



DOCUMENTATION TECHNIQUE

Stage en entreprise – du 20 Mai au 30 Juin 2022

RESUME

Compte rendu des missions effectués sur
l'application métier de Flexocolor

Guillaume PASCAL

BTS SIO 1ère Année – Lycée Valadon, Limoges

Table des matières

L'entreprise : Flexocolor	2
Informations générales :	2
L'activité de Flexocolor :	2
Ma place dans l'entreprise :	2
Présentation de l'application métier :	3
Matériels et logiciels utilisés :	3
Mission 1 : Recherche graphique pour l'entête de la facture	4
Mission 2 : Afficher le récapitulatif du client pour chaque facture	6
Mission 3 : Modification graphique du corps de la facture	13
Mission 4 : Modification graphique de l'ajout d'un intervenant sur les informations de la commande.	15
Mission 5 : Modification graphique de la sélection du donneur d'ordre.	17
Mission 6 : Réaliser l'historique de gestion des M²	19
Mission 7 : Refaire l'interface de recherche des différents champs Administrateur.	23
Conclusion de fin de stage :	24

L'entreprise : Flexocolor

Informations générales :

Flexocolor est une entreprise française dont le siège social est basé au Palais-Sur-Vienne. Elle possède 4 antennes sur l'ensemble de la France : Nantes, Tours, Noisy-le-Grand et Sainte-Sigolène.

L'entreprise a une centaine de salariés (Limoges et ses antennes).

L'activité de Flexocolor :

Flexocolor est une entreprise spécialisée dans la photogravure. La photogravure consiste à graver sur une plaque ou en manchon en polymère en appliquant de l'encre sur ses supports et en la transposant sur des supports plastiques ou carton d'avoir un packaging de qualité.

L'activité de Flexocolor va de la modification du fichier envoyé par le client pour y incorporer toutes les informations techniques, tout en prenant compte des contraintes techniques des imprimeurs, jusqu'à la réalisation de la plaque ou du manchon qui sera envoyé à l'imprimeur.

Ma place dans l'entreprise :

Je suis stagiaire au sein du service Méthode et Technologie dirigé par Mr MERLE. Mon objectif était d'aider à développer certaines parties de l'application métier.

Présentation de l'application métier :

L'application métier se compose de 4 parties :

- L'ancienne application dont presque plus personne ne se sert (*seulement les salariés de Flexocolor Corrugated s'en servent*)
- L'application de développement nommé FALCODev qui sert à développer les nouvelles fonctionnalités
- L'application bêta sert à tester les nouvelles fonctionnalités avec la base de données de l'entreprise. Les fonctionnalités sont testées par des Bêta Testeurs avant d'être mis en place sur l'application. De temps en temps, des réunions sont organisées avec eux pour faire le point sur les fonctionnalités à tester.
- L'application dont se sert l'entreprise.

Matériels et logiciels utilisés :

- Un ordinateur fournit par l'entreprise
- Visual Studio Code
- PHP 5.2.9
- IIS (serveur web en local)

Mission 1 : Recherche graphique pour l'entête de la facture

Dans cette mission, l'objectif était de refaire l'interface de l'entête de la facture afin qu'il soit plus agréable à l'œil avant d'attaquer le PHP.

Afin de permettre une meilleure visibilité, j'ai décidé de trier les informations en fonction de leur nature.

Pour se faire, j'ai réalisé un schéma de la page afin d'y voir plus clair :

[Insérez schéma ici]

Après réflexion et approbation, j'ai commencé à modifier le graphisme de la page. J'ai décidé de faire les regroupements avec des fieldsets :

```
fieldset class="BlocFieldset">
  <legend>Période</legend>
  <table>
    <tr>
      <td class="tabTitle" align="center" colspan="4">
        DU <input type="text" id="startDate" name="startDate" title="Date de fin" size="12" maxlength="10" value="" onkeypress="verifDate(this,3,event)" onblur="sortieDate(this,event)">
        
        AU <input type="text" id="endDate" name="endDate" title="endDate" size="12" maxlength="10" value="" onkeypress="verifDate(this,3,event)" onblur="sortieDate(this,event)">
        
      </td>
    </tr>
  </table>
</fieldset>
```

La class BlocFieldset permet de donner un style au Fieldset :

```
.BlocFieldset{
    border:solid;
    text-align: center;
    margin: auto;
    border-color:var(--dore);
    background-color: var(--blanctransparent);
    border-radius: 25px;
    width: 40%;
}
```

Ce qui donne le résultat final suivant :


L'En-tête de la Facture

Emetteur*	GRENOULAUD Damien
Site*	Site : Limoges
Date*	04/02/2022
N° Facture*	F22020002

Adresse de facturation

Adresses de facturation :	45 Rue Voltaire 87000 France
Zone de facturation*	France
Mode et délais de règlement*	Délais : 30 jours Mode : Chèque

Client*

client demo 

Adresse de facturation*

Nom du client :

Client demo 13

Rue :

45 Rue Voltaire

CP :

87000

Ville :

France

Pays :

France

Informations client

Type de commande	Scamark
Type d'impression	Flexo
Suivant facture :	Aucune
Suivant devis :	Aucun
Référence	test creation
N° Commande Client	Numéro commande client
	N° FC : L470414
Poids clichés (en Kg)	
N° Rangement	2202011863

Memo

 Valider l'entête du devis

Il a ensuite fallu enlever le fond gris transparent de certains champs afin de rendre l'ensemble plus lisible.

L'En-tête de la Facture

Emetteur*	GRENOULAUD Damien
Site*	Site : Limoges
Date*	04/02/2022
N° Facture*	F22020002

Adresse de facturation



Adresses de facturation :	45 Rue Voltaire 87000 France
Zone de facturation*	France
Délais de règlement*	30 jours
Mode de règlement*	Chèque

Client*

client demo 

Nom du client :

Client demo 13

Rue :

45 Rue Voltaire

CP :

87000

Ville :

France

Pays :

France

Informations commandes

Type de commande	Scamark
	Chef de produit : - -
Type d'impression	Flexo
Suivant facture :	Aucune
Suivant devis :	Aucun
Référence	test creation
N° Commande Client	Numéro commande client
	N° FC : L470414
Poids clichés (en Kg)	
N° Rangement	2202011863

Memo

Mission 2 : Afficher le récapitulatif du client pour chaque facture

Dans le but de rendre la facturation plus pratique, j'ai rajouté, à l'aide d'un bouton, l'ouverture d'un nouvel onglet affichant toutes les informations du client concerné par la facture.

Remarque : Un client peut avoir plusieurs adresses (siège social etc...) mais une ou plusieurs peuvent être qualifié « d'adresse par défaut », c'est-à-dire qui remonte en premier lors de la facturation.

Pour les graphismes, j'ai repris le même principe que la mission 1.

Pour commencer j'ai créé un nouveau fichier et j'ai pris une image pour la mettre en tant que bouton, puis mener cette action vers la page nouvellement créer. De plus, il faut faire passer le numéro du client dans l'URL afin de le récupérer dans la page créer et pouvoir exécuter les requêtes nécessaires au fonctionnement de cet outil.

Voici un aperçu du code utilisé pour faire le lien :

```
<div id="BtnInfoClient">
  <a onclick="window.open('recapInfoClient.php?idClient=<?echo $idClient?>','MyPopupWindow','height=600,width=1000, top=0,left=0,resizable=no,menubar=no,status=no ,scrollbars=no, menubar=no')*" class="BtnInfo">
    </a>
</div>
```

Remarque : le « <?echo \$idClient?> » permet de faire passer l'id du client dans l'URL et ainsi permettre de le récupérer dans la page suivante.

Création de la page web :

Afin de mener à bien la mission, je dois tout d'abord récupérer l'id du client passé dans l'URL :

```
$idclient = $_REQUEST['idClient'];
```

Cela me permet de récupérer une donnée d'un formulaire, qu'il soit en méthode GET ou POST.

Pour commencer, je dois afficher le nom du client, son site de rattachement, le transporteur de la commande et s'il a un relevé de facture comme ceci :

Informations Client		
client demo		
Site de rattachement	Transporteur	Relevé de Facture
Limoges	Chronopost	Non

L'ensemble de ces informations sont situées dans la table client. On effectue donc une requête et on récupère les informations dont on a besoin dans des variables. Ces variables seront affichées dans un tableau.

Requête pour obtenir les informations du client :

```
/*Requête pour obtenir les infos du client*/

$req = "Select * from Client where id_client = $idclient";

/*Résultat de la requête*/
$resultat = mssql_query($req);
$result = mssql_fetch_assoc($resultat);
$site = $result["ID_SITE"];
$transporteur = $result["TRANSPORTEUR"];
```

Une fois la requête exécutée et les informations récupérées dans un tableau associatif, on récupère dans les variables \$site et \$transporteur les informations nécessaires.

Cependant, on récupère l'id du site client et pas son nom.

Pour se faire on exécute une nouvelle requête permettant de récupérer le libelle du site.

```
/*Requête pour récupérer le site de rattachement*/

$reqSite = "Select LIBELLE_SITE from Site where ID_SITE = $site";
$resultatNomSite = mssql_query($reqSite);
$resultNomSite = mssql_fetch_assoc($resultatNomSite);
$NomSite = $resultNomSite["LIBELLE_SITE"];
```

On récupère ici, le libelle du site dans la variable \$NomSite qui sera affiché dans le tableau.

Comme précédemment, on récupère seulement le numéro de transporteur, et on doit donc effectuer une nouvelle requête afin de récupérer le mode de livraison (soit le nom du transporteur de la commande).




```

/*Requête pour récupérer le mode de livraison */
$reqTransporteur = "Select NOM_MODE_LIVRAISON from mode_livraison where
ID_MODE_LIVRAISON = $transporteur";

$resultatTransporteur = mssql_query($reqTransporteur);
$resultTransporteur = mssql_fetch_assoc($resultatTransporteur);
$NomTransporteur = $resultTransporteur["NOM_MODE_LIVRAISON"];

```

Affichage des informations :

Pour afficher les informations, on comme dans l'image ci-dessus, on ouvre un Fieldset, puis un tableau dans lequel on affiche les variables avec des « echo » :

```

<fieldset class="BlocFieldset3">
  <legend><b>Informations Client</b></legend>
  <table style="width: 100%;">
    <b><?php echo $result["LIBELLE_CLIENT"]?></b><br/>
    <tr>
      <th>Site de rattachement</th>
      <th>Transporteur</th>
      <th>Relevé de Facture</th>
    </tr>
    <tr>
      <td>
        <?php echo $NomSite;?>
      </td>
      <td>
        <?php echo $NomTransporteur;?>
      </td>
      <td>
        <?php
        if($result["RELEV_CLIENT"] == 0){
          echo "Non";
        }else{
          echo "Oui";
        }
        ?>
      </td>
    </tr>
  </table>
</fieldset><br/>

```

Remarque : le relevé du client n'a que 2 états : 0 et 1. 0 étant égal à Non et 1 à Oui. Donc, afin d'afficher la bonne information, on effectue une comparaison en prenant le résultat de la requête des informations clients.

Affichage des adresses du client :

Cette partie doit contenir le nom client adresse, son adresse, le code postal, la ville, le pays, le type d'adresse (livraison, siège social etc...), si l'adresse est active ou non et si c'est une adresse par défaut.

Adresses du Client							
Nom Client adresse	Adresse	Code Postal	Ville	Pays	Type adresse	Actif	Adresse par défaut
GER'SON PIERROT GOURMAND	LES ESCURES	19120	ALTILLAC	France	Tout	Actif	OUI

Pour ce faire, on doit récupérer toutes les adresses du client (grâce à l'id du client) :

```
/*Requête pour les adresses*/  
  
$reqAdresse = "Select * from adresse where id_client = $idclient";  
$resultatAdresse = mssql_query($reqAdresse);
```

Comme un client peut avoir différentes adresses en fonction du type (livraison...), on doit faire une boucle lors de l'affichage :

```
while($resultAdresse = mssql_fetch_assoc($resultatAdresse)){
```

L'affichage se fait donc dans un tableau comme ceci :

```
<td><?php  
    echo $resultAdresse["NOM_CLIENT"];  
    ?>  
</td>
```

Pour l'affichage du pays, on effectue tout d'abord une requête dans la base de données afin de récupérer le nom du pays :

```
/*Requête pour les pays*/  
  
$reqPays = "Select * from pays";  
$resultatPays = mssql_query($reqPays);
```

Puis, on met tous les pays dans un tableau type array avec en clé l'id du pays puis le libelle du pays :

```
$tableauPays = "";  
while($resultPays = mssql_fetch_assoc($resultatPays)){  
    $tableauPays[$resultPays["ID_PAYS"]] = $resultPays["LIBELLE_PAYS"];  
}
```

L'affichage se fait donc ainsi :

```
</td>  
  
    <td><?php  
        echo $tableauPays[$resultAdresse["ID_PAYS"]];  
    ?>  
    </td>
```

Remarque : Lors de la création de cette page, il était impossible de faire une requête pour déterminer le libelle du type adresse (la table en question n'existait pas dans la base de données). J'ai donc utilisé le système de condition pour effectuer cette partie :

```
<td><?php  
    if($resultAdresse["TYPE_ADRESSE"] == 0){  
        echo "Siège Social";  
    }  
    else if($resultAdresse["TYPE_ADRESSE"] == 1){  
        echo "Facturation";  
    }  
    else if ($resultAdresse["TYPE_ADRESSE"] == 2){  
        echo "Livraison";  
    }  
    else if ($resultAdresse["TYPE_ADRESSE"] == 4){  
        echo "Tout";  
    }  
    ?>
```

On utilise le même principe pour savoir si l'adresse est active ou non et pour l'adresse par défaut.

```
</td>
    <td><?php
        if($resultAdresse["ACTIF"] == 0){
            echo "Inactif";
        }else{
            echo "Actif";
        }
    ?>
</td>
    <td><?php
        if($resultAdresse["DEFAULT_ADRESSE"] == 0){
            echo "NON";
        }else{
            echo "OUI";
        }
    ?>
</td>
</tr><?php
```

Affichage des contacts :

Cette partie doit afficher le nom, prénom mail et téléphone du contact du client. Un client peut avoir plusieurs contacts.

Contacts client			
Nom	Prénom	Mail	Téléphone
ROUSSEL	Frédéric	frederic.rousseau@andros.fr	
GOUYGOU	Guillaume	guillaume.gouygou@andros.fr	
MANAS	Sandrine	sandrine.manas@andros.fr	
PLAZE	SOLANGE	solange.plaze@andros.fr	
FACTU		Gerson.compta@andros.fr	
FOULHAC			

On commence par récupérer les informations nécessaires avec une requête :

```
$reqContact = "Select * from contact as ct inner join contact_client as ctCli ON ct.ID_CONTACT =  
ctCli.ID_CONTACT where ID_CLIENT = $idclient";  
  
$resultatContact = mssql_query($reqContact);
```

Remarque : il peut y avoir plusieurs contacts pour un seul et même client, il sera nécessaire, lors de l'affichage, d'effectuer une boucle.

```
<fieldset class="BlocFiedlset3">  
  <legend><b>Contacts client</b></legend>  
  <table style="width: 100%;">  
    <tr>  
      <th>Nom</th>  
      <th>Prénom</th>  
      <th>Mail</th>  
      <th>Téléphone</th>  
    </tr>  
    <?php  
    while($resultContact = mssql_fetch_assoc($resultatContact)){  
      ?>  
      <tr>  
        <td><?php  
          echo $resultContact["NOM_CONTACT"];  
        ?>  
      </td>  
        <td><?php  
          echo $resultContact["PRENOM_CONTACT"];  
        ?>  
      </td>  
        <td><?php  
          echo $resultContact["MAIL_CONTACT"];  
        ?>  
      </td>  
        <td><?php  
          echo $resultContact["TEL_CONTACT"];  
        ?>  
      </td>  
      </tr><?php  
    }>  
  </table>  
</fieldset>
```

Mission 3 : Modification graphique du corps de la facture

Dans l'optique de mettre le corps de la facture avec l'entête de la facture, il m'a fallu refaire l'interface graphique.

Remarque : la page était codée en TPL (*HTML avec des variables*). Afin de pouvoir faire la mission correctement, il a fallu remplacer les variables TPL en PHP (exemple : {variable} -> <?php echo \$variable ;?>).

Action	Réf.	Désignation	Valeur	Tarif HT	Remise	PU Remisé	Mont. HT	
HP		Epreuve HP	1	100.00	0.0000	100.00	100.00	
ECART		ECART	1	0.00	0.0000	0.00	0.00	
ST		Sous Total					100	
							Total HT	100.00
							TVA (20 %)	20.00
							Total TTC	120.00

VALIDER

Par exemple, en tpl, le fieldset était codé comme ceci :

```
<fieldset style="{visibilityElements}" class="BlocFiedlset"><legend><b>Éléments de l{valTitre}</b></legend>&nbsp;&nbsp;&nbsp;|
```

J'ai remplacé le {visibilityElement} par :

```
<fieldset style="<?php echo $visibilityElements;?>"
```

Et {valTitre} par :

```
class="BlocFiedlset"><legend><b>Éléments de l<?php echo $valTitre;?></b></legend>&nbsp;&nbsp;&nbsp;|
```

La facture simplifiée, la facture détaillée et le relevé :

La facture simplifiée est la facture qui comprend tous les articles alors que la facture détaillée comprend uniquement les sous-totaux. Le dernier bouton permet de créer un relevé de facture (*un groupe de facture*).

Facture détaillée Facture simplifiée Relevé

IMPRIMER MAILER IMPRIMER MAILER IMPRIMER MAILER RELEVÉ F.

Pour la facture détaillée, on utilise ce principe :

```
$ArticleMultiple .= "<tr>";
$ArticleMultiple .= "<td colspan='8' align='center'>";
$ArticleMultiple .= "<br><center><img src='../design/images_site/valid.gif' onclick='sendAdd.redirect.value=1;sendAdd.submit()' style='cursor:hand' alt='Modifier les modifications'></center>";
$ArticleMultiple .= "</td>";
$ArticleMultiple .= "</tr>";
$ArticleMultiple .= "</table>";
$ArticleMultiple .= "<br><br>";
$ArticleMultiple .= "<div id='txt1' style='position:absolute;;z-index:9999'>";
$ArticleMultiple .= "<table width='100%'>";
$ArticleMultiple .= "<tbody><tr>";
$ArticleMultiple .= "<td width='50%'>";
$ArticleMultiple .= "<fieldset><legend><b>Facture d@tall@e</b></legend><br>";
$ArticleMultiple .= "<center>";
$popup = ".../impressionPDF/imp_factures.php?type=1&elementType=FactureMultiple&idComande=".$tabIdComande[$i]."";
$texte = "Imprimer";
$guillemet = """;
$ArticleMultiple .= "<img src='../design/images_site/imprim.gif' onclick='PopupCenter('.$popup.','.$texte.','800,600,','.$guillemet.')' style='cursor:hand' alt='Imprimer cette facture'>";
$ArticleMultiple .= "&nbsp;";
$tabDate = explode("/",transformDate2($tempF->date_facture));
if(!file(PATH_FACTURE_CHEMIN."FILESXOLOR/factures/".$tabDate[1].$tabDate[2]."/".$tempF->num_facture.".pdf")){
    $ArticleMultiple .= "<img src='../design/images_site/mailler.gif' onclick='PopupCenter('\"./mail/pop_envoiFacture.php?type=1&elementType=FactureMultiple&idComande="
    [$tabIdComande[$i]].\"&quot;,&quot;imprimeeFacture&quot;,600,350,&quot;scrollbars=yes&quot;)'\" alt='Mailier la facture' style='cursor:hand'>";
}
```

Remarque : on utilise des variables afin de pouvoir y mettre autant d'article qu'il y en a.

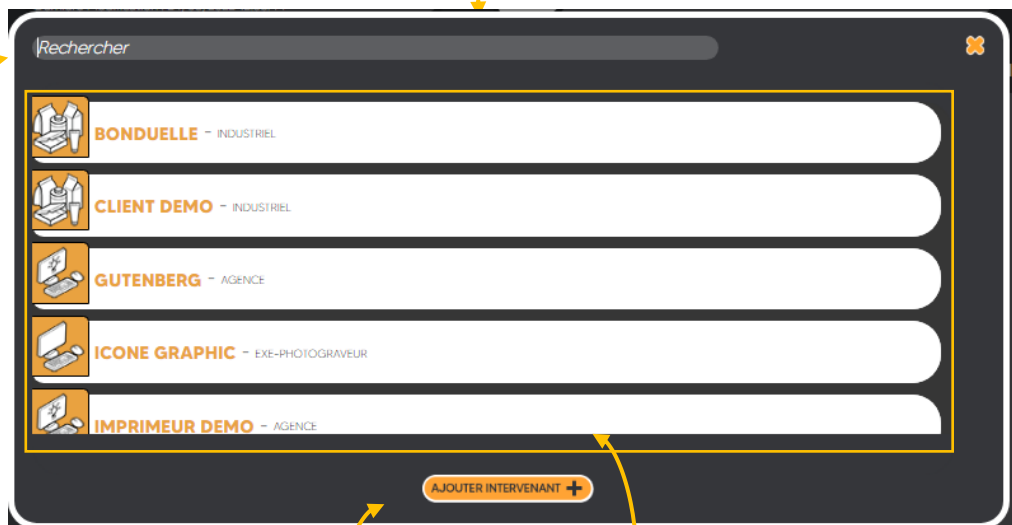
Le même principe est utilisé pour la facture simple et le relevé.

Mission 4 : Modification graphique de l'ajout d'un intervenant sur les informations de la commande.

L'ajout d'un intervenant sur les informations de la commande est une fonctionnalité permettant de gérer les intervenants présents sur la commande.



Barre de recherche



L'interface est séparée en 2 parties :

La partie « add_Intervenant » : Elle contient le contour de l'interface, le bouton et la barre de recherche

La partie « search_Intervenant » : Elle contient l'ensemble des intervenants disponibles.

La modification graphique c'est concentré sur la partie « search_Intervenant » (la partie « add_Intervenant étant déjà faite).

```
while ($row = mysql_fetch_array($res)) {  
    $i++;  
    $typeIntervenant = $row['LIBELLE_TYPE_INTERVENANT'];  
    $Intervenant = $row['LIBELLE_INTERVENANT'];  
    $idIntervenant = $row['ID_INTERVENANT'];  
  
    $onclick = 'onClickIntervenant("' . $Intervenant . '")';  
    echo  
  
    <tr class="" id="ligne3_"$i"" style="height:33px" onclick="$onclick">  
        <td style="width: 50px;height: 50px;vertical-align: top;border-radius: 25px 0 0 25px">  
              
        </td>  
        <td style "color: black;background-color: white;padding-left: 10px;position:relative;left:10px; border-radius:0 25px 25px 20px">  
            <span style"color: var(--dore);font-size: 16px;font-weight: bold;display: inline-block;vertical-align: middle;text-transform: uppercase">".utf8_encode(stripslashes($Intervenant)).  
            "</span> - <span style"font-size: 10px;display: inline-block;vertical-align: middle;text-transform: uppercase">".utf8_encode(stripslashes($typeIntervenant))."</span>  
        </td>  
    </tr>  
    <tr height="10px" id="ligne4_"$i"">  
        <td>  
        </td>  
    </tr>  
    ">  
}</table>";  
echo("</table>");
```

La boucle while permet d'afficher tous les enregistrements.

Ici, on donne aux différentes variables l'enregistrement en question issu de la requête.

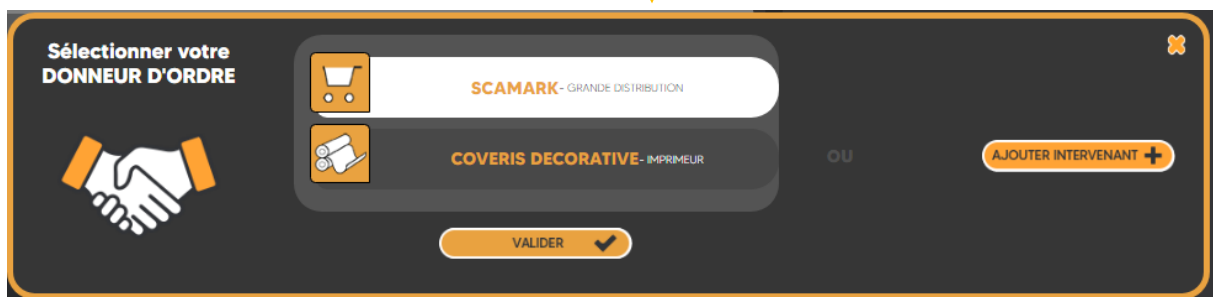
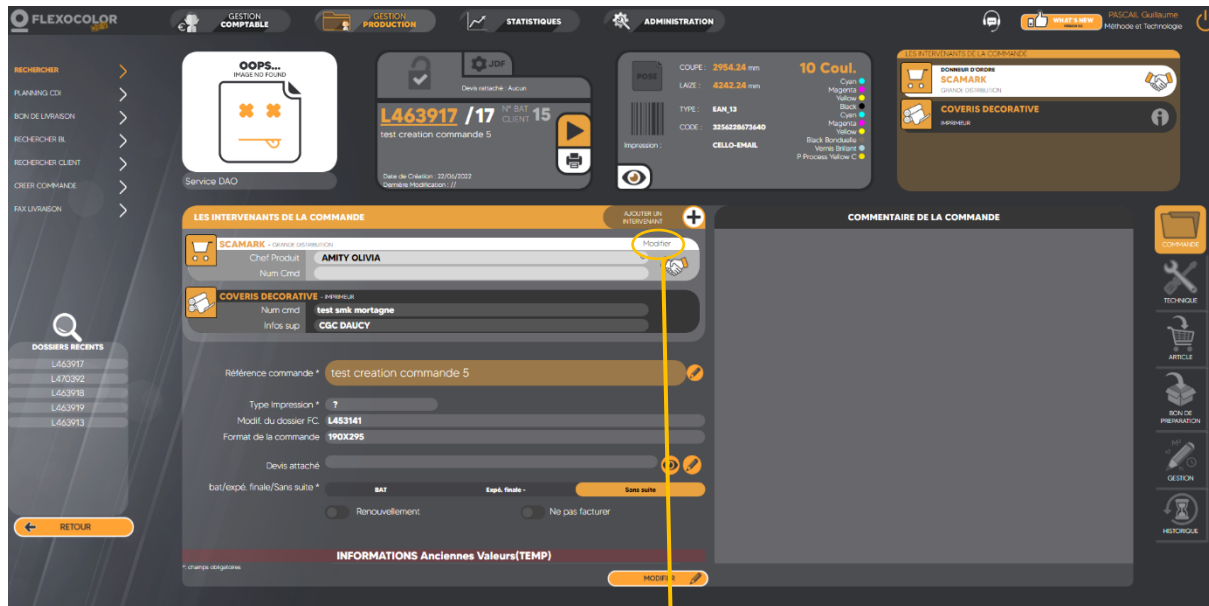
L'affichage : à l'aide des variables, on affiche les résultats

Chaque type d'intervenant à une image qui lui est propre. Donc ici, on affiche l'image correspondant au type d'intervenant.

Ici on définit les caractéristiques CSS de l'images (cela évite de créer une class CSS, surtout si on l'utilise une seule fois comme ici).

Mission 5 : Modification graphique de la sélection du donneur d'ordre.

Cette interface doit permettre modifier un intervenant.



L'ensemble est constitué d'une seule partie, définit par une div dans le code :

```
<div style="display:none;position:fixed;width:50%; background-color:var(--gris);padding:15px;z-index:7;border-radius: 25px;top: 40%;left: 25%; border: 6px solid var(--dore)" id="modifDonneurDordre">
```

La présentation se fait sous la forme d'un tableau :

Une balise `<th>` ici avec le texte : « Sélectionner votre donneur d'ordre »
Avec un `colspan` de 2.

Une balise `<td>` ici avec l'image correspondante

Un bouton de validation

Un tableau dans une balise `<th>` du premier tableau afin d'organiser au mieux les éléments.

Une balise `<th>` puis dans une balise `<p>`

L'image est ajoutée dans une balise `<th>`

```
<td style="width: 50px; vertical-align: top">
  
</td>
<td class="dimInfoClient">
  <span style="color: var(--dore); font-size: 16px; font-weight: bold; display: inline-block; vertical-align: middle; text-transform: uppercase"><?php echo $listeTexteIntervenant[$i] ?></span>
  <span style="font-size: 10px; display: inline-block; vertical-align: middle; text-transform: uppercase"> <?php echo $listeLibelleTypeIntervenant[$i] ?></span>
</td>
```

On insère l'image correspondant au type d'intervenant

Ici, on définit la class de la balise `<td>`

On affiche le libelle du type de l'intervenant

On affiche le nom de l'intervenant

Mission 6 : Réaliser l'historique de gestion des M²

Le but de cette mission était d'afficher l'historique des avenants et, pour chaque avenant, afficher l'historique des articles qui ont été ajoutés.

Avenant n°16											
Valeur M2	Utilisateur	Date M2	Type d'Article	Longueur	Largeur	Quantité	Nombre couleur	Epaisseur	Résolution	Matière	Machine NPF
2	GRENOUAUD Damien	17/12/2021	Epreuve A4	0	0	1	3				1
3	GRENOUAUD Damien	17/12/2021	EMAIL	0	0	1					
0.032775	GRENOUAUD Damien	17/12/2021	PAPIER	57	575	1					
1.025564	GRENOUAUD Damien	17/12/2021	PAPIER	242	4242	1					
1	GRENOUAUD Damien	17/12/2021	TRANSPARENT	0	0	1					
0.300564	GRENOUAUD Damien	17/12/2021	PAPIER	1242	242	1					
1	GRENOUAUD Damien	17/12/2021	TRANSPARENT	0	0	1					
1	GRENOUAUD Damien	17/12/2021	PAPIER	0	0	1					
0.102124	GRENOUAUD Damien	17/12/2021	TRANSPARENT	242	422	1					
0.058564	GRENOUAUD Damien	20/12/2021	PAPIER	242	242	5					
1	GRENOUAUD Damien	20/12/2021	PAPIER	0	0	1					
0.058524264	GRENOUAUD Damien	20/12/2021	CLICHE	242	242	1					
0.058524264	GRENOUAUD Damien	20/12/2021	CLICHE	242	242	1					
1.75	GRENOUAUD Damien	21/12/2021	POSE UNITAIRE	0	0	1					
1.75	GRENOUAUD Damien	21/12/2021	Multipose	0	0	1					
0.000576	GRENOUAUD Damien	22/12/2021	CLICHE	24	24	1					
2	GRENOUAUD Damien	23/12/2021	GMG VALDEE	0	0	1					
2	GRENOUAUD Damien	23/12/2021	DEPOSE FTP	0	0	1					
1	GRENOUAUD Damien	23/12/2021	PAF LIMOGES	0	0	1					
1	GRENOUAUD Damien	22/12/2021	ANCIENNE GMG	0	0	1					
1	GRENOUAUD Damien	23/12/2021	FICHE DE CONFORMITE	0	0	1					
1	GRENOUAUD Damien	23/12/2021	ECHANTILLONS IMPRIMES	0	0	1					
1	DELHOUME MARION	23/12/2021	CIBLE COULEUR	0	0	1					
1	Brun Aurelie	23/12/2021	ECHANTILLONS IMPRIMES	0	0	1					
1	Brun Aurelie	24/12/2021	FICHE DE CONFORMITE	0	0	1					
0.102608	GRENOUAUD Damien	27/12/2021	PAPIER MAT	242	424	2					

Avenant n°17											
Valeur M2	Utilisateur	Date M2	Type d'Article	Longueur	Largeur	Quantité	Nombre couleur	Epaisseur	Résolution	Matière	Machine NPF
1.5	GRENOUAUD Damien	11/05/2022	POSE UNITAIRE	142	24	1					
0.003408	GRENOUAUD Damien	13/05/2022	CLICHE			4		17	4000 HD	ITP	CRYSTAL
1	GRENOUAUD Damien	13/05/2022	EMAIL			1					

Afin de mener à bien cette mission, il faut récupérer les informations qui sont présentes dans la table M2.

```
/**
 * Obtention des articles M2
 */
$reqM2 = "Select * from M2 where ID_COMMANDE = $id_commande ORDER BY NUM_FLEXO_AVENANT";
$resultatM2 = mssql_query($reqM2);
```

Il faut ensuite récupérer les données qui sont stockés dans un tableau de type query.

On entre ensuite dans une boucle (*qui servira également pour l'affichage*). On déclare une variable \$lastNumAvenant et une variable \$i juste avant l'entrée dans la boucle (*ce qui servira également pour l'affichage*).

Ensuite, on récupère les données contenues dans des variables comme ci-dessous :

```
<?php
$lastNumAvenant = "";
$i = 1;
while($resultM2 = mssql_fetch_assoc($resultatM2)){
    /*### Récupération des données issu de la requête des articles M2 */
    $idEpaisseur = $resultM2["ID_EPAISSEUR"]; //Récupération de l'id epaisseur dans $idEpaisseur
    $numUtilisateur = $resultM2["ID_UTILISATEUR"]; //Récupération de l'id utilisateur $numUtilisateur
    $idResolutionCDI = $resultM2["ID_RESOLUTION_CDI"]; //Récupération de l'id de résolution dans $idResolutionCDI
    $idMachine = $resultM2["ID_MACHINE"]; //Récupération de l'id de la machine dans $idMachine
    $idMatiere = $resultM2["ID_MATIERE"]; //Récupération de l'id de la matière dans $idMatiere
    $idArticle = $resultM2["TYPE_SUPPORT"]; //Récupération de l'id de l'article dans $article
    $dateSQL = $resultM2["DATE_M2"]; //Récupération de la date
    $getDate = dateToHTML($dateSQL); //On fait passer la date dans cette fonction
    $NumAvenant = $resultM2["NUM_FLEXO_AVENANT"]; //Récupération de l'avenant
```

Dans la requête juste au-dessus, j'ai récupéré, le numéro de l'utilisateur. Ce numéro va me servir à récupérer le nom et le prénom de l'utilisateur :

```
<?php
//Récupération du nom et prénom de l'utilisateur

$reqUtilisateur = "Select NOM_UTILISATEUR, PRENOM_UTILISATEUR from utilisateur where ID_UTILISATEUR = $numUtilisateur";
$resultatUtilisateur = mssql_query($reqUtilisateur);
$resultUtilisateur = mssql_fetch_assoc($resultatUtilisateur);
$NomUtilisateur = $resultUtilisateur["NOM_UTILISATEUR"];
$PrenomUtilisateur = $resultUtilisateur["PRENOM_UTILISATEUR"];
$Utilisateur = $NomUtilisateur." ".$PrenomUtilisateur; //Concaténation des deux variables
```

Remarque : La variable \$Utilisateur est la concaténation des variables \$NomUtilisateur et \$PrenomUtilisateur qui contiennent respectivement le nom et le prénom de l'utilisateur. On aurait pu faire la concaténation directement dans l'affichage sans la mettre dans cette variable.

Dans la première requête, j'ai récupéré l'id de l'épaisseur. J'ai besoin de récupérer le libelle de l'épaisseur.

```
//Récupération du libelle epaisseur (affiché plus bas)

$reqEpaisseur = "Select LIBELLE_EPAISSEUR from EPAISSEUR where ID_EPAISSEUR = $idEpaisseur";
$resultatEpaisseur = mssql_query($reqEpaisseur);
$resultEpaisseur = mssql_fetch_assoc($resultatEpaisseur);
```

On doit faire également le même principe pour obtenir le Libelle_Resolution_CDI avec \$idResolutionCDI ⁽¹⁾, le Libelle_Machine_Interne avec \$idMachine ⁽²⁾, le Libelle_Matiere avec \$idMatiere ⁽³⁾ et le Libelle_Article avec \$idArticle ⁽⁴⁾.

1 :

```
//Récupération du LIBELLE_RESOLUTION_CDI (affiché plus bas)

$reqResolutionCDI = "Select LIBELLE_RESOLUTION_CDI from RESOLUTION_CDI where ID_RESOLUTION_CDI = $idResolutionCDI";
$resultatResolutionCDI = mssql_query($reqResolutionCDI);
$resultResolutionCDI = mssql_fetch_assoc($resultatResolutionCDI);
```

2 :

```
//Récupération du LIBELLE_MACHINR_INTERNE (affiché plus bas)

$reqMachine = "Select LIBELLE_MACHINE_INTERNE from MACHINE_INTERNE where ID_MACHINE_INTERNE = $idMachine";
$resultatMachine = mssql_query($reqMachine);
$resultMachine = mssql_fetch_assoc($resultatMachine);
```

3 :

```
//Récupération du LIBELLE_MATIERE (affiché plus bas)

$reqMatiere = "Select LIBELLE_MATIERE from MATIERE where ID_MATIERE = $idMatiere";
$resultatMatiere = mssql_query($reqMatiere);
$resultMatiere = mssql_fetch_assoc($resultatMatiere);
```

4 :

```
//Récupération du LIBELLE_ARTICLE (affiché plus bas)

$reqArticle = "Select LIBELLE_ARTICLE from ARTICLE where ID_ARTICLE = $idArticle";
$resultatArticle = mssql_query($reqArticle);
$resultArticle = mssql_fetch_assoc($resultatArticle);
```

L'affichage :

On commence par tester si \$1 = 1. S'il est égal à 1, on peut afficher le début du tableau :

```
if($i == 1){
    ?>
    <fieldset class="BlocFiedlset3">
    <legend><b>Avenant n°<?php echo $NumAvenant;?></b></legend>
    <table style="width: 100%;">
        <tr>
            <th>Valeur M2</th>
            <th>Utilisateur</th>
            <th>Date M2</th>
            <th>Type d'Article</th>
            <th>Longueur</th>
            <th>Largeur</th>
            <th>Quantité</th>
            <th>Nombre couleur</th>
            <th>Epaisseur</th>
            <th>Résolution</th>
            <th>Matériau</th>
            <th>Machine</th>
            <th>NPF</th>
        </tr>
    <?php

    $lastNumAvenant = $NumAvenant;
}
```

On a ensuite 2 manières d'afficher les valeurs :

On affiche le résultat de la requête	<pre><tr> <td><?php echo \$resultM2["VAL_M2"]; ?> </td></pre>
On affiche une variable où se trouve les données	<pre><td><?php echo \$Utilisateur; ?></pre>

Et après ?

Dans le cas où il y a plusieurs avenants, \$i est incrémenté de 1 à chaque tour. De plus au moment où la condition \$i est vérifiée et vraie, une variable \$lastNumAvenant prend la valeur de \$NumAvenant.

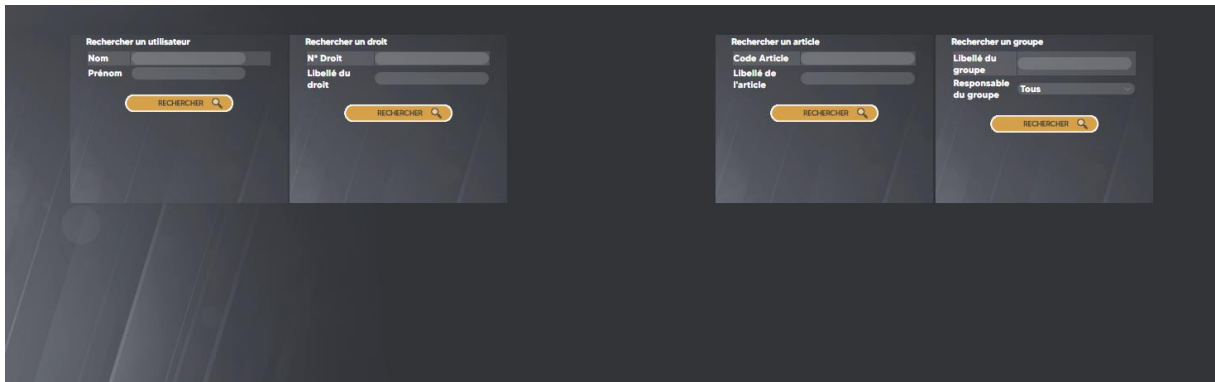
Pour afficher les potentielles autres avenants, on regarde si le \$NumAvenant est différent de \$lastNumAvenant et si \$i est supérieur à 1.

```
<?php
if( $lastNumAvenant != $NumAvenant && $i > 1){
    ?>
    </table>
    </fieldset>
    <fieldset class="BlocFieldset3">
    <legend><b>Avenant n°<?php echo $NumAvenant;?></b></legend>
    <table style="width: 100%;">
        <tr>
            <th>Valeur M2</th>
            <th>Utilisateur</th>
            <th>Date M2</th>
            <th>Type d'Article</th>
            <th>Longueur</th>
            <th>Largeur</th>
            <th>Quantité</th>
            <th>Nombre couleur</th>
            <th>Epaisseur</th>
            <th>Résolution</th>
            <th>Matériau</th>
            <th>Machine</th>
            <th>NPF</th>
        </tr>
    </table>
    <?php
    $lastNumAvenant = $NumAvenant;
}
```

Si les 2 conditions sont vérifiées, on affiche le même début de tableau. Puis on affiche les données comme vue précédemment.

Mission 7 : Refaire l'interface de recherche des différents champs Administrateur.

Avant, l'interface de recherche des différents champs Administrateur ressemblait à ça :



L'ensemble des fonctionnalités de recherche étaient dans iframe, ce qui faisait un rendu pas très beau.

J'ai ensuite mis des boutons pour relier les différentes recherches comme ceci :



Pour le design des formulaires, je l'ai via une page TPL comme ceci :



Conclusion de fin de stage :

Ce stage au sein de Flexocolor m'a permis de découvrir et d'en apprendre plus sur le développement web dans une entreprise.

Je remercie l'entreprise Flexocolor de m'avoir accueilli pendant 1 mois en stage afin de me permettre de valider ma première année ainsi que Mr MERLE pour avoir pris le temps de m'expliquer le fonctionnement de l'entreprise. Je remercie également l'ensemble du personnel de l'entreprise pour leur accueil.