouter une commande ainsi qu'une ou plusieurs lignes de commandes.

Open Stock

Commande

Réf. Article

Régie Regie & Liv

Traitement

Éditions

Paramétrage

Administration

À Propos

Login admin

Idcommande 21

Libellé JUIN

Idfournisseur DUCROS

Datecommande 20/06/2006

Commande c63214

Soldecommande Non

Modifier Commande

Lignes commande

1 - 1 enregistrement(s) sur 1

	Id	Libelle	Quantite	Prixachat
	39	Ordinateur a	2	1000

lignecommande

form.php?obj=commande

Lorsque la commande est entrée, à l'aide du tableau des lignes de commandes, l'utilisateur peut ajouter les articles qu'il souhaite commander. Il faut pour cela ajouter une ligne de commande.

The screenshot shows the 'Open Stock' web application interface. At the top left is the 'Open Stock' logo. On the right, there are icons for a shopping cart and a help question mark. The main heading is 'Mise à jour commande'. On the left side, there is a vertical menu with buttons for 'Article', 'Regie & Liv', 'Traitement', 'Éditions', 'Paramétrage', 'Administration', 'À Propos', and 'Login admin'. The main content area is titled 'Idlignecommande' and contains the following information:

Idcommande	21
Libcommande	JUIN

Below this, there is a section for 'Article' with a form containing:

Idarticle	<input type="text" value="Votre Choix"/>
Quantite	<input type="text"/>
Prixachat	<input type="text"/>

At the bottom of this section, there is a label 'Quantitelivraison' and a red button labeled 'Ajouter Lignecommande' with a left-pointing arrow icon.

sousform.php?idxfformulaire=21&objsf=lignecommande

L'utilisateur peut alors insérer la quantité d'article souhaité ainsi que le prix d'achat.

A la livraison des articles, il faut insérer les informations relatives à cette livraison (libellé, fournisseurs, commande correspondante, date) puis entrer les articles livrés. Il se peut qu'une commande ne soit pas complète alors une vérification est faite.

Lorsqu'on souhaite insérer une entrée, le nombre d'articles commandés est automatiquement inscrit dans le champ « quantité livrée », mais il reste accessible à l'écriture afin de le corriger dans le cas d'une livraison non complète. Dans le cas d'une deuxième livraison c'est le reste d'articles manquant qui est inscrit. Ainsi quand il n'y aura plus d'article à livrer, un message précisera que tous les articles ont bien été livrés.

Mise à jour des entrees

Réf. Article

Règle Regie & Liv

Traitement

Éditions

Paramétrage

A Administration

À Propos

Login
admin

identree :

Article

Idarticle	10
Libartcourt	Ordinateur a
date entree :	<input type="text" value="20/06/2006"/> 
Reste à livrer :	0
Prixachat	<input type="text" value="1000"/>

Livraison

Idlivraison	<input type="text" value="Votre Choix"/>
Idlignecommande	39
Idcommande	21
Libcommande	JUIN

Tous les articles ont été livrés

Ajouter Entre 

sousform.php?idxformulaire=39&objsf=entree

Pour effectuer ces vérification, il faut les lancer à l'ouverture du formulaire c'est-à-dire qu'on a surchargé la méthode setValsousformulaire() du composant OpenMairie. Lors du chargement de la page, une requête va compter le nombre d'articles déjà livrés et le retranche à celui du nombre total commandé. On retrouvera ainsi le nombre d'article manquant.

```

function setValsousformulaire($form,$maj,$validation,$idx,$idxformulaire,$retourformulaire,
                             $typeformulaire,&$db){
if ($validation==0) {
  if ($maj == 0){
    $debug=0;
    $form->setVal('idlignecommande',$idxformulaire);
    $sql = "select lignecommande.idcommande,lignecommande.prixachat, lignecommande.idarticle,
            article.libartcourt,lignecommande.quantite-lignecommande.quantitelivraison as quantite
            from lignecommande,article where lignecommande.idarticle=article.idarticle and
            idcommande = ".$idx." and idlignecommande=".$idxformulaire;
    $req = $db->query($sql);
    if (DB :: isError($req))
      die($req->getMessage());
    else{
      if ($debug == 1)
        echo " la requete ".$sql." est exécutée<br>";
      while($row=& $req->fetchRow())
      {
        $form->setVal('idcommande',$row[0]);
        $form->setVal('prixachat',$row[1]);
        $form->setVal('idarticle',$row[2]);
        $form->setVal('libartcourt',$row[3]);
        $form->setVal('quantite',$row[4]);
      }
      if($form->val['quantite']==0){
        $form->setType('quantite','hiddenstatic');
        $this->msg= $this->msg."<br>Tous les articles ont été livrés";
      }
      $form->setVal('dateentre',date('Y-m-d'));
    }

    $sql = "select libcommande from commande where idcommande = ".$idx;
    $req = $db->query($sql);
    if (DB :: isError($req))
      die($req->getMessage());
    else{
      if ($debug == 1)
        echo " la requete ".$sql." est exécutée<br>";
      while($row=& $req->fetchRow())
      {
        $form->setVal('libcommande',$row[0]);
      }
    }
  }
}
}

```

Méthode setValsousformulaire()

Une fois les entrées validées, la méthode « `pmp_qte_enplus()` » est appelée. Cette méthode permet l'ajout du nombre d'articles livrés ainsi que le calcul du prix moyen pondéré (pmp) utilisé pour la comptabilité.

```
function pmp_qte_enplus($db,$DEBUG)
{
    $sql = "select quantite from article where idarticle ="
    .$this->valF['idarticle'] ;
    $quantitearticle=$db->getOne($sql);
    $sql = "select pmp from article where idarticle ="
    .$this->valF['idarticle'] ;
    $pmparticle=$db->getOne($sql);
    $cle= " idarticle = ".$this->valF['idarticle'];
    $valF['quantite']= $quantitearticle+$this->valF['quantite'];
    $valF['pmp']= round(($pmparticle*$quantitearticle+$this->valF['quantite']*$this->valF['prixachat'])
        /($quantitearticle+$this->valF['quantite']),2);
    $res= $db->autoExecute('article',$valF,DB_AUTOQUERY_UPDATE,$cle);
    if (DB :: isError($res))
        die($res->getMessage()." échec ".$sql);
    else{
        if ($DEBUG == 1)
            echo "La requête de mise à jour est effectuée.<br>";
            $this->msg=$this->msg."Enregistrement ".$this->valF['idarticle']." de la table article"
            ." [".$db->affectedRows()." enregistrement mis à jour] " ;
    }
}
}
```

Méthode `pmp_qte_enplus()`

3.5 Effectuer une sortie

Etant donné que l'application est destinée à plusieurs services et que ceux-ci n'ont pas le même système de gestion des sorties d'articles, il a fallu adapter l'application de trois façons différentes.

3.5.1 Sorties pour les régies

Pour le service des finances, les sorties des articles se font en fonctions de régies. Une régie est un ensemble de sommes encaissées par le régisseur de la ville pendant une période. Une sortie doit donc être relative à une régie donnée.

Régie

Idregie 11
Libellé JUIN
Date debut 20/06/2006
Date fin 20/06/2006
Bordereau 21
Solderegie Non

Modifier Regie

Sorties

1 - 1 enregistrement(s) sur 1

+	Id	Date	Typesortie	Quantité	Prixvente
+	40	15/06/2006	vente	1	1200

sortieregie

form.php?idx=11&obj=regie

Pour insérer une nouvelle sortie l'utilisateur choisit de l'ajouter à l'aide du sous formulaire « sortieregie » et doit remplir les champs relatif à l'article.

Mise à jour des sorties

Idsortie

Article

Libarthlong Ordinateur a → Idarticle 10

Prix 1200

quantite restante 2

Quantite

Typesortie vente

Datesortie 20/06/2006

Idregie 11

Libregie JUIN

Ajouter Sortie

sousform.php?idxformulaire=11&objsf=sortieregie

En utilisant le type « combo » d'OpenMairie (voir annexe 3), on récupère, grâce à du code javascript, l'idarticle, le prix et la quantité restante de l'article sélectionné. Il reste alors à choisir la quantité à sortir. Une vérification de la quantité est alors effectuée et on fait aussi appel à deux autres méthodes : « verifierDate() » et « quantite_enmoins() ».

- La méthode « verifierDate() » lance le code qui valide la date c'est-à-dire que la sortie est effectuée entre la date de début et la date de fin de la régie,
- La méthode « quantite_enmoins » permet de soustraire le nombre d'articles sortis au nombre d'articles en stock pour le mettre à jour. Si on supprime une sortie, c'est la méthode « quantite_enplus » qui est appelée afin de rajouter le nombre d'article pour mettre à jour les stocks.

```
//verifie la coherence des dates (une sortie doit etre comprise entre le debut
// et la fin de la regie
function verifierDate($db,$val,$DEBUG) {
  if ($this->valF['datesortie']!=""){ //verifie si la date est entrée
    $this->correct=false;
    $this->msg= $this->msg."<br>date sortie obligatoire";
  }else{ //si elle l'est récupère sa valeur
    $this->valF['datesortie'] = $this->dateDB($this->valF['datesortie']);
    $this->correct=true;
  }
  $regie= $this->valF['idregie']; //on récupère l idregie
  $sql = "select datedebut from regie where idregie=".$regie; //requete pour obtenir la date debut
  $datedebut = $db->getOne($sql); //on insère la valeur de la date dans la variable
  $req = "select datefin from regie where idregie=".$regie; //requete pour obtenir la date fin
  $datefin = $db->getOne($req); //on insère la valeur de la date dans la variable
  $datesortie = $this->valF['datesortie']; //on récupère la valeur de la date sortie entrée
  if(($datesortie < $datedebut) || ($datesortie > $datefin)){ //on vérifie si la date est bien comprise
    $this->msg= $this->msg."<br>Date non comprise dans la regie";
    $this->correct=false;
    //NE PAS FAIRE LA MISE A JOUR => ANNULER LA TRANSACTION!!!!
  }
  else{
    $this->correct=true;
    //FAIRE LA TRANSACTION!!!
  }
}
```

Méthode verifierDate()

```

function quantite_enmoins($db,$DEBUG)
{
    // initialisation correct (au lieu de verifier)
    $this->correct=True;
    // =====
    $sql ="select quantite from article where idarticle ="
    .$this->valF['idarticle'] ;
    $quantitearticle=$db->getOne($sql);
    //test coherence
    If ($this->valF['quantite'] <= $quantitearticle){
        $cle= " idarticle = ".$this->valF['idarticle'];
        $valF['quantite']= $quantitearticle-$this->valF['quantite'];
        $res= $db->autoExecute('article',$valF,DB_AUTOQUERY_UPDATE,$cle);
        if (DB :: isError($res))
            die($res->getMessage()." échec ".$sql);
        else{
            if ($DEBUG == 1)
                echo "La requête de mise à jour est effectuée.<br>";
            $this->msg="Enregistrement ".$this->valF['idarticle']." de la table article"
                ." [".$db->affectedRows()." enregistrement mis à jour" ] ;
        }
    }else{
        $this->correct=false;
        $this->msg= $this->msg."<br>Quantite superieure à la quantite en stock";
    }
}

function quantite_enplus($db,$DEBUG)
{
    // initialisation correct (au lieu de verifier)
    $this->correct=True;
    // =====
    $sql ="select quantite from article where idarticle =".$this->valF['idarticle'];
    $quantitearticle=$db->getOne($sql);
    $cle= " idarticle = ".$this->valF['idarticle'];
    $valF['quantite']= $quantitearticle+$this->valF['quantite'];
    $res= $db->autoExecute('article',$valF,DB_AUTOQUERY_UPDATE,$cle);
    if (DB :: isError($res))
        die($res->getMessage()." échec ".$sql);
    else{
        if ($DEBUG == 1)
            echo "La requête de mise à jour est effectuée.<br>";
        $this->msg="Enregistrement ".$this->valF['idarticle']." de la table article"
            ." [".$db->affectedRows()." enregistrement mis à jour" ] ;
    }
}

```

Méthodes quantite_enmoins() et quantite_enplus()

3.5.2 Sorties pour les services

Pour le service informatique, les sorties des articles se font en fonctions des autres services de la mairie. En effet lorsqu'un service souhaite avoir une cartouche, une personne vient à la DSIT et la secrétaire enregistre la sortie pour le service.

Mise à jour des sorties

Idsortie

Article

Libartlong LASER POUR HP5L / HP6L / CANON LBP → Idarticle 12

Prix 75.29

quantite restante 0

Quantite

Typesortie vente

Datesortie 20/06/2006

idservice 100

Libservice SECRETARIAT GENERAL

Ajouter Sortie

sousform.php?idxformulaire=100&objsf=sortieservice

La gestion des sorties est la même que celle de régie sauf qu'il n'y a pas de vérification sur les dates. Une sortie peut avoir lieu n'importe quand, la seule condition est que l'article soit en stock. Lorsqu'on sort des articles ou qu'on supprime une sortie, les méthodes de calcul des stocks sont appelés.

3.5.3 Sorties pour les factures

Pour le service des pompes funèbres, les sorties des articles ou des prestations sont facturées à un client. Il a donc fallu adapter un système de facturation qui regroupe les articles vendus ainsi que les prestations.

Après avoir ajouter le client, une facture lui est adressée avec les articles qu'il souhaite prendre ainsi que les prestations.

Facture

Idfacture 6

Numerofacture Date

Etat Conseilier :

Client ou Service : **Client**

Client

Idclient Nom

Reglement

Delaireglement Codereglement

Modereglement

Reçu

Numerotitre Exercice

Daterecu Numereecu

Observation

Modifier Facture

Sorties prestations Sorties articles

1 - 2 enregistrement(s) sur 2

	Id	Date	Typesortie	Quantite	Prixvente
	40	21/06/2006	vente	1	50
	42	21/06/2006	vente	1	34

sortiefactureprestation

form.php?idx=6&obj=facture

L'employé ajoute ensuite les prestations et les articles pour le client de la même façon que vu précédemment. La différence est que les prestations et les articles peuvent posséder une TVA qu'il faut comptabiliser dans la facture.

Idsortie			
Idfacture	6		
Prestation			
Libprestation	CONVOI EN VILLE	Idprestation	P6
		Prixvente	50
		Tva	5.5
		Quantite	1
		Montanttva	2.75
		Montantttc	52.75
Typesortie	vente		
Datesortie	21/06/2006		
		Ajouter Sortie	

sousform.php?idxformulaire=6&objsf=sortiefactureprestation

Du code javascript permet de calculer automatiquement le montant TVA ainsi que le montant TTC.

Une fois que la facture, les articles et les prestations sont compléter, il faut éditer la facture (voir annexe 4).

3.6 Editer des états

L'utilisateur a la possibilité d'éditer plusieurs états. En effet certains sont déjà accessible comme l'édition du stock comptable, les sorties effectuées par les régies, les services et d'autres encore. L'administrateur peut éventuellement créer des états ou en modifier d'autres en choisissant dans le menu « paramétrage ».

L'administrateur choisit l'état ou sous état qu'il souhaite modifier et pourra alors changer toute l'apparence (couleurs, bordures,...) mais aussi modifier la requête qui est effectuée pour récupérer les différents champs qu'il souhaite afficher. Un fichier « objet.sousetat.inc » est alors créé ou modifier lorsqu'on ajoute ou modifie les états.

```

<?php
//12/06/2006 10:03:04
$sousetat['titre']="Liste des prestations";
$sousetat['titrehauteur']="8";
$sousetat['titrefont']="helvetica";
$sousetat['titreattribut']="B";
$sousetat['titretaille']="12";
$sousetat['titrebordure']="1";
$sousetat['titrealign']="C";
$sousetat['titrefond']="1";
$sousetat['titrefondcouleur']=array("109","219","85");
$sousetat['titretextecouleur']=array("255","255","85");
$sousetat['intervalle_debut']=5;
$sousetat['intervalle_fin']=15;
$sousetat['entete_flag']="1";
$sousetat['entete_fond']="1";
$sousetat['entete_hauteur']="10";
$sousetat['entetecolone_bordure']=array("TL","LTB","LTB","LTB","LTBR","LTB","LTBR","LTBR");
$sousetat['entetecolone_align']=array("C","C","C","C","C","C","C","C");
$sousetat['entete_fondcouleur']=array("248","204","160");
$sousetat['entete_textecouleur']=array("60","0","0");
$sousetat['tableau_largeur']="195";
$sousetat['tableau_bordure']="1";
$sousetat['tableau_fontaille']="10";
$sousetat['bordure_couleur']=array("0","0","0");
$sousetat['se_fond1']=array("243","246","246");
$sousetat['se_fond2']=array("255","255","255");
$sousetat['cellule_fond']="1";
$sousetat['cellule_hauteur']="10";
$sousetat['cellule_largeur']=array("35","30","20","25","20","15","25","25");
$sousetat['cellule_bordure_un']=array("TL","TL","LTB","LTB","LTBR","LTB","LTBR","LTBR");
$sousetat['cellule_bordure']=array("TL","LTB","LTB","LTB","LTBR","LTB","LTBR","LTBR");
$sousetat['cellule_align']=array("C","C","C","C","C","C","C","C");
$sousetat['cellule_fond_total']="1";
$sousetat['cellule_fontaille_total']="10";
$sousetat['cellule_hauteur_total']="15";
$sousetat['cellule_fondcouleur_total']=array("250","242","234");
$sousetat['cellule_bordure_total']=array("TBL","TB","TB","TBR","TBL","TBLR","TBLR","TBLR");
$sousetat['cellule_align_total']=array("L","L","C","C","C","C","C","C");
$sousetat['cellule_fond_moyenne']="1";
$sousetat['cellule_fontaille_moyenne']="10";
$sousetat['cellule_hauteur_moyenne']="5";
$sousetat['cellule_fondcouleur_moyenne']=array("250","232","214");
$sousetat['cellule_bordure_moyenne']=array("BTL","BT","BTLR","BTLR","BTLR","BTLR","BTLR","BTLR");
$sousetat['cellule_align_moyenne']=array("C","C","C","R","C","C","C","C");
$sousetat['cellule_fond_nbr']="1";
$sousetat['cellule_fontaille_nbr']="10";
$sousetat['cellule_hauteur_nbr']="15";
$sousetat['cellule_fondcouleur_nbr']=array("249","218","188");
$sousetat['cellule_bordure_nbr']=array("TBL","TB","TB","TBR","TB","TBLR","TBLR","TBLR");
$sousetat['cellule_align_nbr']=array("L","L","C","C","C","C","C","C");
$sousetat['cellule_numerique']=array("999","999","0","2","2","2","2","2");
$sousetat['cellule_total']=array("0","0","0","0","1","0","1","1");
$sousetat['cellule_moyenne']=array("0","0","0","0","0","0","0","0");
$sousetat['cellule_compteur']=array("0","0","0","0","0","0","0","0");
$sousetat['sql']="select prestation as prestations,categorie, sortie.quantite,
sortie.prixvente as 'Prix unitaire', sortie.quantite * sortie.prixvente as 'Montant HT',
sortie.tva, sortie.montanttva, sortie.montantttc as 'Montant TTC' from sortie
inner join prestation on sortie.idarticle=prestation.idprestation
inner join categorie on categorie.idcategorie=prestation.idcategorie
where idfacture= 'fidx'";

```

Sous état facture prestation

Liste des prestations

Titre

Paramètres Titre

Hauteur font

Taille Bordure

Titrefond Fond Texte

Intervalle Début Fin

Entete du Tableau

Flag Fond Hauteur

bordure
LTB
LTB
LTB align
c
c
c Fond Texte

Data du Tableau

largeur bordure taille

bordure fond1 fond2

Cellule du Tableau

Hauteur

Largeur
30
20
25 bordure1
TL
LTB
LTB bordure
LTER
LTER align
c
c
c

Total du Tableau

Taille Hauteur

Fond Bordure
TB
TB
TER Align
L
c
c

moyenne du Tableau

Taille Hauteur

Fond Bordure
ETLR
ETLR
ETLR Align
c
c
B

nombre enregistrement du Tableau

Taille Hauteur

Fond Bordure
TB
TB
TER Align
L
c
c

Opérations du Tableau

Numérique
999
0
z Total
0
0
0 Moyenne
0
0
0 Nombre
0
0
0

Sql

```
select prestation as prestations,categorie, sortie.quantite, sortie.prixvente as 'Prix unitaire', sortie.quantite * sortie.prixvente as 'Montant HT', sortie.tva, sortie.montanttva, sortie.montantttc as 'Montant TTC' from sortie inner join prestation on sortie.idarticle=prestation.idprestation inner join categorie on categorie.idcategorie=prestation.idcategorie where idfacture= 'fidx'
```

Modifier Souseetat

txform.php?obj=souseetat&idx=sortiefactureprestation

Références bibliographiques et réseaugraphiques

Portail de openMairie.

<http://www.openmairie.org/>

Site démonstration openStock.

http://demo.openmairie.org/openmairie_stock/index.php

Librairie Pear, le connecteur de base de données utilisé dans le cadre du projet.

<http://pear.php.net/>

GFORGE de l'ADULLACT.

<http://adullact.net/>

openStock sur le GFORGE.

<http://adullact.net/projects/openstock/>

Site d'aide en français sur PHP

<http://www.phpfrance.com/>

Forums d'aide sur PHP

<http://php.developpez.com/>

Annexe 1 : Description textuelle des cas d'utilisation

Titre

Ajout article ou prestation

Résumé

L'utilisateur ou l'administrateur ajoute les articles qu'il reçoit.

Acteurs

L'utilisateur (acteur principal)

Pré-condition

Le site est en ligne

Scénario nominal

1. L'utilisateur se connecte au site
2. Le système demande ses identifiants à l'utilisateur
3. L'utilisateur saisit son login et son mot de passe
4. Le système vérifie la validité des informations saisies
5. Le système affiche le menu
6. L'utilisateur choisit d'ajouter les articles dans le menu
7. L'utilisateur choisit s'il veut ajouter, modifier ou supprimer un article
8. Le système affiche les informations à compléter ou à modifier
9. L'utilisateur complète les informations
10. Le système enregistre les modifications

Enchaînements alternatifs

A1 : L'identification est incorrecte

L'enchaînement A1 démarre après le point 4 du scénario nominal

5. Le système indique que l'identification a échoué

Le scénario nominal reprend au point 2

Titre

Ajout commande

Résumé

L'utilisateur ou l'administrateur ajoute les commande qu'il souhaite passer.

Acteurs

L'utilisateur (acteur principal)

Pré-condition

Le site est en ligne

Scénario nominal

1. L'utilisateur se connecte au site
2. Le système demande ses identifiants à l'utilisateur
3. L'utilisateur saisit son login et son mot de passe
4. Le système vérifie la validité des informations saisies
5. Le système affiche le menu
6. L'utilisateur choisit d'ajouter les commandes dans le menu
7. L'utilisateur choisit s'il veut ajouter, modifier ou supprimer une commande
8. Le système affiche les informations à compléter ou à modifier
9. L'utilisateur complète les informations
10. Le système enregistre les modifications

Enchaînements alternatifs

A1 : L'identification est incorrecte

L'enchaînement A1 démarre après le point 4 du scénario nominal

6. Le système indique que l'identification a échoué

Le scénario nominal reprend au point 2

Titre

Ajout livraison

Résumé

L'utilisateur ou l'administrateur ajoute les livraisons qu'il reçoit.

Acteurs

L'utilisateur (acteur principal)

Pré-condition

Le site est en ligne

Scénario nominal

1. L'utilisateur se connecte au site
2. Le système demande ses identifiants à l'utilisateur
3. L'utilisateur saisit son login et son mot de passe
4. Le système vérifie la validité des informations saisies
5. Le système affiche le menu
6. L'utilisateur choisit d'ajouter les livraisons dans le menu
7. L'utilisateur choisit s'il veut ajouter, modifier ou supprimer une livraison
8. Le système affiche les informations à compléter ou à modifier
9. L'utilisateur complète les informations
10. Le système enregistre les modifications

Enchaînements alternatifs

A1 : L'identification est incorrecte

L'enchaînement A1 démarre après le point 4 du scénario nominal

7. Le système indique que l'identification a échoué

Le scénario nominal reprend au point 2

Titre

Ajout régie, service ou client.

Résumé

L'utilisateur ou l'administrateur ajoute les régies, les services ou les clients qui prennent un article.

Acteurs

L'utilisateur (acteur principal)

Pré-condition

Le site est en ligne

Scénario nominal

1. L'utilisateur se connecte au site
2. Le système demande ses identifiants à l'utilisateur
3. L'utilisateur saisit son login et son mot de passe
4. Le système vérifie la validité des informations saisies
5. Le système affiche le menu
6. L'utilisateur choisit d'ajouter les régies, services ou clients dans le menu
7. L'utilisateur choisit s'il veut ajouter, modifier ou supprimer un client
8. Le système affiche les informations à compléter ou à modifier
9. L'utilisateur complète les informations
10. Le système enregistre les modifications

Enchaînements alternatifs

A1 : L'identification est incorrecte

L'enchaînement A1 démarre après le point 4 du scénario nominal

8. Le système indique que l'identification a échoué

Le scénario nominal reprend au point 2

Titre

Ajout sortie.

Résumé

L'utilisateur ou l'administrateur ajoute les sorties en fonction du client.

Acteurs

L'utilisateur (acteur principal)

Pré-condition

Le site est en ligne

Scénario nominal

1. L'utilisateur se connecte au site
2. Le système demande ses identifiants à l'utilisateur
3. L'utilisateur saisit son login et son mot de passe
4. Le système vérifie la validité des informations saisies
5. Le système affiche le menu
6. L'utilisateur choisit d'ajouter les sorties en fonction du client
7. L'utilisateur choisit s'il veut ajouter, modifier ou supprimer une sortie
8. Le système affiche les informations à compléter ou à modifier
9. L'utilisateur complète les informations
10. Le système enregistre les modifications

Enchaînements alternatifs

A1 : L'identification est incorrecte

L'enchaînement A1 démarre après le point 4 du scénario nominal

9. Le système indique que l'identification a échoué

Le scénario nominal reprend au point 2

Titre

Traitement.

Résumé

L'administrateur peut archiver les entrées et sorties qu'il y a eut au cours de l'année et faire aussi une reconstitution d'écriture.

Acteurs

L'administrateur (acteur principal)

Pré-condition

Le site est en ligne

Scénario nominal

1. L'administrateur se connecte au site
2. Le système demande ses identifiants à l'administrateur
3. L'administrateur saisit son login et son mot de passe
4. Le système vérifie la validité des informations saisies
5. Le système affiche le menu
6. L'administrateur choisit d'effectuer un traitement dans le menu
7. L'administrateur choisit le traitement à effectuer
8. Le système demande une confirmation
9. L'administrateur confirme la demande
10. Le système enregistre les modifications

Enchaînements alternatifs

A1 : L'identification est incorrecte

L'enchaînement A1 démarre après le point 4 du scénario nominal

10. Le système indique que l'identification a échoué

Le scénario nominal reprend au point 2

Titre

Edition état comptable

Résumé

L'utilisateur ou l'administrateur peut éditer l'état comptable.

Acteurs

L'utilisateur (acteur principal)

Pré-condition

Le site est en ligne

Scénario nominal

1. L'utilisateur se connecte au site
2. Le système demande ses identifiants à l'utilisateur
3. L'utilisateur saisit son login et son mot de passe
4. Le système vérifie la validité des informations saisies
5. Le système affiche le menu
6. L'utilisateur choisit d'éditer l'état
7. L'utilisateur choisit l'état qu'il souhaite
8. Le système affiche les informations
9. L'utilisateur demande l'impression
10. Le système imprime

Enchaînements alternatifs

A1 : L'identification est incorrecte

L'enchaînement A1 démarre après le point 4 du scénario nominal

11. Le système indique que l'identification a échoué

Le scénario nominal reprend au point 2

Titre

Ajout paramètres.

Résumé

L'administrateur peut paramétrer l'application en ajoutant les familles des articles, les différents fournisseurs, pour la régie s'il y eut des délibérations, paramétrer les différents états et sous état qu'il souhaite.

Acteurs

L'administrateur (acteur principal)

Pré-condition

Le site est en ligne

Scénario nominal

1. L'administrateur se connecte au site
2. Le système demande ses identifiants à l'administrateur
3. L'administrateur saisit son login et son mot de passe
4. Le système vérifie la validité des informations saisies
5. Le système affiche le menu
6. L'administrateur choisit d'effectuer un paramétrage dans le menu
7. L'administrateur choisit le paramètre à effectuer
8. Le système affiche les informations à remplir
9. L'administrateur complète les informations
10. Le système enregistre les modifications

Enchaînements alternatifs

A1 : L'identification est incorrecte

L'enchaînement A1 démarre après le point 4 du scénario nominal

12. Le système indique que l'identification a échoué

Le scénario nominal reprend au point 2

Titre

Administration.

Résumé

L'administrateur paramètre la collectivité, gère les utilisateurs, les profils et la gestion des droits d'accès.

Acteurs

L'administrateur (acteur principal)

Pré-condition

Le site est en ligne

Scénario nominal

11. L'administrateur se connecte au site
12. Le système demande ses identifiants à l'administrateur
13. L'administrateur saisit son login et son mot de passe
14. Le système vérifie la validité des informations saisies
15. Le système affiche le menu
16. L'administrateur choisit d'administrer l'application dans le menu
17. L'administrateur choisit ce qu'il souhaite gérer
18. Le système affiche les informations à remplir
19. L'administrateur complète les informations
20. Le système enregistre les modifications

Enchaînements alternatifs

A1 : L'identification est incorrecte

L'enchaînement A1 démarre après le point 4 du scénario nominal

13. Le système indique que l'identification a échoué

Le scénario nominal reprend au point 2

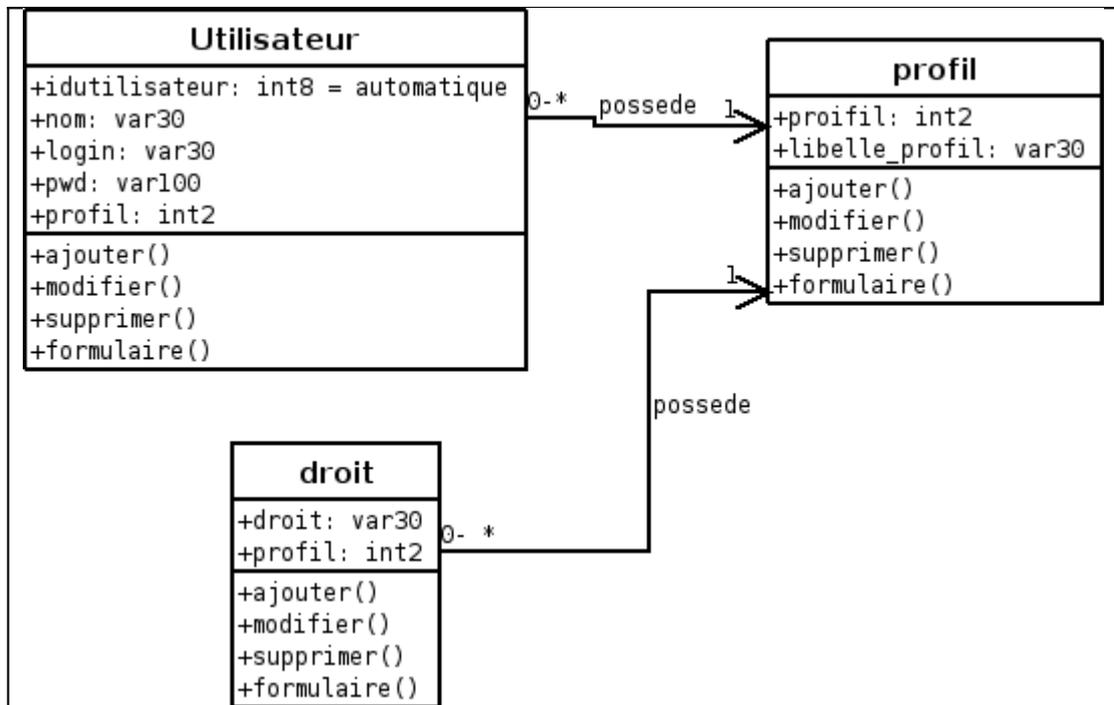
Annexe 2 : Gestion des utilisateurs

Diagramme de classes des utilisateurs

Annexe 3 : Quelques états


Monsieur DURANT PAUL
RUE DE LA REPUBLIQUE
13200 ARLES

Liste des prestations							
PRESTATIONS	CATEGORIE	QUANTITE	PRIX UNITAIRE	MONTANT HT	TVA	MONTANT TVA	MONTANT TTC
CONVOI EN VILLE	transport	1	50,00	50,00	5,50	2,75	52,75
MISE EN BIÈRE	traitement	1	34,00	34,00	19,60	6,66	40,66
TOTAL				84,00		9,41	93,41

Liste des articles							
ARTICLES	FAMILLE	QUANTITE	PRIX UNITAIRE	MONTANT HT	TVA	MONTANT TVA	MONTANT TTC
SAPIN	CERCUEIL	1	150,00	150,00	0,00	0,00	150,00
TOTAL				150,00		0,00	150,00

Total		
MONTANT HT	MONTANT TVA	MONTANT TTC
234,00	9,41	243,41



**ENTREES ET
SORTIES DE L'ARTICLE
SAPIN**

Liste des entrées de l'article

REF	LIBELLÉ LIVRAISON	DATE ENTRÉE	QUANTITÉ	PRIX ACHAT	PMP
1	JUIN	21/06/2006	2	100,00	100,00

Liste des sorties de l'article

REF	NUMERO FACTURE	DATE SORTIE	TYPE SORTIE	QUANTITÉ	PRIX VENTE
41	F012542	21/06/2006	vente	1	150,00