

A thick dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right from this bar, containing the date. In the bottom left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

3 Janvier au 11 Février 2022

Rapport de stage

2ème année BTS SIO

LE CORVEC Thomas

I - Remerciement.....	2
II - Présentation de l'entreprise	2
III - Tâches effectuées	3
IV - Architecture logicielle et matérielle	4
V - Environnement de travail	5
VI - Gestion de projet	5
VII - Réalisation	5
1 - Contexte	5
2 – Design	6
3 - Base de données.....	6
4 – Librairies.....	6
5 - Routeur	7
6 - Organisation du code	8
7 - Code développé.....	9
Base de données	9
API WooCommerce.....	9
Navbar	11
Routeur	12
Utilisateurs et permissions	13
Paramètres.....	13
Clients / Commandes / Produits.....	14
Carte cadeau	15
Factures.....	16
Extraction	18
8 – Sécurité	19
9 - Gestion de version.....	20
VIII - Conclusion.....	20
IX - Annexes.....	20

I - REMERCIEMENT

Je tiens avant toute chose à remercier l'entreprise MLG Voyages de m'avoir ouvert ses portes ainsi que sa gérante Madame LE GAL Muriel de m'avoir accompagné tout au long de cette expérience en milieu professionnel, et d'avoir su me faire confiance dans les différentes tâches qui m'ont été confiées.

Je remercie également le lycée Notre Dame de la Paix ainsi que les professeurs de ma formation pour m'avoir permis d'effectuer ce stage enrichissant tant sur le plan professionnel que personnel

II - PRESENTATION DE L'ENTREPRISE

Dans le cadre de ma formation en BTS Services Informatiques aux Organisations option Solutions Logicielles et Applications Métiers, j'ai pu effectuer un stage au sein de l'entreprise MLG Voyages durant 6 semaines, du 3 Janvier au 11 Février 2022.

L'entreprise MLG Voyages est une agence de voyage sur mesure, sous la forme juridique d'une EURL. Elle permet aux gens de créer leur voyage sur mesure, en se basant sur leurs goûts et leurs envies. Le siège social de cette entreprise se situe au 25 rue Sombreuil à Auray. L'entreprise a été créée en 2010 par sa gérante actuelle, Madame LE GAL Muriel, mais a évolué en 2017 en changeant de statut. L'entreprise ne possède aucun employé.

La gérante Madame LE GAL Muriel possède plusieurs entreprises telles que : HappyBox, Solos&Co, Les Voyages du 6^{ème} sens, Le Petit Morbihannais ou encore Idyllik Rencontres.

Grâce à l'ensemble de ses entreprises, elle réalise du commerce de biens, à l'aide des box vendues par la marque HappyBox, ainsi que de la prestation de services avec ses autres marques, création de voyages sur mesure ou encore des offres pour son site de rencontre.

La principale identité visuelle de l'entreprise MLG Voyages est le logo de cette dernière :



III - TACHES EFFECTUEES

La principale mission qui m'a été confiée durant mon stage a été la création d'un logiciel de gestion pour 4 des marques de l'entreprise. Ce logiciel avait pour but de permettre de centraliser l'ensemble des clients / commandes des différentes marques. Il devait également permettre la création de facture au format PDF ainsi que l'extraction comptable sous forme de tableau Excel.

Durant ma première semaine, j'ai tout d'abord modifié le logiciel Zoe, qui est le logiciel sur lequel j'avais pu travaillé lors de mon précédent stage. J'ai dû y ajouter de nouvelles fonctionnalités, notamment l'ajout d'un nouveau type de facture et la différenciation entre les clients professionnels et les clients particuliers. J'ai également profité de cette semaine pour corriger certains problèmes du logiciel. Ensuite la gérante et moi avons effectué une réunion pour me permettre de comprendre les besoins du logiciel qu'elle souhaitait. Suite à cette dernière, j'ai commencé la création des maquettes pour le logiciel afin de valider le visuel des différentes pages auprès de la gérante. Une fois le visuel validé, j'ai développé les différentes pages du logiciel.

Lors de ma seconde semaine j'ai continué le développement des différentes pages (tableaux dynamiques, etc...). Puis j'ai mis en place la base de données, ainsi que le fichier modèle permettant d'interagir avec cette dernière. Suite à cela j'ai réalisé le système de création de commande depuis le logiciel, puis j'ai mis en place les différentes fonctions permettant d'interagir avec l'API WooCommerce pour pouvoir récupérer les commandes des sites. J'ai également effectué la migration d'un site sous Wordpress d'un hébergement à l'autre.

Ensuite durant ma troisième semaine, j'ai créé le système d'extraction comptable pour les différentes marques, puis j'ai mis en place la fonctionnalité de génération des factures au format PDF. J'ai également fait le système utilisateur du logiciel (identifiant / mot de passe / permissions). J'ai aussi eu l'occasion de créer un nouveau sous-domaine, et d'y mettre en place un site sous Wordpress, et de le sécuriser avec un certificat SSL.

Pour ma quatrième semaine, j'ai décidé d'optimiser les requêtes SQL ainsi que le fonctionnement général du logiciel Zoe qui subissait de très grosses lenteurs. J'ai également créé une page permettant l'activation des cartes cadeaux (vendu dans les box Happybox) ainsi que la création de ces dernières directement sous WooCommerce automatiquement.

Durant ma cinquième semaine, j'ai pu mettre en place une page « Nos commerçants » sur le site Le Petit Morbihannais, ainsi que modifier le design de certaines pages et produits. J'ai également modifié le mail de confirmation lors de

l'activation d'une nouvelle carte (Happybox). J'ai aussi pu ajouter la livraison Colissimo sur les différents sites. J'ai mis en place des partages sous Windows avec authentification, installer un nouvel ordinateur et mis en place un système de sauvegarde des partages.

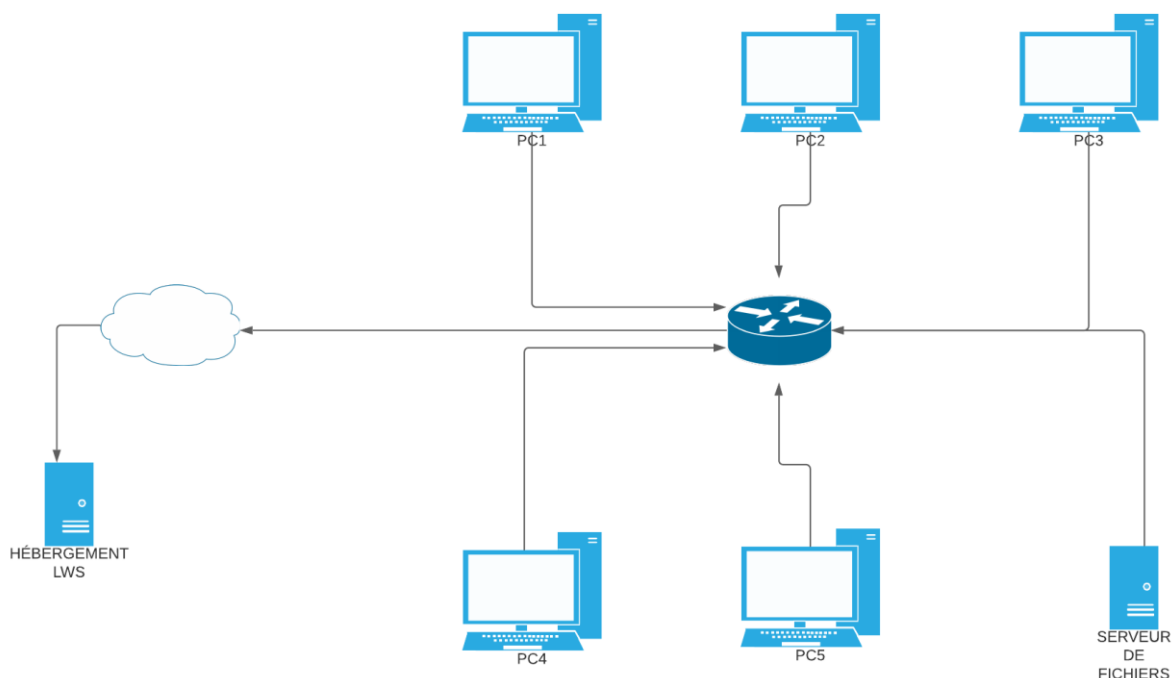
Pour ma dernière semaine, j'ai principalement ajouté / modifié des produits et des pages sur le site Happybox, et également ajout la livraison en point relais. Puis j'ai corrigé les derniers problèmes des 2 logiciels existants.

IV - ARCHITECTURE LOGICIELLE ET MATERIELLE

L'entreprise possède 5 ordinateurs ainsi qu'un serveur de fichiers. L'ensemble des sites et bases de données sont hébergés chez LWS, et accessible via un FTP pour l'hébergement du site Web ou via le site de l'hébergeur pour la base de données et le nom de domaine.

Les 5 ordinateurs ainsi que le serveur de fichiers sont sous Windows 10. Le serveur de fichiers possède 3 partages qui sont « commun » pour l'ensemble des postes, « direction » pour la gérante et « muriel » pour son partage personnel.

L'ensemble est raccordé à la fibre grâce une box Internet de chez Orange.



L'entreprise MLG Voyages possède au total 7 sites sous WordPress, ainsi que 2 logiciels en PHP et possède une vingtaine de noms de domaine.

V - ENVIRONNEMENT DE TRAVAIL

Mon environnement de travail était un ordinateur sous Windows sur lequel j'ai installé un serveur WAMP afin de pouvoir tester PHP et MYSQL directement. J'ai également installé Visual Studio Code comme éditeur de code et Postman pour permettre de tester les différentes requêtes sur l'API WooCommerce.

J'ai dû installer FileZilla afin de transférer l'ensemble des fichiers du site sur le FTP de l'hébergeur.

VI - GESTION DE PROJET

Durant mon stage, je n'ai pas utilisé de logiciel de gestion tel que Trello, car étant seul je n'en ai pas vu l'utilité.

Pour la communication avec la gérante, nous avons privilégié les réunions journalières / hebdomadaires pour pouvoir discuter plus facilement des problèmes et des besoins de la gérante concernant ce logiciel.

Avec ces réunions j'avais une meilleure compréhension des besoins ainsi qu'une remontée des problèmes très rapide, ce qui m'a permis de ne pas partir dans de mauvaises directions et de régler les problèmes très rapidement, sans être surchargé.

VII - REALISATION

1 - Contexte

Mon stage s'est concentré sur un projet principal, il s'agit de la création d'un logiciel interne pour gérer 4 des marques de l'entreprise sous un seul et même logiciel nommé « MLG ».

Ce logiciel m'a été demandé par la gérante de l'entreprise afin qu'elle puisse gérer les clients / commandes / produits des différentes marques sous ce logiciel, et qu'elle est la possibilité d'extraire les factures au format PDF et de réaliser des extractions comptables pour chacune des marques.

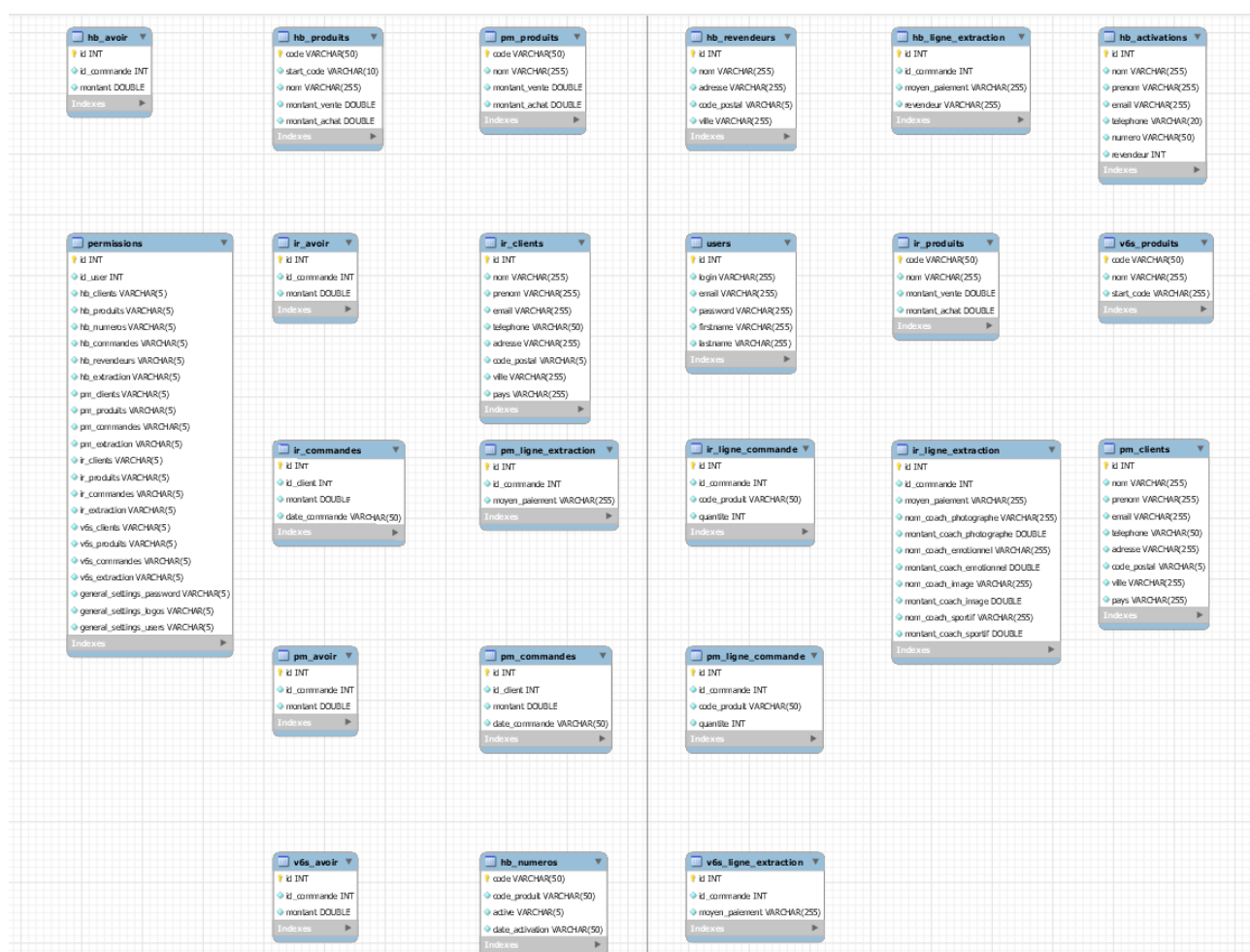
2 – Design

J’ai commencé par la création des visuels pour les différentes pages du logiciel afin de pouvoir les valider et commencer le développement au plus vite. J’ai donc proposé quelques visuels pour que la gérante est une idée du design général du logiciel.

3 - Base de données

J’ai ensuite créé la base de données après une étude rapide de l’ensemble des tables et champs nécessaires au logiciel. Puis je l’ai exporté dans un fichier au format .sql afin de permettre une restauration rapide de cette dernière.

Elle est composé de 23 tables :



4 – Librairies

Pour l’ensemble du logiciel, 2 librairies sont utilisées :

FPDF : elle permet la génération des PDF avec PHP

XLSXWriter : elle permet la génération de document Excel avec PHP

5 - Routeur

Pour ce logiciel j'ai décidé de créer un routeur central qui afficherait les pages demandées en fonction de l'URL mais où tout serait géré depuis le même fichier.

Pour ce faire j'ai donc dû créer un fichier .htaccess afin de rediriger l'ensemble des requêtes vers le fichier index.php.

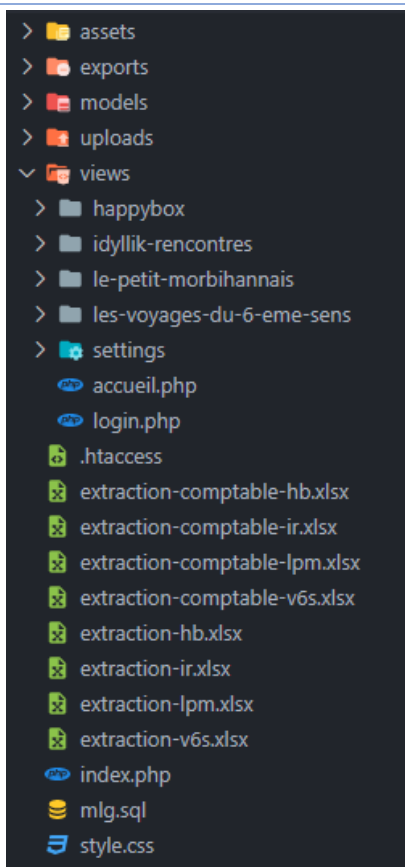
```
RewriteBase /  
Options +FollowSymLinks  
  
RewriteEngine On  
  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteRule ^(.*)$ index.php?url=$1 [NC,L]
```

Les 3 premières permettent d'activer la redirection et la réécriture des URL.

Ensuite la 4^{ème} et 5^{ème} lignes sont des conditions qui permettent de ne pas réécrire l'URL si le chemin demandé correspond à un fichier ou un dossier existant.

La dernière effectue la réécriture de l'URL de manière invisible en redirigeant la requête vers le fichier index.php et en lui passant un paramètre nommé « URL » contenant le chemin demandé.

6 - Organisation du code



Le code est séparé dans 5 dossiers distincts, le premier dossier « assets » contient toutes les images, polices d'écritures et librairies (FPDF et XLSXWriter).

Le second dossier « exports » qui contient les fichiers générés lors de la création de carte cadeau pour la marque Happybox (fichiers sous format Excel contenant l'ensemble des numéros générés).

Le troisième dossier « models » contient les 2 modèles qui interagissent avec les données, un fichier avec la base de données, et un second avec l'API WooCommerce.

Le quatrième dossier « uploads » contient les logos téléchargés sur le site directement depuis l'interface du logiciel.

Le dernier dossier est le dossier « views » qui contient les vues de l'application, c'est à dire les pages de cette dernière.

Il a ensuite 4 fichiers importants présents dans l'arborescence :

index.php qui est le fichier principal

.htaccess qui est le fichier qui protège l'ensemble des routes et permet au routeur de fonctionner

mlg.sql qui est le fichier contenant la base de données du logiciel

style.css qui contient l'ensemble du style du logiciel

D'autres fichiers commençant par « extraction » sont présents, il s'agit des fichiers générés lors de la création de l'extraction comptable.

7 - Code développé

Base de données

L'ensemble des interactions avec la base de données est géré par un seul fichier nommé « database.php ». Il contient de nombreuses fonctions permettant d'utiliser directement la base de données, sans avoir à réécrire le code à chaque fois.

Ce fichier contient tout d'abord les requêtes concernant le logiciel dans sa globalité, puis il contient les requêtes pour chaque marques.

```
function PM_getCustomerById($idClient){
    $req = $this->bdd->prepare('SELECT * FROM pm_clients WHERE id = :id');
    $req->execute(array(
        'id' => $idClient
    ));
    return $req->fetch();
}

function PM_createCustomer($nom, $prenom, $email, $telephone, $adresse, $codePostal, $ville, $pays){
    $req = $this->bdd->prepare('INSERT INTO pm_clients VALUES(0, :nom, :prenom, :email, :telephone, :adresse, :code_postal, :ville, :pays)');
    $req->execute(array(
        'nom' => $nom,
        'prenom' => $prenom,
        'email' => $email,
        'telephone' => $telephone,
        'adresse' => $adresse,
        'code_postal' => $codePostal,
        'ville' => $ville,
        'pays' => $pays
    ));
}

function PM_editCustomer($idClient, $nom, $prenom, $email, $telephone, $adresse, $codePostal, $ville, $pays){
    $req = $this->bdd->prepare('UPDATE pm_clients SET nom = :nom, prenom = :prenom, email = :email, telephone = :telephone, adresse = :adresse, code_postal = :code_postal, ville = :ville, pays = :pays WHERE id = :id');
    $req->execute(array(
        'nom' => $nom,
        'prenom' => $prenom,
        'email' => $email,
        'telephone' => $telephone,
        'adresse' => $adresse,
        'code_postal' => $codePostal,
        'ville' => $ville,
        'pays' => $pays,
        'id' => $idClient
    ));
}

function PM_deleteCustomer($idClient){
    $req = $this->bdd->prepare('DELETE FROM pm_clients WHERE id = :id');
    $req->execute(array(
        'id' => $idClient
    ));
}
```

Voici quelques exemples de fonctions disponibles. L'ensemble des fonctions respecte une convention de nommage « MARQUE_roleDeLaFonction ».

API WooCommerce

Le logiciel a également besoin d'interagir avec les différents sites Wordpress, et notamment avec l'extension « WooCommerce ». Pour ce faire j'ai donc créé un fichier « api-wordpress.php » contenant l'ensemble des interactions avec WooCommerce.

Ce fichier respecte la même convention de nommage que le fichier « database.php ».

Pour pouvoir communiquer avec l'API il a tout d'abord fallu l'activer, ainsi que récupérer les clés publiques et privées permettant de nous authentifier. J'ai donc stocké ces dernières dans des constantes.

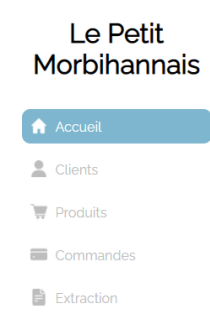
```
define('KEY_CK_HB', 'ck_564590ff2c04422bb77dbe96b7d85f00fbef82c6');  
define('KEY_CS_HB', 'cs_582f96a766a341cd3a3227a8e9c9958624b84a55');
```

Pour effectuer les requêtes vers l'API, j'ai utilisé cURL qui est intégré dans PHP.

```
function HB_getAllCustomers($nbPage){  
    $endpoint = 'https://happybox-boutique.com/wp-json/wc/v3/customers/';  
    $params = array(  
        'consumer_key' => KEY_CK_HB,  
        'consumer_secret' => KEY_CS_HB,  
        'per_page' => 20,  
        'page' => $nbPage  
    );  
    $url = $endpoint.'?'.http_build_query($params);  
  
    $cr = curl_init();  
  
    curl_setopt($cr, CURLOPT_SSL_VERIFYPEER, FALSE);  
    curl_setopt($cr, CURLOPT_RETURNTRANSFER, TRUE);  
    curl_setopt($cr, CURLOPT_URL, $url);  
    curl_setopt($cr, CURLOPT_TIMEOUT, 80);  
  
    $response = json_decode(curl_exec($cr));  
  
    return $response;  
}
```

Navbar

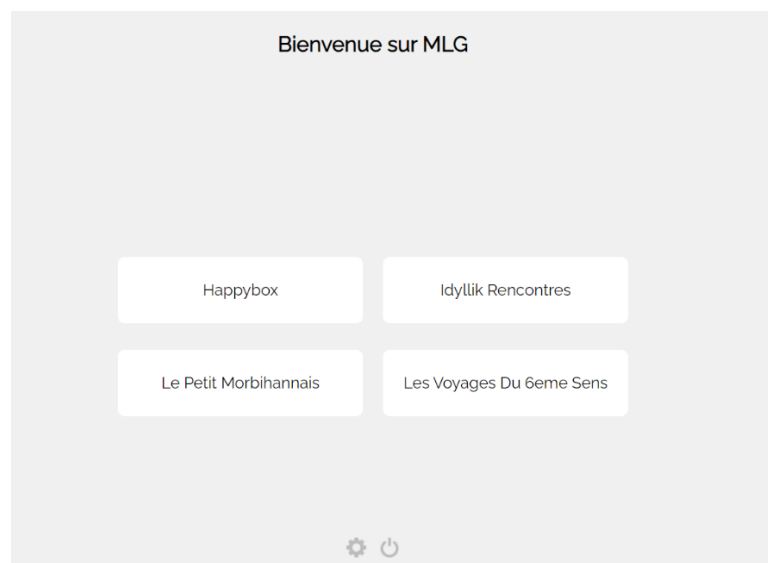
La barre de navigation diffère selon les marques mais reste semblable tout de même :



Tout d'abord le nom de la marque, puis en dessous les différents onglets disponibles pour chaque marques : Accueil, Clients, Produits, Commandes et Extraction.

Puis tout en bas les différents boutons : Retour à la page d'accueil, Paramètres et Déconnexion.

Pour naviguer entre les marques il faut se rendre la page d'accueil :



Routeur

L'ensemble des vues est affichée à l'aide du routeur contenu dans le fichier « index.php ».

Il affiche la vue correspondante au paramètre « url » contenu dans l'URL.

Voici un exemple :

```
else if($url[0] == 'settings'){
    require('views/settings/navbar.php');
    if(isset($url[1])){
        if($url[1] == 'password' && ($userPermissions['general_settings_password'] == 'on')){
            viewSettingsPassword();
        }
        else if($url[1] == 'logos' && ($userPermissions['general_settings_logos'] == 'on')){
            viewSettingsLogos();
        }
        else if($url[1] == 'users' && ($userPermissions['general_settings_users'] == 'on')){
            if(isset($url[2])){
                if($url[2] == 'create'){
                    viewSettingsUserCreate();
                }
                else if($url[2] == 'edit'){
                    viewSettingsUserEdit();
                }
                else if($url[2] == 'delete'){
                    functionSettingsUserDelete();
                }
                else{
                    viewSettingsUsers();
                }
            }
            else{
                viewSettingsUsers();
            }
        }
        else{
            viewSettings();
        }
    }
    else{
        viewSettings();
    }
}
```

Le paramètre « url » est séparé entre chaque « / » puis chaque partie est testée une à une pour pouvoir afficher la bonne page, ainsi que la bonne barre de navigation.

Comme vous pouvez voir le routeur vérifie les permissions de l'utilisateur actuel.

Utilisateurs et permissions

A partir du compte « admin » il est possible de créer d'autres utilisateurs depuis les paramètres :

Paramètres
[Modifier le mot de passe](#)
[Modifier les logos](#)
[Utilisateurs](#)

+ Utilisateur

Nom	Prénom	Ville	Actions
admin	direction@envie-de-voyages.com	Admin Admin	 
Aziliz	commercial@envie-de-voyages.com	Pasco Aziliz	 

En cliquant sur le bouton « + Utilisateur » :

Identifiant : *

Email : *

Mot de passe : *

Prénom : *

Nom : *

Permission	Active
Général - Paramètres - Mot de passe	<input type="checkbox"/>
Général - Paramètres - Logos	<input type="checkbox"/>
Général - Paramètres - Utilisateurs	<input type="checkbox"/>
HB - Clients	<input type="checkbox"/>
HB - Produits	<input type="checkbox"/>
HB - Numéros	<input type="checkbox"/>
HB - Commandes	<input type="checkbox"/>
HB - Revendeurs	<input type="checkbox"/>
HB - Extraction	<input type="checkbox"/>
PM - Clients	<input type="checkbox"/>
PM - Produits	<input type="checkbox"/>

On peut y renseigner les différentes informations de l'utilisateur, mais surtout y définir ses permissions. Ces mêmes permissions qui sont ensuite vérifiées par le routeur.

Lorsqu'un utilisateur n'a pas une permission, l'onglet en lien avec cette dernière n'apparaît plus dans la barre de navigation de l'utilisateur.

La gérante souhaitait cette fonctionnalité pour lui permettre de créer des comptes pour ses collaborateurs tout en leur restreignant leur accès.

Paramètres

Les paramètres sont composés de 3 onglets :

- Utilisateurs comme nous venons de le voir
- Mot de passe : Permettant de modifier le mot de passe du compte utilisateur actuel
- Logos : Permettant de modifier les logos visibles sur les factures de chaque marques

Clients / Commandes / Produits

L'ensemble de ces éléments ont été développé sur le même principe : Ajout / Modification et Suppression. Il s'agit de simples pages avec des champs et qui engendrent des interactions avec la base de données.

Voici un exemple :

```
<head>
  <title><?php echo $title; ?></title>
</head>

<div class="page-content">

  <div class="header">
    <h3>Le Petit Morbihannais > Produits > Nouveau</h3>
  </div>

  <div class="box">

    <form method="POST" action="/le-petit-morbihannais/produits/create/">

      <div>
        <label for="code">Code : *</label>
        <input id="code" type="text" placeholder="Code" name="code" required/>
      </div>

      <div>
        <label for="nom">Nom : *</label>
        <input id="nom" type="text" placeholder="Nom" name="nom" required/>
      </div>

      <div>
        <label for="montantVente">Montant vente : *</label>
        <input id="montantVente" type="text" placeholder="Montant vente" name="montantVente" required/>
      </div>

      <div>
        <label for="montantAchat">Montant achat : *</label>
        <input id="montantAchat" type="text" placeholder="Montant achat" name="montantAchat" required/>
      </div>

      <button type="submit" name="createProduit">Créer</button>
    </form>

  </div>

</div>
```

Puis l'interaction avec la base de données :

```
//Produits
function viewLePetitMorbihannaisProduitsCreate(){
  $title = "MLG | Le Petit Morbihannais | Créer produit";
  if(isset($_POST['createProduit'])){
    $database = $GLOBALS['database'];

    $code = trim($_POST['code']);
    $nom = trim($_POST['nom']);
    $montantVente = trim($_POST['montantVente']);
    $montantAchat = trim($_POST['montantAchat']);

    if($code != '' && $nom != '' && $montantVente != '' && $montantAchat != ''){
      $database->PM_createProduct($code, $nom, $montantVente, $montantAchat);
      header('Location: /le-petit-morbihannais/produits/');
    }
  }
  else{
    require('views/le-petit-morbihannais/produits/create-produit.php');
  }
}
```

Carte cadeau

Pour la marque Happybox, la gérante avait besoin de pouvoir savoir quand était activée une carte cadeau, et de la rendre utilisable par le client et cela de manière automatique.

Pour cela j'ai donc ajouté un onglet sur le logiciel pour lui permettre de générer des numéros pour les cartes cadeaux selon les gammes qu'elle vendait. Puis une fois cela fait, j'ai mis en place une page pour pouvoir activer sa carte cadeau.

Félicitations vous venez d'acquérir une Happy Box

Afin d'en profiter, veuillez activer celle-ci en remplissant les champs suivants

Une fois celle-ci validée par nos services, vous pourrez faire votre choix sur notre site et obtenir votre bon cadeau à remettre auprès du partenaire que vous aurez sélectionné.

Activer mon Happybox

Nom : *

Prenom : *

Email : *

Téléphone : *

Lieu d'achat : (optionnel) ----- Sélectionner un lieu d'achat ----- ▼

Code de la Happybox : *

* champ obligatoire

Activer mon
Happybox

Une fois les différentes informations renseignées, la page vérifie dans la base de données du logiciel si le code existe bien et qu'il n'a pas encore été activé. Une fois cela fait, si tout est bon la page envoie un mail au client pour lui confirmer l'activation, puis le redirige vers le site contenant l'ensemble des prestations disponibles.

```
$subject = 'Activation';
$headers = "From: activation@happybox-boutique.com\r\n";
$headers .= "MIME-Version: 1.0\r\n";
$headers .= "Content-Type: text/html; charset=UTF-8\r\n";

$message = "Madame, Monsieur, ".$nom."<br /><br />";
$message .= "Votre carte cadeau Happy Box a bien été validé.<br />";
$message .= "Vous pouvez désormais faire votre sélection dans la catégorie correspondant à votre carte. <br /><br />";

$message .= "Vous ne souhaitez pas choisir immédiatement, pas de soucis, reconnecter vous quand vous serez décidé sur";
$message .= "<center><a href='https://prestations.happybox-boutique.com'>https://prestations.happybox-boutique.com</a></center><br /><br />";

$message .= "<u><b>Voici la marche à suivre :</b></u><br />";
$message .= "Une fois votre choix effectué, rajouter la prestation dans votre panier, y intégrer votre numéro de carte";
$message .= "Insérer vos coordonnées, une fois le tout validé vous recevrez par mail votre bon à remettre de votre pre";
$message .= "Contactez-le (s) prestataire (s) pour réserver.<br />";
$message .= "Le Jour J, présentez votre bon Happy Box au prestataire de l'offre sélectionnée.<br /><br /><br />";
$message .= "<center><i><b>Profitez pleinement de votre moment privilégie !</b></i><br /><br /><br />";
$message .= "L'équipe d'Happy Box<br />";
$message .= "<center>06 95 53 22 23<br /><a href='mailto:contact@happybox-boutique.com'>contact@happybox-boutique.com</a><br /><a href='https://www.happybox-boutique.com'>www.happybox-boutique.com</a></center>";
$message .= "<img width='100px' src='https://activer.happybox-boutique.com/happy.png' /></center>";
mail($email, $subject, $message, $headers);

header('Location: https://activer.happybox-boutique.com/confirm');
```


Au même que l'envoi du mail au client, la page émet une requête vers l'API WooCommerce du site Happybox en créant un code promo d'un montant correspondant à la gamme de la carte cadeau.

```
$endpoint = 'https://prestations.happybox-boutique.com/wp-json/wc/v3/coupons/';
$params = array(
    'consumer_key' => KEY_CK_PHB,
    'consumer_secret' => KEY_CS_PHB,
);
$url = $endpoint.'?'.http_build_query($params);











$data = array(
    'code' => $_POST['code'],
    'discount_type' => 'fixed_cart',
    'amount' => $resultPrice['montant_vente'],
    'free_shipping' => TRUE,
    'usage_limit' => 1,
    'limit_usage_to_x_items' => 1,
    'minimum_amount' => $resultPrice['montant_vente']
);

$cr = curl_init();

curl_setopt($cr, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($cr, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($cr, CURLOPT_URL, $url);
curl_setopt($cr, CURLOPT_POST, TRUE);
curl_setopt($cr, CURLOPT_POSTFIELDS, $data);
curl_setopt($cr, CURLOPT_TIMEOUT, 80);

curl_exec($cr);
curl_close($cr);
```

Dans le logiciel les cartes cadeaux et leur état sont récupérées depuis la base de données :

Code	Produit	Activation	Date activation	Revendeur	Actions
HAGL0003	Happy Glas	●	19/02/2022		 
HAGL0004	Happy Glas	●			 
HAGL0005	Happy Glas	●			 
HAGL0008	Happy Glas	●	15/02/2022		 
HAGL0009	Happy Glas	●			 

Factures

La gérante souhaitait pouvoir générer des factures directement depuis le logiciel pour n'importe quelle commande. J'ai donc installé la librairie FPDF (que j'avais utilisé lors de mon précédent stage).

Pour la création des factures il fallait pouvoir afficher l'ensemble des produits de la commande, le montant de livraison, le montant de l'avoir, la TVA, etc...

Voici une partie du code de la facture, cette partie du code permettait la génération du tableau de produits :

```


$facture->SetFont('Arial','B',9);
$facture->SetFillColor(171,171,171);
$facture->Ln(20);
$facture->SetX(15);
$facture->Cell(20,7,utf8_decode('Référence'),1,0,'L',true);
$facture->Cell(80,7,utf8_decode('Description'),1,0,'L',true);
$facture->Cell(20,7,utf8_decode('PU HT'),1,0,'R',true);
$facture->Cell(20,7,utf8_decode('Quantité'),1,0,'R',true);
$facture->Cell(20,7,utf8_decode('Montant HT'),1,0,'R',true);
$facture->Cell(20,7,utf8_decode('Taux TVA'),1,0,'R',true);

//Columns
$facture->SetX(15);
$facture->Cell(20,123,utf8_decode(''),1,0,'L');
$facture->Cell(80,123,utf8_decode(''),1,0,'L');
$facture->Cell(20,123,utf8_decode(''),1,0,'R');
$facture->Cell(20,123,utf8_decode(''),1,0,'R');
$facture->Cell(20,123,utf8_decode(''),1,0,'R');
$facture->Cell(20,123,utf8_decode(''),1,0,'R');

//Content
$totalMontantHT = 0;
$facture->SetFont('Arial','',9);
$facture->Ln(7);
foreach($commande as $line){
    $facture->SetX(15);
    $montantHT = round($line['montant_vente']/1.2,2);
    $totalMontantHT += $montantHT;
    $facture->Cell(20, 10, utf8_decode($line['code_produit']),0,0,'L');
    $facture->Cell(80, 10, utf8_decode($line['nom']),0,0,'L');
    $facture->Cell(20, 10, utf8_decode(number_format($montantHT, 2, '.', ' ')),0,0,'R');
    $facture->Cell(20, 10, utf8_decode($line['quantite']),0,0,'R');
    $facture->Cell(20, 10, utf8_decode(number_format($line['quantite']*$montantHT, 2, '.', ' ')),0,0,'R');
    $facture->Cell(20, 10, utf8_decode('20.00%'),0,0,'R');
    $facture->Ln(10);
}

```

Et voici le rendu d'une facture :

LE PETIT MORBIHANNAIS 		Facture N° FPM000013 Date : 15/02/2022											
LE PETIT MORBIHANNAIS 25 rue Sombreuil 56400 Auray Téléphone : 06.95.53.22.23 / 09.97.50.83.76		Nom / Prénom : LE CORVEC Thomas Adresse : 1 Rue Branly 56400 Auray Code client : 01LECORV											
Référence	Description	PU HT	Quantité	Montant HT	Taux TVA								
70722000	1/4 Page	166.67	2	333.34	20.00%								
70722002	Page entière	625.00	1	625.00	20.00%								
70722003	Page de couverture	1 000.00	1	1 000.00	20.00%								
Échéance : 15/02/2022		€ HT 1 791.67											
Règlement :		<table border="1"> <tr> <td rowspan="2">TOTAL €</td> <td>HT</td> <td>1 791.67</td> </tr> <tr> <td>TTC</td> <td>2 150.00</td> </tr> </table>				TOTAL €	HT	1 791.67	TTC	2 150.00			
TOTAL €	HT	1 791.67											
	TTC	2 150.00											
Le Petit Morbihannais vous remercie pour votre confiance		<table border="1"> <tr> <td>TVA à 5.5% :</td> <td></td> </tr> <tr> <td>TVA à 10% :</td> <td></td> </tr> <tr> <td>TVA à 20% :</td> <td>358.33</td> </tr> <tr> <td>TOTAL TVA :</td> <td>358.33</td> </tr> </table>				TVA à 5.5% :		TVA à 10% :		TVA à 20% :	358.33	TOTAL TVA :	358.33
TVA à 5.5% :													
TVA à 10% :													
TVA à 20% :	358.33												
TOTAL TVA :	358.33												

Extraction

L'un des objectifs de ce logiciel était de pouvoir générer une extraction comptable pour la comptable de l'entreprise afin de lui faciliter la tâche.

Après donc de nombreux appels avec la comptable sur ses besoins, nous avons réussi à faire un tableau Excel qui lui correspondait. Mais cela ne correspondait pas aux besoins de la gérante, j'ai donc décidé de créer 2 extractions comptables différentes.

Pour générer les fichiers Excel j'ai donc utilisé la librairie `PHPExcel`.

J'ai donc ajouté cette fonctionnalité au logiciel, en incluant la possibilité de modifier certaines informations pour chaque commande.

Il suffisait ensuite simplement de générer l'extraction entre 2 dates (avec la possibilité de modifier les informations en sélectionnant les commandes en bas) :

Date début :

Date fin :

Extraire (Comptable)

Extraire (Murel)

Client	Date	Nombre de produits	Montant	Actions
LE CORVEC Thomas	14/02/2022	3	2350€	

Voici une partie du code permettant de générer le fichier Excel :

```
$writer = new XLSXWriter();
$writer->writeSheetHeader('Extraction', $header, $styleHeaders);

foreach($listeCommandes as $commande){
    $client = $database->PM_getCustomerById($commande[0]['id_client']);
    $codeJournal = "LPM";
    $zeros = "000";
    $numeroPiece = 'FPM'.str_pad($commande[0]['id'], 6, '0', STR_PAD_LEFT);
    $date = $commande[0]['date_commande'];

    $numeroCompte = '01'.str_replace(" ", "", (trim($client['nom'])));
    $accent = array("é", "è", "à");
    $replaceAccent = array("e", "e", "a");
    $numeroCompte = str_replace($accent, $replaceAccent, $numeroCompte);
    $numeroCompte = strtoupper($numeroCompte);
    $numeroCompte = substr($numeroCompte, 0, 8);

    $libelle = $client['nom'] . " " . $client['prenom'];

    $montantTotal = $commande[0]['montant'];

    $dataClient = array(
        $codeJournal,
        $zeros,
        $numeroPiece,
        $date,
        $numeroCompte,
        $libelle,
        "",
        $montantTotal,
        ""
    );

    $writer->writeSheetRow('Extraction', $dataClient, $styleData);
}
```

Et voici le rendu :

	A	B	C	D	E	F	G	H	I
1	Code Journal		N° de pièce	Date	Code	Libellé	Quantité	Prix de vente	Montant
2	LPM	000	FPM000007	15/02/2022	01LECORV	LE CORVEC Thomas		2 350,00 €	
3	LPM	000	FPM000007	15/02/2022	70722000	LE CORVEC Thomas	2		333,33 €
4	LPM	000	FPM000007	15/02/2022	70722002	LE CORVEC Thomas	1		625,00 €
5	LPM	000	FPM000007	15/02/2022	70722003	LE CORVEC Thomas	1		1 000,00 €
6	LPM	000	FPM000007	15/02/2022	44573000	LE CORVEC Thomas			470,00 €
7									

8 – Sécurité

Je me suis également intéressé à la sécurité du logiciel, car il s'agissait de données comptables, et donc confidentielles.

L'une des premières sécurités que j'ai mis en place, était le blocage de l'injection SQL, et cela c'est fait dès la création du logiciel en utilisant les fonctions proposées par PDO de PHP pour passer des paramètres aux requêtes SQL :

```
function connect($login, $password){
    $req = $this->bdd->prepare('SELECT * FROM users WHERE login = :login AND password = :password');
    $req->execute(array(
        ':login' => $login,
        ':password' => $password
    ));
    $result = $req->fetch();
    return $result;
}
```

La seconde sécurité que j'ai mis en place a été la protection contre les utilisateurs non-identifiés. Cette solution a été mise rapidement en place car l'ensemble des routes est géré par le même fichier :

```
if(!isset($_SESSION['id']) && ((isset($_url[0]) && $_url[0] != 'login') || !isset($_url[0]))){
    header('Location: /login');
}
```

Puis la dernière sécurité, a été la protection des mots de passes à l'aide des fonctions de chiffage proposées par PHP, j'ai ici utilisé l'algorithme MD5 pour chiffrer les mots de passes :

```
if(isset($_POST['connect-btn'])){
    $database = $GLOBALS['database'];

    $login = htmlspecialchars($_POST['login']);
    $password = htmlspecialchars($_POST['password']);

    if(trim($login) != '' && trim($password) != ''){
        $password = md5($password);
        $result = $database->connect($login, $password);
        if($result){
            $_SESSION['id'] = $result['id'];
            $_SESSION['login'] = $result['login'];
            header('Location: /');
        }
        else{
            $err_msg = 'Identifiant ou Mot de passe incorrect !';
        }
    }
}
```

9 - Gestion de version

Durant la réalisation du projet, étant tout seul, j'ai préféré éviter l'utilisation d'une solution de versioning telle que Git, même si cela aurait pu m'être utile. J'ai préféré faire de simple sauvegarde sur une clé USB et sur le serveur de fichiers. Cela m'a également permis de pouvoir travailler dessus même hors connexion.

VIII - CONCLUSION

Au cours de ce stage, j'ai pu finir et réaliser entièrement ce projet, qui je l'espère a donné entière satisfaction à la gérante. Malgré les problèmes et les fonctionnalités apparus au cours du projet, j'ai eu le temps de finir entièrement ce dernier dans les temps.

Ce stage m'aura beaucoup apporté sur le plan professionnel mais également sur le plan personnel, en me permettant d'évoluer techniquement, d'acquérir de nouvelles compétences, telles que la gestion d'un projet au sein d'une entreprise, le développement et son organisation au sein de l'entreprise.

Ce stage m'a également permis de développer mon esprit d'équipe et mon esprit critique, mais également mon sens des responsabilités et de l'organisation.

IX - ANNEXES

Document	Rôle
Documentation MLG.pdf	Documentation utilisateur du logiciel MLG
Documentation Technique MLG.pdf	Documentation technique du logiciel MLG
Procédure Partages.pdf	Procédure expliquant l'ajout de partages existants sur un ordinateur.