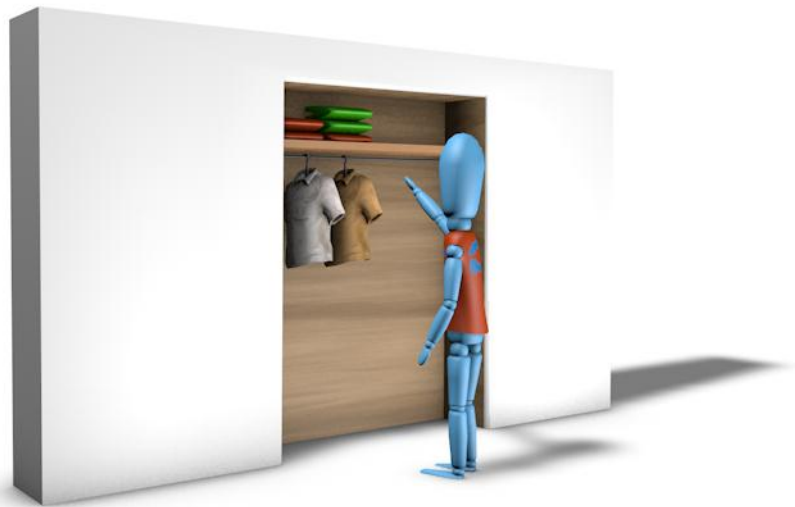


**Delphine Gloria**  
**Justin Le Guennec**



# **E-Wardrobe**

**Réalisation d'une penderie augmentée  
communicante**

**[e-wardrobe.polytech-lille.net](http://e-wardrobe.polytech-lille.net)**

**Thomas Vantroys / Alexandre Boé**

**Année 2011/2012**

## Table des matières

I.	INTRODUCTION.....	2
II.	DECOUVERTE DU PROJET .....	3
A.	DEBUT DU PROJET.....	3
1.	LE CAHIER DES CHARGES .....	3
2.	LA SOLUTION RFID .....	3
B.	LE SCHEMA DE PRINCIPE ET LA BASE DE DONNEES .....	4
C.	LA REPARTITION DES TACHES .....	5
III.	LECTURE DES TAGS .....	6
A.	OBJECTIF DE L'APPLICATION DE LECTURE .....	6
B.	L'APPLICATION .....	6
1.	LECTURE D'UN TAG .....	6
2.	MISE A JOUR DE LA BASE DE DONNEES .....	7
3.	LES DIFFERENTS PROBLEMES RENCONTRES.....	7
V.	APPLICATION WEB .....	9
A.	OBJECTIF DE L'APPLICATION.....	9
B.	CREATION DE L'APPLICATION .....	9
1.	LA PAGE : CHOIX DU VETEMENT .....	9
2.	L'INDEX.....	10
3.	LA PAGE : ACCUEIL .....	10
4.	LA PAGE : VOTRE PENDERIE .....	10
5.	LA PAGE : GESTION PENDERIE .....	11
6.	LA PAGE : AJOUTER UN VETEMENT .....	11
7.	LA PAGE : SUPPRIMER UN VETEMENT.....	11
C.	LES DIFFICULTES DE L'APPLICATION .....	12
VI.	CONCLUSION.....	13
VII.	ANNEXES.....	14

# I. INTRODUCTION

Afin d'améliorer nos différentes connaissances, nous devons réaliser pendant le semestre 8 un projet tuteuré. Ce dernier va nous permettre d'appliquer les différentes notions que nous avons vues en cours et également en découvrir de nouvelles. Etant tous les deux dans la filière « systèmes communicants », nous avons choisi un projet portant principalement sur les notions d'informatiques. En effet ce projet correspond davantage à nos envies professionnelles. Nous allons également apprendre à créer entièrement un site internet et découvrir les langages tels que le HTML, le PHP et le CSS.

De plus, ce projet nous permet d'utiliser la technologie tactile que nous utilisons chaque jour mais aussi de découvrir la technologie RFID. Pour vous expliquer le travail que nous effectués, nous avons choisi de le séparer en quatre parties. Nous commencerons avec la découverte du projet (cahier des charges, répartition des tâches ...) puis nous vous expliquerons notre méthode pour lire les différents tags que nous passons devant la Phidgets. Ensuite dans une troisième partie, nous vous expliquerons comment fonctionne notre application Web. Enfin, dans la quatrième partie nous vous donnerons les différents résultats que nous avons obtenus.

# II. DECOUVERTE DU PROJET

## A. DEBUT DU PROJET

### 1. LE CAHIER DES CHARGES

Voici le cahier des charges que nous avons récupéré au début du projet sur l'intranet de l'IMA :

**Objectif** Concevoir et développer une penderie qui identifie automatiquement les vêtements présents (utilisation de la technologie RFIF/NFC) et réalise des propositions d'habits en fonction d'éléments extérieurs comme la météo.

**Description** Pour réaliser la e-Wardrobe, la première étape consistera à étudier la technologie RFID et déterminer quelle est la fréquence radio (125 KHz, 13,56 MHz ou UHF) utilisée pour scanner en continu les différents vêtements présents dans la penderie. Vous réaliserez ensuite la carte électronique. L'étape suivante consistera à faire communiquer le système RFID avec un système FoxBoard. Ce dernier permettra la communication avec différentes machines offrant des services (météo, affichage sur grand écran, ...). Vous développerez ensuite les services nécessaires pour la recommandation d'habits.

Nous pouvons en conclure que le but de notre projet est de créer une penderie intelligente : une E-Wardrobe. Elle sera capable de référencer tous les vêtements qu'elle contient et de savoir ceux qui sont présents et ceux qui manquent. De plus, grâce à une application implantée sur un écran tactile, elle conseillera l'utilisateur sur la meilleure tenue à porter en fonction de la météo et de la coordination des couleurs. Elle sera par exemple capable de lui dire si deux vêtements ne vont pas ensemble ou si la tenue est trop légère par rapport au temps qu'il fait dehors. Pour référencer les différents habits, nous allons utiliser le protocole RFID.

### 2. LA SOLUTION RFID

Au début de notre projet, nous avons cherché avec les professeurs d'électronique la meilleure façon pour détecter les tags que nous fixerons sur nos vêtements. Nous avons tout d'abord travaillé sur l'option d'utiliser un capteur RFID du type Q5M005 du constructeur Lextronic. Cette solution fonctionne avec le système Arduino.

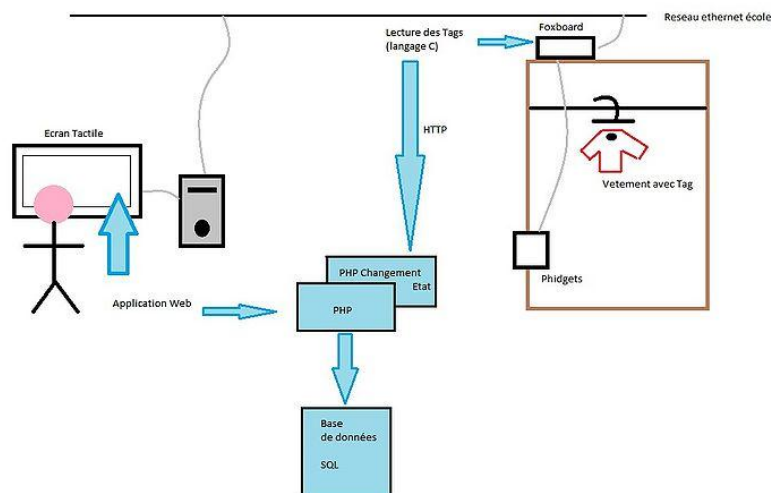


Par conséquent, nous avons cherché une librairie RFID compatible avec l'Arduino. Nous en avons trouvé une mais nous nous sommes rapidement rendus compte qu'elle ne fonctionnait pas avec la liaison RS232. Nous avons continué à en chercher une et nous avons même pensé à l'élaborer nous-même grâce à la datasheet du composant Q5M005. Cette solution, nous a causé beaucoup de problèmes et afin de pouvoir avancer dans le projet, nous avons choisi de travailler à la carte Phidgets qui est plus facilement utilisable.

Nous sommes donc partis sur cette solution et nous avons de ce fait installé toutes les librairies nécessaires à son utilisation afin de pouvoir commencer à travailler sur le programme de détection des tags (il sera expliqué dans la partie suivante).

## B. LE SCHEMA DE PRINCIPE ET LA BASE DE DONNEES

Au début du projet, nous avons réalisé un schéma de principe qui nous permet de définir les différents blocs de notre projet. Il nous permet de définir les différents langages que nous allons utiliser et les différentes communications qu'il est nécessaires de réaliser.



Pour gérer les données nécessaires à la création de notre penderie intelligente, nous avons choisi de créer une base de données sous phpMyAdmin. Dans cette base nous allons créer deux tables.

La première : la table vêtement, nous permettra de stocké tous les vêtements qu'ils soient présents ou non dans la penderie. Nous allons y saisir plusieurs champs contenant les différentes informations associées à chaque vêtement :

- ✚ Le tag du vêtement
- ✚ La description du vêtement
- ✚ Les températures minimale et maximale supportées par le vêtement
- ✚ La présence du vêtement dans la penderie (1 : présent, 0 : absent)

La deuxième table : tag temporaire, comprend deux champs :

- ✚ Le dernier tag lu par la Phidgets
- ✚ La lecture ou non d'un vêtement (1 : lecture en cours, 0 : pas de lecture)

Cette table nous sert de communication entre le programme C et notre application WEB. En effet, à chaque fois qu'un vêtement passe devant la Phidgets nous mettons à jour le champ `tag_vetement_en_cours` de cette table et nous mettons à 1 le champ `lecture_vetement`. De son côté, l'application regarde toutes les secondes si le champ `lecture_vetement` passe à 1 et si c'est le cas agit en conséquence en affichant par exemple la photographie associée à cet habit.

## C. LA REPARTITION DES TACHES

Pour optimiser l'avancement de notre projet, nous avons décidé de nous répartir les différentes tâches à effectuer dans notre projet. Nous nous les sommes répartis en fonction de nos compétences personnelles et de nos envies.

Par conséquent, voici notre répartition qui est en rapport avec le schéma de principe vu précédemment :

Delphine : création de la base de données et de l'application Web.

Justin : Gestion de la lecture des Tags et de la communication avec la Foxboard.  
Gestion de la page php changement d'état.

# III. LECTURE DES TAGS

## A. OBJECTIF DE L'APPLICATION DE LECTURE

La première étape du processus d'affichage sur l'écran du vêtement ayant été retiré, est l'identification de celui-ci. Pour ce faire, nous utilisons une Foxboard ainsi qu'un lecteur RFID Phidgets qui mettent à jour les informations de la base de données. Cet ensemble va nous permettre de faire la communication entre le vêtement que l'utilisateur sort de la penderie et l'application WEB avec laquelle il interagit. A cause de plusieurs problèmes techniques sur lesquels nous reviendrons plus tard, le programme sera testé sur un PC et non sur la Phidgets.

## B. L'APPLICATION

L'application de liaison entre le matériel (vêtements) et la base de données, peut être séparée en 2 parties :

### 1. LECTURE D'UN TAG

Pour détecter la présence d'un tag devant le lecteur RFID, nous utilisons une bibliothèque fournie par la carte Phidgets. Au début du projet, nous avons quelques difficultés avec Linux et les versions de bibliothèques disponibles sur le site. Nous nous sommes rendu compte que la bibliothèque qui fonctionne est la "libphidget\_2.1.7.20101222". Une fois la bonne bibliothèque installée, nous avons compilé un exemple de programme c permettant de lire et d'afficher les tags présents près de l'antenne RFID.

La première compilation ne fonctionnait pas car il manquait dans le code la déclaration d'une des variables. Une fois ce problème résolu, nous avons pu récupérer les différents identifiants afin de compléter notre base de données.

Cette bibliothèque possède toutes les fonctions dont nous avons besoin pour l'utilisation que nous allons faire :

- ✚ Connexion au lecteur
- ✚ Créations de Handlers

Le travail à faire sur cette partie a été minimal. En effet, il a suffi de garder les fonctions utiles pour notre application et de supprimer les autres. Le programme fonctionne maintenant de la façon suivante :

Il se connecte au lecteur puis attache deux fonctions Handler, une qui sera appelée lorsqu'un badge est passé devant la Phidgets et l'autre sera appelée lorsque qu'un badge quitte son « champ de vision » (ceci permettant aussi d'allumer la Led du lecteur, afin que l'utilisateur sache que le badge passé a bien été pris en compte).





Une fois que nous avons pu constater que notre programme fonctionnait lorsque la Phidgets était directement branchée sur un PC, nous avons souhaité l'exporter sur la Foxboard afin que le programme fonctionne en autonomie et afin de ne pas mettre une tour d'ordinateur dans la penderie. Nous avons donc pour cela cousu la carte Phidgets sur la penderie et nous avons installé notre Foxboard. Nous avons ensuite exporté tous nos programmes sur cette dernière.

## 2. MISE A JOUR DE LA BASE DE DONNEES

Pour mettre à jour la base de données par le programme, nous utilisons une bibliothèque créée par Mysql à savoir : Mysql Connector. Pour ce faire, nous utilisons des requêtes SQL.

Ainsi, la première chose que fait le programme C après avoir été lancé, est de se connecter à la base de données indiquée en paramètre et de mettre à 0 (=FAUX) le champ de la table indiquant qu'un vêtement a été retiré.

Puis, la fonction Handler créée plus haut, qui s'active à chaque fois qu'un tag est détecté, récupère la valeur du tag passé devant le lecteur et le met dans la base de données et plus précisément dans le champ « tag\_vetement\_en\_cours » de la table « tag\_temporaire ». Elle indique qu'un vêtement a été retiré en passant la valeur du champ « lecture\_vetement » à 1 (=VRAI) pendant 10 secondes. Pendant ces 10 secondes un Led est activée sur la carte Phidgets afin de signaler que le vêtement a bien été détecté. Ce temps permet également à l'application web de traiter la commande. Une fois ce temps écoulé, le programme C remet la valeur du champ « lecture\_vetement » à 0, état initial.

#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 tag_vetement_en_cours	varchar(10)	latin1_swedish_ci		Non	Aucune		 Modifier  Supprimer plus ▼
<input type="checkbox"/>	2 lecture_vetement	tinyint(1)			Non	Aucune		 Modifier  Supprimer plus ▼

## 3. LES DIFFERENTS PROBLEMES RENCONTRES

Parmi les principales difficultés que nous avons rencontrées, nous pouvons noter l'installation de la bibliothèque mysql c-connector. Celle-ci ne voulait pas s'installer à cause de Cmake qui n'était pas à jour. Mais avant de trouver cela, nous avons tenté de cross-compiler la bibliothèque sur un PC fixe pour la Foxboard, cela a pris beaucoup de temps, à nous ainsi qu'à M.Vantroys, et cette recherche a finalement échouée.

Puis nous avons découvert que la Foxboard ne pouvait pas faire automatiquement ses mises à jour, car elle allait les chercher sur un mauvais serveur. Avec l'aide de Mr Redon, la Foxboard fut capable à nouveau de faire ses propres mises à jour et a pu récupérer la dernière version de Cmake. Cette solution nous a permis de compiler la bibliothèque, et donc notre programme.

L'autre principal problème rencontré est celui que nous avons eu lors des dernières séances. En effet, nous avons découvert que la bibliothèque provoquait d'énormes fuites mémoires, allant de 3Mo à 87Mo de RAM utilisée lors de l'appel du



programme. Nous avons donc essayé de trouver d'où provenait ces fuites dans la bibliothèque, mais nous n'avons malheureusement rien trouvé.

Pour conclure, nous pouvons dire que le programme fonctionne donc bien sur une machine possédant suffisamment de RAM, mais ne fonctionne qu'une fois sur 2 sur la Foxboard.

# V. APPLICATION WEB

## A. OBJECTIF DE L'APPLICATION

Lorsque nous regardons le schéma du début du dossier, nous nous rendons compte que l'objectif principal de notre application est qu'elle doit être capable de récupérer le numéro du tag qui vient de passer devant la Phidgets. Une fois ce numéro récupéré, l'application doit nous donner toutes les informations sur le vêtement qui vient d'être sélectionné et nous dire si il est en adéquation avec la météo extérieure.

## B. CREATION DE L'APPLICATION

Tout d'abord, nous avons choisi de développer notre application sur internet afin de la rendre accessible depuis n'importe quel poste informatique. Nous allons y créer plusieurs pages Web qui auront chacune une fonction bien spécifique.

### 1. LA PAGE : CHOIX DU VETEMENT<sup>1</sup>

Cette page est la page la plus importante de notre site internet. C'est pour cela que nous commencerons avec celle-ci. Les autres pages correspondent à des fonctions optionnelles à notre application. En effet, cette page permet à l'utilisateur de sélectionner le vêtement qu'il souhaite porter et d'obtenir toutes les informations relatives à cet habit (adéquation avec la météo ...). Ensuite, cette page nous permet également de conseiller l'utilisateur. Nous listons en début de page tous les habits présents dans la penderie qui correspondent à la météo. Le choix est donc facilité pour l'usager.

Dans cette page nous récupérons tout d'abord la température extérieure grâce au METAR<sup>1</sup> de Lille/Lesquin (code LFQQ). Une fois cette valeur sauvegardée, nous la comparons aux températures minimales et maximales supportées par les vêtements présents dans la penderie et nous listons ceux qui sont en adéquation.

Ensuite, nous permettons à l'utilisateur de sélectionner le vêtement qu'il désire porter. Pour cela, il doit passer l'habit devant la Phidgets et grâce au programme de détection détaillé dans la partie précédente, notre application récupère le tag dans la table « tag\_temporaire » de notre base de données, et en fonction de ce tag l'application affiche la photo du vêtement sélectionné. Pour mettre à jour la photographie en fonction du tag qui passe devant la Phidgets, nous utilisons du code Ajax qui s'exécute toutes les secondes une fois que nous sommes sur la page vêtement. Dans cette fonction nous récupérons le tag du vêtement dans la table temporaire et nous allons chercher la photographie correspondante. Puis nous

---

<sup>1</sup> Un **METAR** (METeorological Airport Report) est un rapport d'observation (et non de prévision) météorologique pour l'aviation.

actualisons la présence du vêtement dans la penderie en inversant son état. Vous pouvez voir le code de cette « mise à jour <sup>ii</sup> » en annexe.

Enfin, notre application récupère grâce à différentes requêtes les températures associées au vêtement et signale à l'utilisateur si elles sont en correspondance avec la météo.

## 2. L'INDEX<sup>iii</sup>

L'index va nous permettre de mettre en forme le site internet. En effet, nous allons créer un style fixe qui sera sur toutes les autres pages de notre application. Nous avons choisi un style assez simple afin de mettre en avant les différentes images que nous allons créer. Nous avons voulu insérer le logo de Polytech'Lille car notre site y est hébergé.

Nous avons choisi de faire un menu à gauche de l'application pour accéder aux différentes options et de mettre à gauche un menu avec la météo. Il faut savoir que météo affichée sur le site est une vignette JavaScript que nous avons trouvée toute faite sur internet et qui est



reliée au site : <http://www.meteorologic.net>, alors que la météo avec laquelle nous comparons les températures de nos vêtements fonctionne avec les METARs. Enfin, nous avons créé grâce à un logiciel de 3D un logo avec un personnage appuyé sur un mur qui sera présent sur chacune de nos pages. C'est un peu le logo de notre site.



## 3. LA PAGE : ACCUEIL<sup>iv</sup>

La page d'accueil se lance dès que nous entrons l'adresse de notre site WEB («[e-wardrobe.polytech-lille.net](http://e-wardrobe.polytech-lille.net)»). Nous l'avons voulu communicative avec l'utilisateur. Par conséquent, nous avons créé un personnage qui salue l'utilisateur lorsque la page d'accueil est chargée.

## 4. LA PAGE : VOTRE PENDERIE<sup>v</sup>

La page Votre Penderie permet à l'utilisateur de savoir précisément tous les vêtements qui sont présents dans sa penderie. Nous avons donc fait une liste de la description de chacun des habits avec sur la même ligne la photo qui lui est associée.

D'ailleurs pour associer une photo à un vêtement, nous avons choisi de nommer chacune des images grâce au numéro du tag correspondant au vêtement. Prenons en exemple le vêtement « Pull Vert-Noir » qui associé au tag 01068dfccf, son image sera enregistrée sous le nom « 01068dfccf.jpg ». De cette manière, nous avons besoin que du tag du vêtement pour afficher toutes les informations qui lui sont associées telle que sa photo.

## 5. LA PAGE : GESTION PENDERIE<sup>vi</sup>

Cette page est une page intermédiaire qui permet de rassembler en menu les trois options qui seront le plus utilisées (selon nous) par l'utilisateur. Ces trois fonctions sont donc :

- ✚ Ajouter un vêtement
- ✚ Choix du vêtement
- ✚ Supprimer un vêtement

Sur cette page, nous avons quelques soucis avec l'image de la penderie. Après plusieurs essais et plusieurs modifications de l'image, nous nous sommes rendu compte que c'était juste un souci de droits d'accès à l'image qui empêchaient son chargement.



## 6. LA PAGE : AJOUTER UN VETEMENT<sup>vii</sup>

Cette page permet comme son l'indique à l'utilisateur d'ajouter un nouveau vêtement à sa penderie. Pour effectuer un ajout, l'utilisateur doit suivre les étapes que nous lui avons indiquées sur la page.

Premièrement, il doit passer le tag associé à son nouveau devant la Phidgets. Une fois le tag détecté par l'application, la suite des étapes s'affichent et lui permettent de saisir les informations associées à ce nouvel habit (sa description, ses températures maximale et minimale et la photo). Pour télécharger la photo, il suffit que l'utilisateur clique sur le bouton « Parcourir » et qu'il sélectionne dans ses dossiers personnels la photographie. Cette dernière est, grâce au code, renommée sous la forme standard « \$numéro\_tag.jpg » et enregistré dans la racine du site internet afin de pouvoir l'utiliser dans les autres fonctions de l'application.

## 7. LA PAGE : SUPPRIMER UN VETEMENT<sup>viii</sup>

Cette dernière page permet à l'utilisateur de supprimer un vêtement de sa penderie. Pour cela, nous listons grâce à un menu déroulant tous les vêtements présents dans l'armoire. Il suffit de sélectionner celui que l'on veut supprimer et de cliquer sur le bouton « Valider la suppression du vêtement ». Suite à cela, un message d'alerte s'affiche pour signaler la bonne suppression du vêtement.

## C. LES DIFFICULTES DE L'APPLICATION

Pendant la création de l'application nous avons eu quelques soucis de mise en forme. En effet, comme nous découvrons le langage CSS, nous avons eu quelques difficultés à positionner nos images, nos tableaux et autres objets à l'endroit voulu sur la page WEB.

De plus, comme nous l'avons vu précédemment, nous avons créé un table « tag\_temporaire » dans notre base de données qui possède un attribut lecture\_tag. Lorsqu'il est à 1, nous le refaisons repasser à 0 avec l'application une fois que les données étaient récupérées. Mais cette option nous causait quelques erreurs d'affichage. Par conséquent, nous avons choisi de faire remettre à 0 la lecture\_tag par le programme C. Au bout de 10 secondes, il le remet automatiquement à 0. Cette solution nous pose donc plus de soucis et permet en plus de laisser la LED de la Phidgets allumée pendant les 10 secondes de la lecture du tag.

Nous avons eu également un problème avec l'enregistrement de l'image lorsque l'utilisateur ajoute un nouveau vêtement. En effet, le fichier « numero\_tag.jpg » est bien créé dans la racine du site internet mais la taille du fichier n'est pas la bonne. Par conséquent, l'image ne s'affiche pas correctement sur le site.

## VI. CONCLUSION

Pour conclure ce rapport, ce projet nous a permis d'aborder l'HTML, le CSS et le PHP de façon plus approfondie. Mais nous avons été confronté à de nombreux problèmes, et avons pu, pour la majorité d'entre eux, les résoudre.

Nous avons donc été capable de fournir une version 1.0 d'une E-wardrobe pouvant dire à son utilisateur si le vêtement qu'il vient de retirer est compatible avec la température extérieure. Reste à améliorer l'application web pour fournir des options supplémentaires (la coordination des couleurs des vêtements par exemple), ainsi que le programme de liaison, pour trouver et colmater les fuites mémoires.







```

<td align="left" width="105"><a href="http://www.polytech-lille.fr/">
<td align="right">
<div class="head">
<!-- Title of Your Website and description, keep </div> tag behind the title of the website, if you don't want a descri
<FONT COLOR="#FFFFFF">E-Wardrobe <br>
Polytech'Lille</FONT>
</div>
<!-- End of Title -->
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td align="left" background="images/trait.png">
<table cellpadding="4" cellspacing="0" border="0">
<tr>
<td align="left">
Bienvenue <?php echo $_SESSION['nom_user'] ; ?>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td valign="top">
<table border="0" width="100%" cellspacing="0" cellpadding="6">
<tr>
<td width="20%" valign="top"><br><br>
<!-- Put your links in here, use arrow2.gif for any Sub Links -->
&nbsp;<a href="index.php?page=accueil"><b>Accueil</b></a><br><br>
&nbsp;<a href="index.php?page=votre_penderie"><b>Votre Penderie</b></a><br><br>
&nbsp;<a href="index.php?page=gestion_penderie"><b>Gestion Penderie</b></a><br><br>
&nbsp;<a href="index.php?page=ajouter_vetement"><b>Ajouter Vêtement</b></a><br><br>
&nbsp;<a href="index.php?page=choix_vetement"><b>Choix Vêtement</b></a><br><br>
&nbsp;<a href="index.php?page=supprimer_vetement"><b>Supprimer Vêtement</b></a><br><br>
<!-- End of Your Links -->
</td>
<td width="2" valign="top">
</td>
<td width="80%" valign="top"><br>
<div style="position: relative;">
<!-- Body content goes here -->
<?php
// Include de la liste des pages autorisées a être affiché
include('include/conf/page_autorisee.php');
// La variable $page existe-elle dans l'url ?
if(!empty($_GET['page']))
{
// Vérification de la valeur passée dans l'url : est-elle une clé du tableau ?
if(array_key_exists(strtolower($_GET['page']), $arrayPages))
{
// Oui, alors on l'importe
if (file_exists('include/page/'. $arrayPages[ strtolower($_GET['page']) ] ) ) include('include/page/'. $arrayPa
else echo "<center> La page n'est pas présente sur le serveur !</center>";
}
}
else
{
// Non, alors on importe un fichier par défaut
echo "<center> La page demandé n'est pas définie dans la liste des pages autorisées a être publier !</center>";
}
}
else
{
// Non, on affiche la page d'accueil par défaut
include('include/page/'. $arrayPages['']);
}
mysql_close;
?>
<!-- End of body content-->
</div>
</td>
<td width="2" valign="top">
</td>
<td width="80%" valign="top"><br>
<script language="javascript">
var fond=''; //couleur de fond de la vignette en hexadécimal sans #
var texte = ''; //couleur du texte en hexadécimal sans #
var id_commune = '29591'; //id de la commune qui s'affiche par défaut
var police = ''; //police d'écriture
</script>
<script language="javascript"
src="http://www.meteorologic.net/webmaster/new/javascript_vignette.js">
</script>
<br><br><br><br><br><br><br><br><br><br>
<div id="logo" style="position: relative;">
</div>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td background="images/trait.png" align="left">

```



## vii Code de la page Ajouter un Vêtement

```
<audio src="/images/ajout.wav" autoplay></audio>
<form action="<?php echo $_SERVER[REQUEST_URI]; ?>" method="post" name="ajouter_vetement">
<?php
$requete = "SELECT * FROM tag_temporaire";
$phpreq = mysql_query($requete);
while($srep = mysql_fetch_object($phpreq))
{
    $stag = $srep->tag_vetement_en_cours;
    $lecture = $srep->lecture_vetement;
}
?>
<center><h3> Vous voulez ajouter un nouveau vêtement ... </h3><br><h4>Etape 1 : Passez un tag devant la phidget</h4>
<?php
if($lecture == 1)
{
    echo "Voici le numéro de tag que vous venez de passer : $tag"?>
    <br><br>
    <table border="0" width="40%" cellpadding="0" cellspacing="5">
        <tr>
            <td><h4>Etape 2 : Entrez la description du vêtement<b><br></b></td><input type="text" name="description_vetement" value="<?php echo $_POST['descrip
            </tr>
            <tr>
            <td><h4>Etape 3 : Entrez les températures minimales et maximales supportées par le vêtement<b><br></b><br> Température minimale<input type="text"
            <br>Température maximale<input type="text" name="temp_max" value="<?php echo $_POST['temp_max'] ?>"/></td>
            </tr>
            <tr>
            <td><h4>Etape 4 : Choisissez la photo correspondante à votre nouveau vêtement<b><br></b><br><input type="file" name="textfield" /></td></tr>
        </table></center><br><br>
    <?php
}
?>
<br><br>
<center><input type="submit" name="valider" value="Valider la saisie du vêtement" onClick="ajouter_vetement;" /></center><br> <br>
<?php
//print_r($_POST);
if($_POST['valider']=="Valider la saisie du vêtement")
{
    $stag=$tag.".jpg";
    echo $stag;
    $sql = "INSERT INTO vetement(tag_vetement,description_vetement,presence_vetement,temp_min_vetement,temp_max_vetement) VALUES ('".$stag."','".$_POST['descrip
    $requete = @mysql_query($sql) or die($sql."<br>".mysql_error());
    echo " Insertion effectuée";
    $f1 = fopen("/images/".$stag.".","w");
    if (fwrite($f1,$_POST['textfield']) === FALSE) {
        echo "Impossible d'écrire dans le fichier ($filename)";
        exit;
    }
    fclose($f1);
}
?>
</form>
```

## viii Code de la page Supprimer un Vêtement

```
<form action="<?php echo $_SERVER[REQUEST_URI]; ?>" method="post" name="supprimer_vetement">
<center>
<h3>Pour supprimer un vêtement sélectionnez le dans la liste déroulante suivante puis validez la suppression</h3>
<?php
//menu déroulant pour le vetement
$sql = "SELECT * FROM vetement WHERE presence_vetement = 1 ORDER BY description_vetement ";
$affiche = mysql_query($sql);
echo "<table border='0' width='30%' cellpadding='0' cellspacing='0' align='left'>";
echo "<tr align='left'>";
<td><b>Voici tous les vêtements : </b></td><td><select name="tag_vetement">
<option value=""></option>";
while($res = mysql_fetch_object($affiche))
{
    $stag = $res->description_vetement;
    echo "<option value='".$res->tag_vetement.'";";
    if ($_POST['tag_vetement'] == $res->tag_vetement)
    {
        echo " selected ";
    }
    echo ">$stag</option>";
}
echo "</select></td>";
echo "</tr></table>";
echo "<br><br><br><br><br>";
?>
<br><br>
<center><input type="submit" name="supprimer" value="Valider la suppression du vêtement" onClick="supprimer_vetement;" /></center><br> <br>
<?php
if($_POST['supprimer']=="Valider la suppression du vêtement")
{
    $sql = "DELETE FROM vetement WHERE tag_vetement = '".$_POST['tag_vetement']."'";
    $requete = @mysql_query($sql) or die($sql."<br>".mysql_error());
    echo " Suppression effectuée";
}
?>
</center>
</form>
```