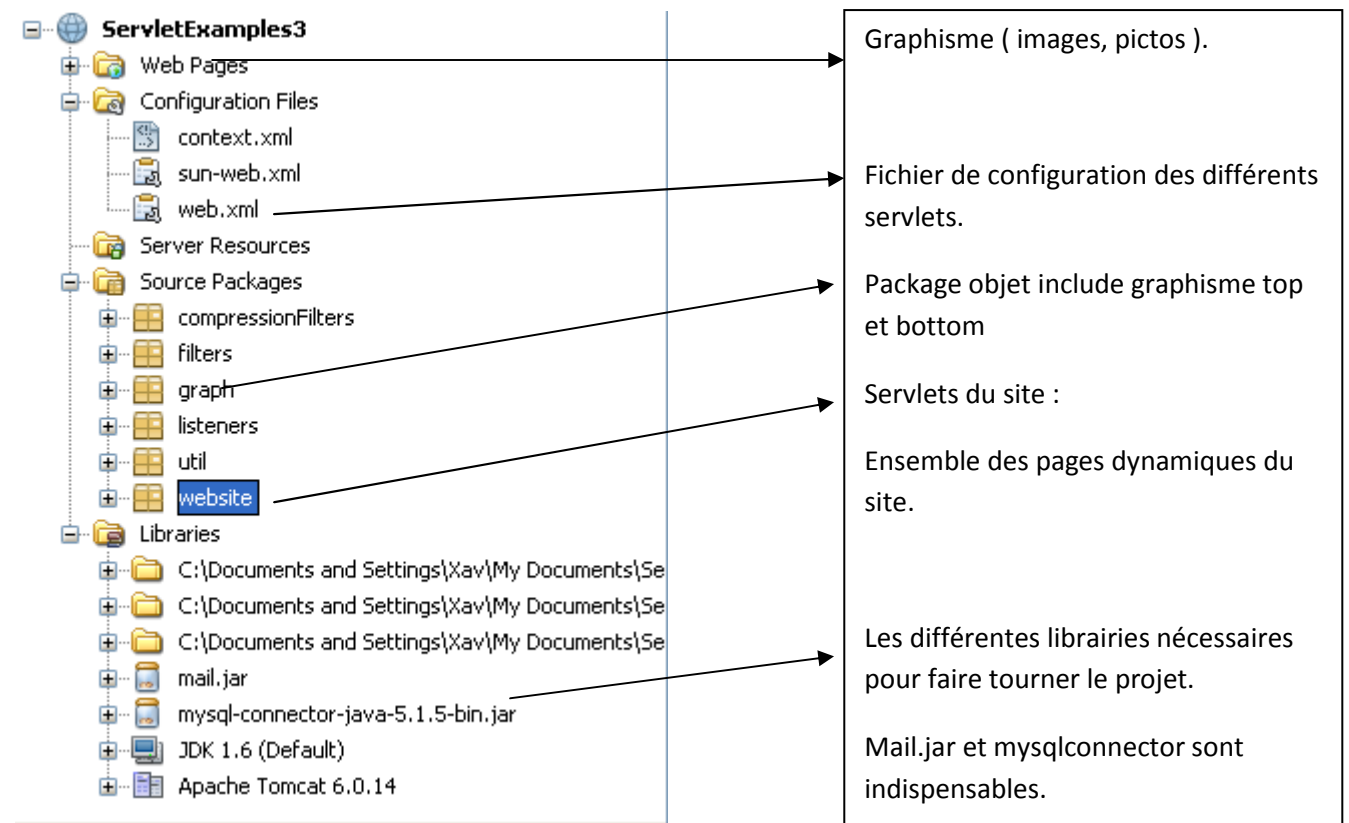


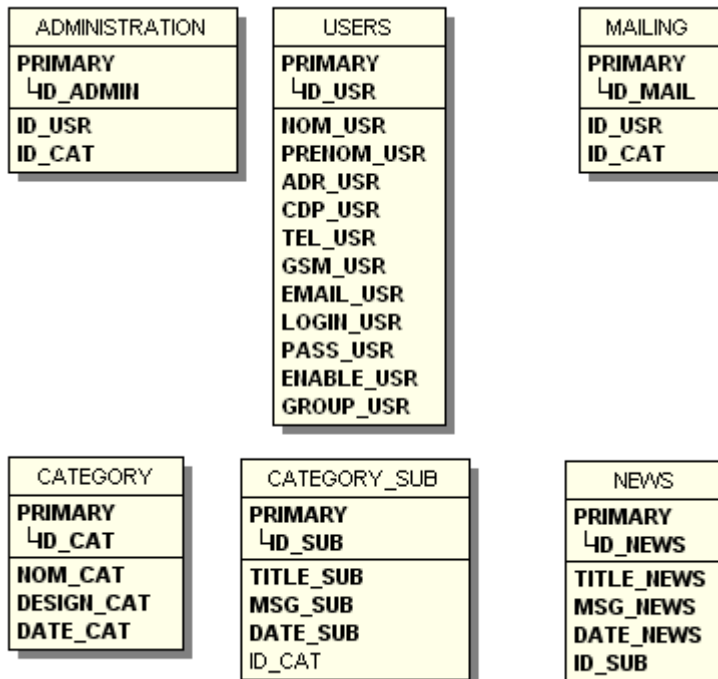
Projet java : Documentation.



Architecture du projet :



La base de données :



Administration :

Reprend l'utilisateur et la catégorie qu'il peut administrer.

Users : ensemble des utilisateurs avec leur niveau.

Mailing : reprend quel utilisateur est abonné à quelle catégorie.

Category : nom-libellé – date de création de la catégorie

Category sub : sous catégorie composant les catégories.

News : news composant les sous catégories.

Javadoc :

La javadoc a été en grande partie implémenté dans le projet :

- 1) La description des servlets
- 2) Les définitions des fonctions de mysqlaccess.
- La description des packages n'a pas été faite ☹

1) La description des servlets :

Package website

Class Summary	
Accueil	Page d'accueil du site.
AddAdminCategory	Servlet Ajouter un utilisateur de lvl 1 à une catégorie
AddCategory	Servlet Ajouter une catégorie dans la BDD
AddCatMailing	Servlet S'abonner à la mailing liste d'une catégorie.
AddNews	Servlet Ajouter une news à une catégorie
Dates	Servlet générant la date du jour.
EditCatMailingUser	Servlet gestion des abonnements de la mailing liste
EditUsersCat	Servlet gestion des administrateurs de catégories.
ListAdmin	Servlet liste des administrateurs (page inactive).
ListMain	Servlet affichage des catégories et sous-catégories avec gestion des permissions.
ListNews	Servlet Liste des news d'une sous-catégories

2) La classe mysqlaccess :

Constructor Summary	
MySQLAccess	()

Method Summary	
void	AddAdminCat (java.sql.Connection connexion, java.lang.String IDUSER, java.lang.String cat) ajout d'un administrateur à une catégorie
void	AddCategory (java.sql.Connection connexion, java.lang.String name, java.lang.String libelle) Ajout d'une catégorie
void	AddCatMailing (java.sql.Connection connexion, java.lang.String IDUSER, java.lang.String cat) ajout d'une catégorie à ma mailing liste
void	AddNews (java.sql.Connection connexion, java.lang.String title, java.lang.String libelle, java.lang.String subcat) ajouter une news à une sous catégorie
void	AddSubCategory (java.sql.Connection connexion, java.lang.String name, java.lang.String libelle, java.lang.String subcatget) ajout d'une sous catégorie
java.lang.Boolean	CheckAdminSubcat (java.sql.Connection connexion, java.lang.String ID_USR, java.lang.String Cat) vérifier que l'utilisateur est bien admin de la catégorie

Présentation des différents écrans ainsi que leurs codes source :

La page session :

Cette page propose à l'utilisateur de se connecter au site.



Accueil
Login
Liste Cat.
Mailing
Edit Mailing
Liste Users
S'enregistrer

Authentification :

Login :

Password :

Envoyer

Projet Java - Badot Xavier - Esa 3 ème Info

Dans le cas ou l'utilisateur se voit l'accès autorisé :

```
true Level accordé : 2 | IDUSR _ : 1  
Bienvenue - superadmin
```

[Se déloguer !](#)

Le système se chargera de lui afficher le niveau qu'il lui est accordé.

Et il lui souhaitera la bienvenue 😊

*Les commentaires pour le debug sont volontairement laissés affichés dans cette présentation.

Dans le cas d'une erreur :

```
false
```

Authentification :

Login :

Password :

Envoyer

Dans le cas d'une erreur le système nous retourne false !

Les messages d'erreur seront changés enfin de projet.

```

1 public void doGet(HttpServletRequest request,
2                   HttpServletResponse response)
3     throws IOException, ServletException
4     {
5         response.setContentType("text/html");
6
7         PrintWriter out = response.getWriter();
8         /** On init l'objet graphique et on affiche le haut */
9         incgraph getgraph = new incgraph();
10        String top = getgraph.top();
11        out.println(top);
12
13        HttpSession session = request.getSession(true);
14        /** Déclaration et récupération des variables du formulaire */
15        String Login = request.getParameter("dataLogin");
16        String Password = request.getParameter("dataPassword");
17        String Connect = request.getParameter("connect");
18        String unlog = null; /** on voit si il veut se délogger */
19        unlog = request.getParameter("unlog");
20
21
22        /** Session actuelle */ :
23        String session_name = (String)
session.getAttribute("session_name");
24        String session_lvl = (String)
session.getAttribute("session_lvl");
25        String valideyes = "yes";
26        /** si l'on veut se délogger, on délogue et on affichera le
formulaire de login en bas de page */
27        if(unlog!=null){
28            if (unlog.equals("yes"))
29            {
30                // On réinitialise tout à ZERO !
31                session.putValue("session_name",null);
32                session.putValue("session_lvl",null);
33                session_name=null;
34                session_lvl=null;
35            }
36        }
37        /** On initialise les valeurs sessions à null : */
38        boolean valide = false;
39        String LevelAdmin = null;
40        String ID_USR=null;
41        /** On cherche les infos session si le formulaire de login a été
remplit et si les variables sont différentes de null ! : */
42

```

```

43     if(Login != null && Login!="" && Password != "" && Password !=
null && Connect!=null) {
44     /** on ouvre la connexion */
45         MysqlAccess Conn = new MysqlAccess();
46         Connection connexion = Conn.Open();
47         if(connexion!=null) {
48             valide = Conn.GetSession(Login, Password, connexion);
49             out.println(valide); /** on demande la confirmation que
le login et le mot de passe sont bon */
/** Récupération Level et ID */
50             LevelAdmin = Conn.CheckLevelAdmin(connexion, Login);
51             ID_USR = Conn.Get_ID_USER(connexion, Login);
52
53         }
54
55     }
56
57 /** Si l'utilisateur est valide on set les données dans la session */
58
59
60
61     String sess_ID_USR=null;
62     if (valide==true){
63         session.putValue("session_name",Login);
64         session.putValue("session_lvl",LevelAdmin);
65         session.putValue("session_ID_USR",ID_USR);
66         /** on affiche les variables de la session : */
67
68
69         out.println("Level accordé : "+
session.getAttribute("session_lvl").toString() + " | IDUSR _ :
"+session.getAttribute("session_ID_USR").toString()+" <br>");
70
71
72     }
73     /** Si la session est nul on propose a l'utilisateur de se logger ☺
*/
74     if(session_name==null){
75         out.println("<P>");
76         out.println("Authentication :");
77         out.print("<form action=\"");
78         out.print(response.encodeURL("Session"));
79         out.print("\ " ");
80         out.println("method=POST>");
81
82         out.println("Login : <input type=text size=20
name=dataLogin>");
83         out.println("<br>");
84
85         // out.println(rb.getString("sessions.datavalue"));

```

```

86
87     out.println("Password : <input type=password size=20
name=dataPassword>");
88     out.println("<br>");
89     out.println("<input type=hidden name=connect value=true>");
90     out.println("<input type=submit>");
91     out.println("</form>");
92
93
94     out.println("</body>");
95     out.println("</html>");
96 }
97 else {
98     out.println("Bienvenue - "+session_name+" <br><br>");
99     /** sinon c'est qu'il est loggué on lui propose de se le délogguer
*/
100     out.println("<a href='./Session?unlog=yes'>Se délogguer
!</a>");
101
102 }
103 /** On affiche la fin du graphisme grâce a l'objet getgraph., on
selectionne le bon graphisme en fonction du lvl_user */
104     String bottom = "";
105     if(session_lvl==null) {
106         bottom = getgraph.bottom("0");
107     } else {
108         bottom = getgraph.bottom(session_lvl);
109     }
110     out.println(bottom);
111
112 }

```

Grâce à cette page nous définissons donc le niveau de l'utilisateur :

Les différents niveaux :

Level 2 : L'administrateur tout puissant (superadmin).

Il possède tous les droits.

Level 1 : Les administrateurs simples :

Ils ont des droits sur les catégories qu'ils leurs sont attribués par le superadmin.

- Edition des catégories sur lesquels il a des droits !
- Poster des news dans les catégories dont il est administrateur.
- Envoi d'un mail lorsqu'il ajoute une news.

Level 0 : L'utilisateur n'a aucun droit ! Juste de consulter les catégories / news.

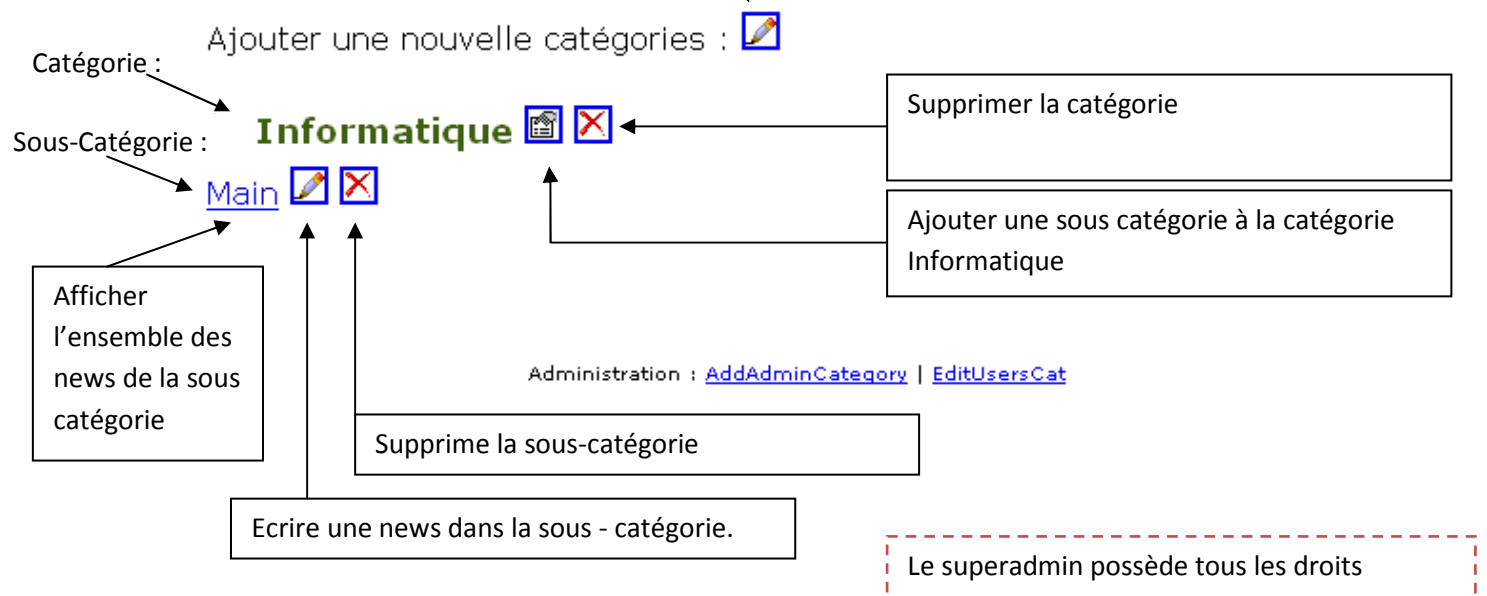
Ainsi que s'abonné à la mailing liste d'une catégorie.

Les différentes options disponibles pour les utilisateurs seront affichées dans les pages même.

Utilisateur de niveau 2 :

La page Liste Catégorie (/Listmain) :

Sur cette page le superadmin va pouvoir ajouter une catégorie



Utilisateur de niveau 1 :

Informatique

Main  

Afficher
l'ensemble des
news de la sous
catégorie

Ecran dans le cas ou l'utilisateur de niveau 1 est
administrateur de la catégorie !!!!

Supprime la sous-catégorie

Ecrire une news dans la sous - catégorie.

Informatique

Main

Dans le cas ou il n'est pas administrateur de la catégorie :

- Il aura simplement le droit de consulter les messages de la catégorie !

Utilisateur de niveau 0 :

Informatique

Main

Afficher
l'ensemble des
news de la sous
catégorie

- Juste le droit de consulter les news.

Servlet /ListMain :

```
1  protected void processRequest(HttpServletRequest request,
   HttpServletResponse response)
2      throws ServletException, IOException {
3      response.setContentType("text/html;charset=UTF-8");
4      PrintWriter out = response.getWriter();
5      try {
6
7          /** création de l'objet et include du graphisme */
8          incgraph getgraph = new incgraph();
9          String top = getgraph.top();
10         out.println(top);
11         /** récupération des variables session */
12         HttpSession session = request.getSession(true);
13         String session_name = (String)
   session.getAttribute("session_name");
14         String session_lvl = (String)
   session.getAttribute("session_lvl");
15         String session_ID_USR = (String)
   session.getAttribute("session_ID_USR");
16
17
18
19         /** ouverture de la base de donnée et on définit la requête */
20         // OPEN DATABASE
21         MysqlAccess Conn = new MysqlAccess();
22         Connection connexion = Conn.Open();
23         Class.forName("com.mysql.jdbc.Driver");
24         Statement instruction = connexion.createStatement();
25         Statement instructionsub = connexion.createStatement();
26         String queryString = "SELECT * FROM category";
27
28         /** on vérifie la suppression AVANT l'affichage !!! */
29         if (session_lvl!=null){
30             String DelSubCat = request.getParameter("DelSubCat");
31             String DelCat = request.getParameter("DelCat");
32             String FromIdCat = request.getParameter("FromIdCat");
33             /** on a récupéré les variables de session de suppression */
34             Boolean isadmindel = false;
35             /** on définit qu'il n'est pas admin et on verifie qu'il l'est dans la
   bdd ! */
36             isadmindel = Conn.CheckAdminSubcat(connexion,
   session_ID_USR, FromIdCat);
37
38             /** si il est level 1 ou 2 alors on peut entamer la procédure de
   suppression pour autant qu'il soit admin de la catégorie ! */
39             if(session_lvl.equals("2") || (session_lvl.equals("1")
   && isadmindel) ) {
40                 out.println("Ajouter une nouvelle catégories : <a
```

```

href='AddCategory'><img src='../images/b_edit.png'></a><br><br>");
41         if(DelSubCat!="" && DelSubCat!=null) {
42             out.println("DelSubCat : "+DelSubCat+"");
43             Conn.DelSubCat(connexion, DelSubCat);
44         }
45         if(DelCat!="" && DelCat!=null) {
46             out.println("DelCat : "+DelCat+"");
47             Conn.DelCat(connexion, DelCat);
48         }
49     }
50 }
51
52 /** une fois qu'on a traité les suppression on peut afficher les
catégories et sous catégories restante et on affiche les options en
fonction de son niveau */
53
54     ResultSet rs = instruction.executeQuery(queryString);
55     while(rs.next()){
56         // On récupère la catégorie analysée :
57         String IDCAT = rs.getString("ID_CAT");
58
59         // On vérifie si le user est au grade admin pour la
catégorie
60         Boolean isadmin = false;
61         if(session_ID_USR!=null){
62             isadmin = Conn.CheckAdminSubcat(connexion,
session_ID_USR, IDCAT);
63         }
64
65         if (session_lvl!=null){
66             if(session_lvl.equals("1") ) {
67                 out.println("<h2> "+rs.getString("NOM_CAT")+ " </h2>
");
68             }
69             if(session_lvl.equals("2")) {
70                 out.println("<h2> "+rs.getString("NOM_CAT")+ " <a
href='SubCategory?subcat="+IDCAT+"'><img
src='../images/b_props.png'></a> <a
href='ListMain?DelCat="+IDCAT+"'><img src='../images/b_drop.png'></a>
</h2> ");
71             }
72         }
73         else {
74             out.println("<h2>
"+rs.getString("NOM_CAT")+ "</h2>");
75         }
76         ResultSet rssub = instructionsub.executeQuery("SELECT * FROM
category_sub where ID_CAT="+IDCAT+"");
77         while(rssub.next()){
78             String idsub =
rssub.getString("ID_SUB");

```

```

79         if (session_lvl!=null){
80
81             if(session_lvl.equals("1"))
82             {
83                 out.println("<a
href='ListNews?subcat="+idsub+"'>"+rssub.getString("TITLE_SUB")+ "</a>
");
84                 if(isadmin) {
85                     out.println("<a
href='AddNews?subcat="+idsub+"'><img src='../images/b_edit.png'></a> <a
href='ListMain?DelSubCat="+idsub+"&FromIdCat="+IDCAT+"'><img
src='../images/b_drop.png'></a>");
86                 }
87                 out.println("<br>");
88             }
89             if(session_lvl.equals("2"))
90             {
91                 out.println("<a
href='ListNews?subcat="+idsub+"'>"+rssub.getString("TITLE_SUB")+ "</a>
<a href='AddNews?subcat="+idsub+"'><img src='../images/b_edit.png'></a>
<a href='ListMain?DelSubCat="+idsub+"'><img
src='../images/b_drop.png'></a><br><br>");
92             }
93         }
94         else {
95             out.println("<a
href='ListNews?subcat="+idsub+"'>"+rssub.getString("TITLE_SUB")+ "</a><b
r><br>");
96         }
97         out.println("<br><br>");
98     }
99 }
100
101
102
103
104 }
105
106 /** on termine par l'affichage de la fin du graphisme */
107     String bottom = "";
108     if(session_lvl==null) {
109         bottom = getgraph.bottom("0");

```

```
110     } else {
111         bottom = getgraph.bottom(session_lvl);
112     }
113     out.println(bottom);
114
115     } catch (SQLException ex) {
116         Logger.getLogger(ListMain.class.getName()).log(Level.SEVERE, null,
117             ex);
118     } catch (ClassNotFoundException ex) {
119         Logger.getLogger(ListMain.class.getName()).log(Level.SEVERE, null,
120             ex);
121     } finally {
122         // out.close();
123
124
125     }
```

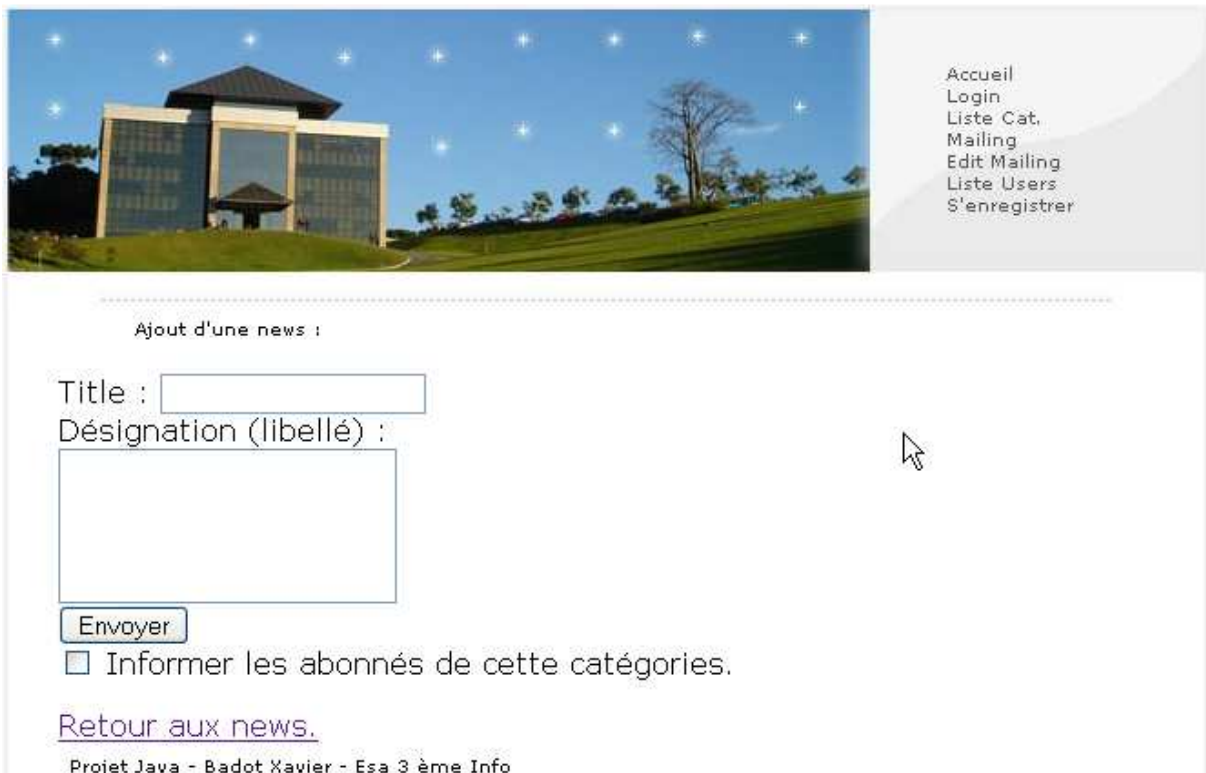
Ajout d'une news :

Informatique

[Main](#)  

Ecrire une news dans la sous - catégorie.

Il faut être [administrateur](#) de la catégorie ! ou superadmin (vl 2).



The screenshot shows a web application interface. At the top left is a banner image of a modern building. To the right is a navigation menu with the following items: Accueil, Login, Liste Cat., Mailing, Edit Mailing, Liste Users, and S'enregistrer. Below the banner, the page title is "Ajout d'une news :". The form contains a "Title :" field, a "Désignation (libellé) :" field, an "Envoyer" button, and a checkbox labeled "Informers les abonnés de cette catégories.". Below the form is a link "Retour aux news." and a footer "Projet Java - Badot Xavier - Esa 3 ème Info".

Prérequis :

Titre et désignation obligatoire.

La case à cocher permet d'envoyer un email à tous les utilisateurs abonnés pour cette catégories.

Résultat du formulaire :

Encodage de la news : Nouvelle de dimanche - statut terminé.

[Retour](#)

Le système nous retourne un message afin de retourner à la liste des news de la catégorie :

Résultat :



The screenshot shows a web application interface. At the top left is a banner image of a modern building with a glass facade and a central entrance, set against a blue sky with stars. To the right of the banner is a navigation menu with the following items: Accueil, Login, Liste Cat., Mailing, Edit Mailing, Liste Users, and S'enregistrer. Below the banner, there is a link "Retour aux catégories." followed by a small icon of a pencil. The main content area displays the word "bonjour" in a large, bold, black font. Below it, the text "bien le bonjour" is shown. Underneath that is a blue underlined link "Supprimer". The next line features the title "Nouvelle de dimanche" in a large, bold, black font. Below the title, the text "bonjour Ã tous" is displayed. At the bottom of this section is another blue underlined link "Supprimer".

Le système nous permet ici de supprimer les news :

- Nous sommes loggué en tant qu'admin de la catégorie ou superadmin.

AddNews.java

```
1  protected void processRequest(HttpServletRequest request,
   HttpServletResponse response)
2      throws ServletException, IOException {
3
4      response.setContentType("text/html;charset=UTF-8");
5      PrintWriter out = response.getWriter();
6      MysqlAccess Conn = new MysqlAccess();
7      Connection connexion = null;
8
9      try {
10     /** on crée l'objet graphique et on affiche le haut */
11     incgraph getgraph = new incgraph();
12     String top = getgraph.top();
13     out.println(top);
14
15     /** récupère les variables session et les variables du formulaire */
16     HttpSession session = request.getSession(true);
17     String session_lvl = (String)
   session.getAttribute("session_lvl");
18     String title = request.getParameter("title");
19     String libelle = request.getParameter("libelle");
20     String subcat = request.getParameter("subcat");
21     String choixmail = request.getParameter("choixmail");
22
23     /** va vérifier que le user est bien admin et non tombé sur cette page
   par hasard */
24     String session_ID_USR = (String)
   session.getAttribute("session_ID_USR");
25     String IDCAT=null;
26     Boolean isadmin = false;
27     if(session_ID_USR!=null){
28     connexion = Conn.Open();
29     IDCAT = Conn.FindCatFromSubCat(connexion, subcat);
30     isadmin = Conn.CheckAdminSubcat(connexion, session_ID_USR,
   IDCAT);
31     }
32     /** si il est admin ou superadmin on valide la news et on envoi la
   requete a la bdd */
33     if (isadmin==true ||session_lvl.equals("2")) {
34     if (title!=null && libelle!=null && subcat!=null) {
35     out.println("Encodage de la news : "+title+ " - statut
   terminé.<br><br><a href='ListNews?subcat="+subcat+"'> Retour </a>");
36     if(connexion!=null) {
37     Conn.AddNews(connexion, title, libelle, subcat);
38     }
   /** vérifie si il a demandé l'envoi des emails ; */
```



```

39     if(choixmail.equals("on")) {
40         SendMail go = new SendMail();
41         // A completer trouvé le nom de la catégorie
42         go.Send(""+IDCAT+"");
43         out.println("<br>Emails envoyés.<br>");
44     }
45
46
47     }
48     /** On propose l'ajout d'une nouvelle news */
49     out.println("<p>Ajout d'une news :</p>");
50     out.print("<form action=\"");
51     out.print("\ " ");
52     out.println("method=POST>");
53     out.println("Title : <input type=text size=20 name=title");
54     out.println("<br>");
55
56     out.println("Désignation (libellé) : <br><TEXTAREA rows='4'
name=libelle></TEXTAREA>");
57     out.println("<br>");
58     out.println("<input type=submit");
59     out.println("<br><INPUT type='checkbox' name='choixmail'>
Informers les abonnés de cette catégories.");
60
61     out.println("</form>");
62
63     out.println("<a href='ListNews?subcat="+subcat+"'>Retour aux
news.</a>");
64
65
66
67
68     }
69     else {
70     /** si l'utilisateur est tombé la par hasard on l'informe gentilleme
*/
71         out.println("Vous n'avez pas droit à l'administration !");
72     }
73
74     /** fin du graphisme
75     String bottom = "";
76     if(session_lvl==null) {
77         bottom = getgraph.bottom("0");
78     } else {
79         bottom = getgraph.bottom(session_lvl);
80     }
81     out.println(bottom);
82     } catch (AddressException ex) {

```

```

83     Logger.getLogger(AddNews.class.getName()).log(Level.SEVERE,
null, ex);
84     } catch (MessagingException ex) {
85         Logger.getLogger(AddNews.class.getName()).log(Level.SEVERE,
null, ex);
86     } finally {
87         out.close();
88     }
89 }

```

La gestion des comptes administrateurs :

Seul le superadmin a l'autorisation d'ajouter des administrateurs / administrateurs de catégories :

Lorsqu'il se loggue le menu admin lui est proposé en bas de page :

Administration : [AddAdminCategory](#) | [EditUsersCat](#)

Projet Java - Badot Xavier - Esa 3 ème Info - Design de CatWeb pour [Kits-Gratuits.net](#)

Servlet AddAdminCategory :

Ajouter : comme administrateur de la catégorie :

Le servlet va nous proposer la liste des administrateurs dans la première liste déroulante, et les différentes catégories dans la seconde :

Ajouter : comme administrateur de la catégorie :

Informatique

Dans le cas ou la selection est doublée une erreur nous est retournée :

Cet utilisateur est déjà administrateur de cette catégorie

Ajouter : comme administrateur de la catégorie :

```

public class AddAdminCategory extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and
     * POST methods.
     * @param request servlet request
     * @param response servlet response
     */
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession(true);
        String session_lvl = (String) session.getAttribute("session_lvl");
        /** on vérifie qui demande la requête */
        if (session_lvl != null) {
            if (session_lvl.equals("2")) {
                try {
                    /** création de l'objet graphique + affichage du top */
                    incgraph getgraph = new incgraph();
                    String top = getgraph.top();
                    out.println(top);
                    /** on ouvre la connexion */
                    MysqlAccess Conn = new MysqlAccess();
                    Connection connexion = Conn.Open();
                    Class.forName("com.mysql.jdbc.Driver");
                    /** récupère l'iduser et la catégorie ! */
                    String IDUSER = request.getParameter("IDUSER");
                    String cat = request.getParameter("cat");

                    /** si les variables sont nulles on demande l'ajout a la bdd */
                    if ((IDUSER != null && cat != null) && (IDUSER != "" && cat != "")) {
                        if (!Conn.CheckAdminSubcat(connexion, IDUSER, cat)) {
                            if (connexion != null) {
                                Conn.AddAdminCat(connexion, IDUSER, cat);
                            }
                            out.println("Encodage terminé");
                        } else {
                            out.println("Cet utilisateur est déjà administrateur de
                                cette catégorie");
                        }
                    }
                }
            }

            out.println("<FORM>");
        }
    }
}

```

```

/** On affiche le formulaire */
Statement instruction = connexion.createStatement();
String queryString = "SELECT * FROM users where
GROUP_USR='1'";
    out.println("<br><br> Ajouter : <SELECT name='IDUSER'
size='1'>");
/** administrateur */
    out.println("<OPTION>");
    ResultSet rs = instruction.executeQuery(queryString);
    while(rs.next()){
        String ID_USR = rs.getString("ID_USR");
        String NOM_USR = rs.getString("NOM_USR");
        String PRENOM_USR = rs.getString("PRENOM_USR");
        out.println("<OPTION value="+ID_USR+"> " + NOM_USR + " - "
+PRENOM_USR+ " ");
    }
    out.println("</SELECT>");

    Statement instructionsec = connexion.createStatement();
    String queryStringsec = "SELECT * FROM category";
    ResultSet rssec = instructionsec.executeQuery(queryStringsec);
    out.println(" comme administrateur de la catégorie : <SELECT
name='cat' size='1'>");
/** les catégories */
    out.println("<OPTION>");
    while(rssec.next()){
        String ID = rssec.getString("ID_CAT");
        String Nom = rssec.getString("NOM_CAT");
        out.println("");

        out.println("<OPTION value="+ID+">" + Nom + " ");

    }
    out.println("</SELECT>");
    out.println("<input type=submit>");
    out.println("</FORM>");

String bottom = "";
/** on ajout la fin du graphisme*/
if(session_lvl==null) {
bottom = getgraph.bottom("0");
} else {
bottom = getgraph.bottom(session_lvl);
}

```


La page mailing liste :



Cet écran sera disponible aux personnes connectées sur le site !

Toutes personnes peut s'inscrire à la mailing d'une catégorie, et sera ainsi tenu informé des mises à jour de celle-ci.

Si l'utilisateur est connecté :

Etre informé des mises à jour de la catégorie suivante :

Le système liste les différentes catégories du système à l'utilisateur, celui-ci peut choisir à quel catégorie s'abonner !

Dans le cas d'une erreur :


Vous semblez être déjà abonné a cette catégorie !
Etre informé des mises à jour de la catégorie suivante :

Le système vérifie que l'on n'est pas encore abonné. (Afin d'éviter les doublons).

Gestion de ses abonnements :

Servlet : /EditCatMailingUser


Liste des catégories auxquelles vous êtes abonnés.

Informatique 

Dans cette page nous retrouvons l'ensemble des catégories auxquelles nous sommes abonnés, et possibilité de supprimer l'abonnement.

La mailing liste code source : /Sendmail

Servlet /ListUsers :



The screenshot shows a web application interface. At the top left is a photograph of a modern building with a large glass facade and a central entrance, set against a blue sky with stars. To the right of the photo is a navigation menu with the following items: Accueil, Login, Liste Cat., Mailing, Edit Mailing, Liste Users, and S'enregistrer. Below the navigation menu is a horizontal dashed line. The main content area features a large heading 'Listings Users !' followed by a list of users: 0. superadmin - [supprimer](#), 1. admin - [supprimer](#), and 2. mcflyy - [supprimer](#). At the bottom of the content area, there is an 'Administration' section with links to [AddAdminCategory](#) and [EditUsersCat](#). The footer of the page reads 'Design de CatWeb pour [Kits-Gratuits.net](#)'.

Cette page permet de lister les utilisateurs et de supprimer ceux que l'on désire.

Cette option est uniquement proposé dans le cadre d'un administrateur de **level 2**.

Les différentes fonctions utilisés pour les requêtes à la base de donnée.

Class sendmail :

```
1 package website;
2
3 /*
4  * To change this template, choose Tools | Templates
5  * and open the template in the editor.
6  */
7 import java.sql.Connection;
8 import java.sql.ResultSet;
9 import java.sql.SQLException;
10 import java.sql.Statement;
11
12 import java.util.Properties;
13 import java.util.logging.Level;
14 import java.util.logging.Logger;
15 import javax.mail.Message;
16 import javax.mail.MessagingException;
17 import javax.mail.Transport;
18 import javax.mail.internet.AddressException;
19 import javax.mail.internet.InternetAddress;
20 import javax.mail.internet.MimeMessage;
21 /**
22  * Servlet envoi de la mailing de la catégorie souhaité.
23  *
24  * @author Badot Xavier <badotx@gmail.com>
25  */
26
27 public class SendMail {
28
29
30 public void send(String categorie) throws AddressException,
31 MessagingException
32 { /** déclaration des variables emailfrom email to*/
33     String emailFrom = "\"Esa-valves.com\"<mclain587@gmail.com>";
34     String emailTo = "mclain587@hotmail.com";
35 /** contenu de l'email */
36     String messageText = "Javamail";
37 /** sujet*/
38     String subject = "Mise à jour d'une catégorie sur le site Esa-
39 valves.com";
40     Properties props = System.getProperties();
41     /** SMTP du fournisseur d'accès */
42     props.put("mail.smtp.host", "smtp.tvcablenet.be");
43     // javax.mail.Session pour que ça fonctionne aussi avec
44     WebObjects
45     javax.mail.Session maSession =
46     javax.mail.Session.getDefaultInstance(props, null);
47     MimeMessage message = new MimeMessage(maSession);
48     message.setFrom(new InternetAddress(emailFrom));
```



```

44     message.addRecipient(Message.RecipientType.TO, new
InternetAddress(emailTo));
45     //message.addRecipient(Message.RecipientType.CC, new
InternetAddress(emailCc));
46     message.setSubject(subject);
47     message.setText(messageText);
48
49
50     try
51     {
52
53
54         /** ouverture de la connection */
55         MysqlAccess Conn = new MysqlAccess();
56         Connection connexion = Conn.Open();
57         Class.forName("com.mysql.jdbc.Driver");
58
59         /** début de la selection pour la mailing liste */
60         Statement instruction = connexion.createStatement();
61         Statement instructionsec = connexion.createStatement();
62         ResultSet resultat = instruction.executeQuery("SELECT
ID_USR FROM mailing where ID_CAT="+Categorie+"");
63
64         while (resultat.next()) {
65             String ID_USR = resultat.getString("ID_USR");
66             System.out.println("<h1> Users : "+ID_USR+"</h1>
<p>");
67
68
69             ResultSet result = instructionsec.executeQuery("SELECT
* FROM users where ID_USR="+ID_USR+"");
70             /** on selectionne les infos de l'utilisateur abonné pour avoir son
email */
71             System.out.println("SELECT * FROM users where
ID_USR="+ID_USR+"");
72
73             while (result.next()) {
74                 String emailgorecp = result.getString("EMAIL_USR");
75                 System.out.println(""+emailgorecp+"");
76             /** on l'ajoute au récipient d'envoi.*/
77             message.addRecipient(Message.RecipientType.TO, new
InternetAddress(emailgorecp));
78
79         }
80
81     }
82
83
84
85

```

```
86
87
88  /** on envoi*/
89      Transport.send(message);
90
91
92
93
94  }
95      catch (SQLException ex) {
96  Logger.getLogger(SendMail.class.getName()).log(Level.SEVERE, null,
ex);
97  }      catch (ClassNotFoundException ex) {
98  Logger.getLogger(SendMail.class.getName()).log(Level.SEVERE, null,
ex);
99  }
100
101
102
103  }
104 }
```

Ensemble des fonctions utilisées : (mysqlaccess.java).

```
package website;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Mcflyyy
 */

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;

public class MysqlAccess {

    public Connection Open () {
        Connection connexion = null;
        try {
            /** fonction créant la connection ( login, password, server ) */
            Class.forName("com.mysql.jdbc.Driver");

            try {
                connexion =
                DriverManager.getConnection("jdbc:mysql://localhost/valves_esa", "root",
                "");
                Statement instruction = connexion.createStatement();
            } catch (SQLException ex) {

                Logger.getLogger(MysqlAccess.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
}
```

```

        } catch (ClassNotFoundException ex) {
            Logger.getLogger(MySqlConnection.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return connexion;
    }

    /** fermeture de la connexion */
    public void CloseCon(Connection connexion) {
        try {
            connexion.close();
        } catch (SQLException ex) {
            Logger.getLogger(MySqlConnection.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }

    public void EncodeClients (String FirstName,String LastName,String
Adresse,String Cdp,String Tel,String Gsm,String Mail,String Login,String
Password,String Grp,Connection connexion) {
        try {
            Statement instruction = connexion.createStatement();
            /** On récupère toutes les données transmises et on encode
l'utilisateur dans la base de donnée. */
            int resultat = instruction.executeUpdate("Insert INTO users
(NOM_USR,PRENOM_USR,ADR_USR,CDP_USR,TEL_USR,GSM_USR,MAIL,LOGIN_USR,PASS_USR
,ENABLE_USR,GROUP_USR) values
('"+LastName+"','"+FirstName+"','"+Adresse+"','"+Cdp+"','"+Tel+"','"+Gsm+"',
'"+Mail+"','"+Login+"','"+Password+"','1','0')");
        } catch (SQLException ex) {
            Logger.getLogger(MySqlConnection.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }

    /** affichage tous les utilisateurs */
    public void ListingClient (Connection connexion) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Statement instruction = connexion.createStatement();
            ResultSet resultat = instruction.executeQuery("SELECT * FROM
users");

            while (resultat.next()) {
                String Nom = resultat.getString("NOM_USR");
                System.out.println("<h1> Users : "+Nom+"</h1> <p>");
            }
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(MySqlConnection.class.getName()).log(Level.SEVERE,

```

```

null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
        }

    }

    /** vérification que le login et le mdp est correct */
    public boolean GetSession (String Login,String Password,Connection
connexion) {
        System.out.println("Vérification Password - For login -> "+Login+"
with Password : " + Password + "");
        String PasswdOk = "";
        try {
            Class.forName("com.mysql.jdbc.Driver");

            Statement instruction = connexion.createStatement();
            ResultSet resultat = instruction.executeQuery("SELECT
LOGIN_USR,PASS_USR FROM users where LOGIN_USR='"+ Login +"'");
            while (resultat.next()) {
                PasswdOk = resultat.getString("PASS_USR");
                System.out.println("Passwd requis : "+PasswdOk+"");
            }

        }
        catch (ClassNotFoundException ex) {
            Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
        }
        catch (SQLException ex) {
            Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
        }

        if(PasswdOk.equals(Password)) {
            System.out.println("<h1> Utilisateur authentifié </h1> <p>");
            return true;
        }
        else {
            System.out.println("<h1> Echec de l'authentification </h1>
<p>");
            return false;
        }
    }
}

```

```

/** ajout d'une catégorie */
    public void AddCategory(Connection connexion,String name,String
libelle) {
        try {
            Dates a = new Dates();
            String dateStandard = a.date();
            Statement instruction = connexion.createStatement();
            System.out.println("Insert INTO category
(NOM_CAT,DESIGN_CAT,DATE_CAT) values ('" + name + "','" + libelle + "','" +
dateStandard +"'");
            instruction.executeUpdate("Insert INTO category
(NOM_CAT,DESIGN_CAT,DATE_CAT) values ('" + name + "','" + libelle + "','" +
dateStandard +"'");

        } catch (SQLException ex) {
            Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
}

```

```

/** ajout d'une sous catégorie */
    public void AddSubCategory(Connection connexion,String name,String
libelle, String subcatget) {
        try {
            Dates a = new Dates();
            String dateStandard = a.date();
            Statement instruction = connexion.createStatement();
            System.out.println("Insert INTO category_sub
(TITLE_SUB,MSG_SUB,DATE_SUB) values ('" + name + "','" + libelle + "','" +
dateStandard +"'");
            instruction.executeUpdate("Insert INTO category_sub
(TITLE_SUB,MSG_SUB,DATE_SUB,ID_CAT) values ('" + name + "','" + libelle +
 "','" + dateStandard + "','" + subcatget +"'");

        } catch (SQLException ex) {
            Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
}

```

```

/** ajout d'un administrateur à une catégorie */

    public void AddAdminCat (Connection connexion,String IDUSER,String
cat) {
        try {
            Statement instruction = connexion.createStatement();
            System.out.println("Insert INTO administration (ID_USR,ID_CAT)
values ('" + IDUSER + "','" + cat + "'");
            instruction.executeUpdate("Insert INTO administration
(ID_USR,ID_CAT) values ('" + IDUSER + "','" + cat + "'");
        }
    }
}

```

```

    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

/** ajout d'une catégorie à ma mailing liste */
public void AddCatMailing (Connection connexion,String IDUSER,String
cat) {
    try {
        Statement instruction = connexion.createStatement();
        System.out.println("Insert INTO mailing (ID_USR,ID_CAT) values
('" + IDUSER + "','" + cat + "')");
        instruction.executeUpdate("Insert INTO mailing (ID_USR,ID_CAT)
values ('" + IDUSER + "','" + cat + "')");

    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

/** vérifier le level de l'utilisateur */
public String CheckLevelAdmin (Connection connexion,String Login) {
    String LevelUsr = "0";
    try {

        Statement instruction = connexion.createStatement();
        ResultSet resultat = instruction.executeQuery("SELECT GROUP_USR
FROM users where LOGIN_USR='" + Login + "'");
        resultat.next();
        LevelUsr = resultat.getString("GROUP_USR");

    }
    catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
    finally {
        return LevelUsr;
    }
}

/** ajouter une news à une sous catégorie */
public void AddNews(Connection connexion,String title,String
libelle,String subcat) {
    try {
        Dates a = new Dates();
        String dateStandard = a.date();
        Statement instruction = connexion.createStatement();
        System.out.println("Insert INTO category_sub

```

```

(TITLE_NEWS,MSG_NEWS,DATE_NEWS,ID_SUB) values ('" + title + "','" + libelle
+ "','" + dateStandard + "','" + subcat + "')");
        instruction.executeUpdate("Insert INTO news
(TITLE_NEWS,MSG_NEWS,DATE_NEWS,ID_SUB) values ('" + title + "','" + libelle
+ "','" + dateStandard + "','" + subcat + "')");

    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

/** supprimer une news */
public void DelNews(Connection connexion,String ID) {
    try {
        Statement instruction = connexion.createStatement();
        System.out.println("DELETE FROM news WHERE id_news="+ID+"");
        instruction.executeUpdate("DELETE FROM news WHERE
id_news="+ID+"");

    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

/** suppression d'un utilisateur */
public void DelUsers (Connection connexion,String ID) {
    try {
        Statement instruction = connexion.createStatement();
        System.out.println("DELETE FROM users WHERE ID_USR="+ID+"");
        instruction.executeUpdate("DELETE users news WHERE
ID_USR="+ID+"");

    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

/** trouver une catégorie a partir d'une sous catégorie */
public String FindCatFromSubCat(Connection connexion,String ID_SUB) {
    String ID_CAT = null;
    try {

        System.out.println("SELECT ID_CAT from category_sub where
ID_SUB='"+ID_SUB+"'");
        Statement instruction = connexion.createStatement();
        ResultSet resultat = instruction.executeQuery("SELECT ID_CAT
from category_sub where ID_SUB='"+ID_SUB+"'");
        resultat.next();
    }
}

```



```

        ID_CAT = resultat.getString("ID_CAT");

    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
    finally{
        return ID_CAT;
    }
}

/** récupérer l'id de l'utilisateur */
public String Get_ID_USER (Connection connexion,String Log) {
    String ID=null;
    try {
        Boolean valid = false;
        System.out.println("SELECT ID_USR from users where
LOGIN_USR='"+Log+"'");
        Statement instruction = connexion.createStatement();
        ResultSet resultat = instruction.executeQuery("SELECT ID_USR
from users where LOGIN_USR='"+Log+"'");
        while (resultat.next()) {
            ID = resultat.getString("ID_USR");
        }
    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return ID;
}

/** vérifier que l'utilisateur est bien admin de la catégorie */
public Boolean CheckAdminSubcat(Connection connexion,String
ID_USR,String Cat) {
    Boolean valid = false;
    String USERIDOK = null;
    try {
        System.out.println("SELECT ID_USR from administration where
ID_CAT='"+Cat+"'");
        Statement instruction = connexion.createStatement();
        ResultSet resultat = instruction.executeQuery("SELECT ID_USR
from administration where ID_CAT='"+Cat+"'");
        while (resultat.next()) {
            USERIDOK = resultat.getString("ID_USR");
            System.out.println("Valeur : "+USERIDOK+" <br>");
            if (USERIDOK!=null) {
                valid=true;
            }
            else {
                valid=false;
            }
        }
    }
}

```

```

    }

} catch (SQLException ex) {
    Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
}
finally {
    System.out.println(""+valid+"");
    return valid;
}
}

/** vérifier si l'utilisateur est déjà abonné */
public Boolean CheckMailCat(Connection connexion,String ID_USR,String
Cat) {
    Boolean valid = false;
    String USERIDOK = null;
    try {

        System.out.println("SELECT ID_MAIL from mailing where
ID_CAT='"+Cat+"' and ID_USR='"+ID_USR+"'");
        Statement instruction = connexion.createStatement();
        ResultSet resultat = instruction.executeQuery("SELECT ID_MAIL
from mailing where ID_CAT='"+Cat+"' and ID_USR='"+ID_USR+"'");
        while (resultat.next()) {
            USERIDOK = resultat.getString("ID_MAIL");
            System.out.println("Valeur : "+USERIDOK+" <br>");
            if (USERIDOK!=null) {
                valid=true;
            }
            else {
                valid=false;
            }
        }

    }

} catch (SQLException ex) {
    Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
}
finally {
    System.out.println(""+valid+" - non trouvé ...");
    return valid;
}
}

/** supprimer un administrateur d'une catégorie */
public void DelUsersFromCatAdmin(Connection connexion,String ID) {
    try {
        Statement instruction = connexion.createStatement();

```

```

        System.out.println("DELETE FROM administration WHERE
ID_ADMIN="+ID+"");
        instruction.executeUpdate("DELETE FROM administration WHERE
ID_ADMIN="+ID+"");

    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public void DelCat(Connection connexion,String ID) {
    try {
        Statement instruction = connexion.createStatement();
        System.out.println("DELETE FROM category WHERE
ID_CAT="+ID+"");
        instruction.executeUpdate("DELETE FROM category WHERE
ID_CAT="+ID+"");

        System.out.println("DELETE FROM category_sub WHERE
ID_CAT="+ID+"");
        instruction.executeUpdate("DELETE FROM category_sub WHERE
ID_CAT="+ID+"");

        //TODO delete news from subcat del _

    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

/** supprimer sous catégorie */
public void DelSubCat(Connection connexion,String ID) {
    try {
        Statement instruction = connexion.createStatement();
        System.out.println("DELETE FROM category_sub WHERE
ID_SUB="+ID+"");
        instruction.executeUpdate("DELETE FROM category_sub WHERE
ID_SUB="+ID+"");

    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

/** supprimer un utilisateur de la mailing */
public void DelCatFromMailing(Connection connexion,String ID) {
    try {
        Statement instruction = connexion.createStatement();
        System.out.println("DELETE FROM mailing WHERE

```

```

ID_MAIL="+ID+"");
        instruction.executeUpdate("DELETE FROM mailing WHERE
ID_MAIL="+ID+"");
    } catch (SQLException ex) {
        Logger.getLogger(MySqlAccess.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
}
}

```

Conclusion :



Toutes requêtes ou mises à jour de la base de données se fait via un objet bien spécifique.

L'ensemble des fonctions sont tenus dans un fichier class : MySqlAccess.java

Nous aurions pu imaginer faire de même a l'intérieur des servlets afin de clarifier le code.

Définition des servlets (web.xml) :



Problèmes rencontrés :

La définition des servlets ce fait via le fichier web.xml et est assuré de manière automatique dans netbeans.

Cependant j'ai eu quelques soucis avec la génération automatique, certainement déclaration n'était apparament pas correcte pour lui. Je les ai donc modifiées à la main.

La partie la plus importante pour nous était donc :

```

<servlet>
    <servlet-name>Accueil</servlet-name>
    <servlet-class>website.Accueil</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>Accueil</servlet-name>
    <url-pattern>/Accueil</url-pattern>
</servlet-mapping>

```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

Copyright 2004 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

```
-->
```

```
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
```

```
<display-name>Servlet 2.4 Examples</display-name>
```

```
<description>
```

Servlet 2.4 Examples.

```
</description>
```

```
<!-- Define servlet-mapped and path-mapped example filters -->
```

```
<filter>
```

```
<filter-name>Servlet Mapped Filter</filter-name>
```

```
<filter-class>filters.ExampleFilter</filter-class>
```

```
<init-param>
```

```
<param-name>attribute</param-name>
```

```
<param-value>filters.ExampleFilter.SERVLET_MAPPED</param-value>
```

```
</init-param>
```

```
</filter>
```

```
<filter>
```

```
<filter-name>Path Mapped Filter</filter-name>
```

```
<filter-class>filters.ExampleFilter</filter-class>
```

```
<init-param>
```

```
<param-name>attribute</param-name>
```

```
<param-value>filters.ExampleFilter.PATH_MAPPED</param-value>
```

```
</init-param>
```

```
</filter>
```

```
<filter>
```

```
<filter-name>Request Dumper Filter</filter-name>
```

```
<filter-class>filters.RequestDumperFilter</filter-class>
```

```
</filter>
```

```
<!-- Example filter to set character encoding on each request -->
```

```
<filter>
```

```
<filter-name>Set Character Encoding</filter-name>
```

```
<filter-class>filters.SetCharacterEncodingFilter</filter-class>
```

```
<init-param>
```

```
<param-name>encoding</param-name>
```

```
<param-value>EUC_JP</param-value>
```

```
</init-param>
```

```
</filter>
```

```
<filter>
  <filter-name>Compression Filter</filter-name>
  <filter-class>compressionFilters.CompressionFilter</filter-class>

  <init-param>
    <param-name>compressionThreshold</param-name>
    <param-value>10</param-value>
  </init-param>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
</filter>
```

```
<!-- Define filter mappings for the defined filters -->
```

```
<filter-mapping>
  <filter-name>Servlet Mapped Filter</filter-name>
  <servlet-name>invoker</servlet-name>
</filter-mapping>
<filter-mapping>
  <filter-name>Path Mapped Filter</filter-name>
  <url-pattern>/servlet/*</url-pattern>
</filter-mapping>
```

```
<jsp-config>
  <jsp-property-group>
```

```
<url-pattern>*.jsp</url-pattern>

<page-encoding>UTF-8</page-encoding>

</jsp-property-group>

</jsp-config>
```

```
<!-- Example filter mapping to apply the "Set Character Encoding" filter
```

```
to *all* requests processed by this web application -->
```

```
<!--
```

```
<filter-mapping>
```

```
<filter-name>Set Character Encoding</filter-name>
```

```
<url-pattern>/*</url-pattern>
```

```
</filter-mapping>
```

```
-->
```

```
<!--
```

```
<filter-mapping>
```

```
<filter-name>Compression Filter</filter-name>
```

```
<url-pattern>/CompressionTest</url-pattern>
```

```
</filter-mapping>
```

```
-->
```

```
<!--
```

```
<filter-mapping>
```

```
<filter-name>Request Dumper Filter</filter-name>
```

```
<url-pattern>/*</url-pattern>
```

```
</filter-mapping>
```

```
-->
```

```
<!-- Define example application events listeners -->
```



```
<listener>
  <listener-class>listeners.ContextListener</listener-class>
```

```
</listener>
```

```
<listener>
  <listener-class>listeners.SessionListener</listener-class>
```

```
</listener>
```

```
<!-- Define servlets that are included in the example application -->
```

```
<servlet>
  <servlet-name>AddNews</servlet-name>
  <servlet-class>website.AddNews</servlet-class>
```

```
</servlet>
```

```
<servlet>
  <servlet-name>CompressionFilterTestServlet</servlet-name>
  <servlet-class>compressionFilters.CompressionFilterTestServlet</servlet-class>
```

```
</servlet>
```

```
<servlet>
  <servlet-name>HelloWorldExample</servlet-name>
  <servlet-class>website.HelloWorldExample</servlet-class>
```

```
</servlet>
```

```
<servlet>
  <servlet-name>RequestInfoExample</servlet-name>
  <servlet-class>website.RequestInfoExample</servlet-class>
```

```
</servlet>
```

```
<servlet>
  <servlet-name>RequestHeaderExample</servlet-name>
  <servlet-class>website.RequestHeaderExample</servlet-class>
```

```
</servlet>
```

```
<servlet>
```

```
<servlet-name>RequestParamExample</servlet-name>

<servlet-class>website.RequestParamExample</servlet-class>

</servlet>

<servlet>

  <servlet-name>CookieExample</servlet-name>

  <servlet-class>website.CookieExample</servlet-class>

</servlet>

<servlet>

  <servlet-name>SessionExample</servlet-name>

  <servlet-class>website.SessionExample</servlet-class>

</servlet>

<servlet>

  <servlet-name>Session</servlet-name>

  <servlet-class>website.Session</servlet-class>

</servlet>

<servlet>

  <servlet-name>ListMain</servlet-name>

  <servlet-class>website.ListMain</servlet-class>

</servlet>

<servlet>

  <servlet-name>ListNews</servlet-name>

  <servlet-class>website.ListNews</servlet-class>

</servlet>

<servlet>

  <servlet-name>AddAdminCategory</servlet-name>

  <servlet-class>website.AddAdminCategory</servlet-class>

</servlet>

<servlet>
```

```
<servlet-name>ListUsers</servlet-name>

<servlet-class>website.ListUsers</servlet-class>

</servlet>

<servlet>

<servlet-name>SubCategory</servlet-name>

<servlet-class>website.SubCategory</servlet-class>

</servlet>

<servlet>

<servlet-name>NewUser</servlet-name>

<servlet-class>website.NewUser</servlet-class>

</servlet>

<servlet>

<servlet-name>AddCatMailing</servlet-name>

<servlet-class>website.AddCatMailing</servlet-class>

</servlet>

<servlet>

<servlet-name>EditCatMailingUser</servlet-name>

<servlet-class>website.EditCatMailingUser</servlet-class>

</servlet>

<servlet>

<servlet-name>EditUsersCat</servlet-name>

<servlet-class>website.EditUsersCat</servlet-class>

</servlet>

<servlet>

<servlet-name>AddCategory</servlet-name>

<servlet-class>website.AddCategory</servlet-class>

</servlet>

<servlet>

<servlet-name>Accueil</servlet-name>
```

```
<servlet-class>website.Accueil</servlet-class>

</servlet>

<servlet-mapping>

  <servlet-name>Accueil</servlet-name>

  <url-pattern>/Accueil</url-pattern>

</servlet-mapping>

<servlet-mapping>

  <servlet-name>EditUsersCat</servlet-name>

  <url-pattern>/EditUsersCat</url-pattern>

</servlet-mapping>

<servlet-mapping>

  <servlet-name>EditCatMailingUser</servlet-name>

  <url-pattern>/EditCatMailingUser</url-pattern>

</servlet-mapping>

<servlet-mapping>

  <servlet-name>AddNews</servlet-name>

  <url-pattern>/AddNews</url-pattern>

</servlet-mapping>

<servlet-mapping>

  <servlet-name>SubCategory</servlet-name>

  <url-pattern>/SubCategory</url-pattern>

</servlet-mapping>

<servlet-mapping>

  <servlet-name>AddCategory</servlet-name>

  <url-pattern>/AddCategory</url-pattern>

</servlet-mapping>

<servlet-mapping>

  <servlet-name>NewUser</servlet-name>

  <url-pattern>/NewUser</url-pattern>
```

```
</servlet-mapping>

<servlet-mapping>
  <servlet-name>EditCatMailingUser</servlet-name>
  <url-pattern>/EditCatMailingUser</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>AddCatMailing</servlet-name>
  <url-pattern>/AddCatMailing</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>ListMain</servlet-name>
  <url-pattern>/ListMain</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>ListNews</servlet-name>
  <url-pattern>/ListNews</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>ListUsers</servlet-name>
  <url-pattern>/ListUsers</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>Session</servlet-name>
  <url-pattern>/Session</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>AddAdminCategory</servlet-name>
  <url-pattern>/AddAdminCategory</url-pattern>
</servlet-mapping>
```

```
<servlet-mapping>
  <servlet-name>CompressionFilterTestServlet</servlet-name>
  <url-pattern>/CompressionTest</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>HelloWorldExample</servlet-name>
  <url-pattern>/servlet/HelloWorldExample</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>RequestInfoExample</servlet-name>
  <url-pattern>/servlet/RequestInfoExample/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>RequestHeaderExample</servlet-name>
  <url-pattern>/servlet/RequestHeaderExample</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>RequestParamExample</servlet-name>
  <url-pattern>/servlet/RequestParamExample</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>CookieExample</servlet-name>
  <url-pattern>/servlet/CookieExample</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>SessionExample</servlet-name>
  <url-pattern>/servlet/SessionExample</url-pattern>
</servlet-mapping>
```

```
<security-constraint>

<display-name>Example Security Constraint</display-name>

<web-resource-collection>

  <web-resource-name>Protected Area</web-resource-name>

  <!-- Define the context-relative URL(s) to be protected -->

  <url-pattern>/jsp/security/protected/*</url-pattern>

  <!-- If you list http methods, only those methods are protected -->

  <http-method>DELETE</http-method>

  <http-method>GET</http-method>

  <http-method>POST</http-method>

  <http-method>PUT</http-method>

</web-resource-collection>

<auth-constraint>

  <!-- Anyone with one of the listed roles may access this area -->

  <role-name>tomcat</role-name>

  <role-name>role1</role-name>

</auth-constraint>

</security-constraint>

<!-- Default login configuration uses form-based authentication -->

<login-config>

  <auth-method>FORM</auth-method>

  <realm-name>Example Form-Based Authentication Area</realm-name>

  <form-login-config>

    <form-login-page>/jsp/security/protected/login.jsp</form-login-page>

    <form-error-page>/jsp/security/protected/error.jsp</form-error-page>

  </form-login-config>

</login-config>
```

<!-- Security roles referenced by this web application -->

<security-role>

<role-name>role1</role-name>

</security-role>

<security-role>

<role-name>tomcat</role-name>

</security-role>

<!-- Environment entry examples -->

<!--env-entry>

<env-entry-description>

The maximum number of tax exemptions allowed to be set.

</env-entry-description>

<env-entry-name>maxExemptions</env-entry-name>

<env-entry-value>15</env-entry-value>

<env-entry-type>java.lang.Integer</env-entry-type>

</env-entry-->

<env-entry>

<env-entry-name>minExemptions</env-entry-name>

<env-entry-type>java.lang.Integer</env-entry-type>

<env-entry-value>1</env-entry-value>

</env-entry>

<env-entry>

<env-entry-name>foo/name1</env-entry-name>

<env-entry-type>java.lang.String</env-entry-type>

<env-entry-value>value1</env-entry-value>

</env-entry>

<env-entry>

<env-entry-name>foo/bar/name2</env-entry-name>

<env-entry-type>java.lang.Boolean</env-entry-type>

<env-entry-value>true</env-entry-value>

</env-entry>

<env-entry>

<env-entry-name>name3</env-entry-name>

<env-entry-type>java.lang.Integer</env-entry-type>

<env-entry-value>1</env-entry-value>

</env-entry>

<env-entry>

<env-entry-name>foo/name4</env-entry-name>

<env-entry-type>java.lang.Integer</env-entry-type>

<env-entry-value>10</env-entry-value>

</env-entry>

</web-app>