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. Below the banner, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left corner.

25 Mai au 25 Juin 2021

Rapport de stage

1ère 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	6
1 - Contexte	6
2 - Base de données.....	7
3 - Arborescence	9
4 - Organisation du code	10
5 - Code développé.....	11
Calcul automatique du prix du contrat.....	11
Fonction pour rechercher dans les <select>.....	12
Mise à jour automatique du contenu des <select>	13
Tableau pour les suppléments pour les devis	14
Génération du PDF pour les convocations	16
Archivage des éléments	17
Extraction comptable (génération Excel)	19
6 - Sécurité	23
7 - Gestion de version	24
VIII - Conclusion.....	24
IX - Annexes.....	24

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 confié.

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 5 semaines, du 25 Mai au 25 Juin 2021.

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 tel que : HappyBox, Solos&Co, Les Voyages du 6^{ème} sens 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

Durant mon stage, j'ai tout d'abord dû faire évoluer et modifier un site sous WordPress afin de lui rajouter un système de cagnotte en ligne, modifier la mise en page des mails, ainsi que rajouter des pays pour l'agence de voyage. A la suite de ces différentes modifications, j'ai réalisé 2 procédures pour faciliter l'utilisation futur pour la gérante ainsi que pour ses futurs collaborateurs.

Dans un seconde temps, à partir de la deuxième semaine, j'ai commencé à récupérer le code du logiciel interne à l'entreprise permettant la gestion des devis/contrats/clients/partenaires, etc... Une fois m'être familiarisé avec le code de ce dernier, nous avons effectué une réunion avec la gérante afin de comprendre et prendre note de ses différents besoins. Nous avons ensuite réparti les différentes tâches entre nous via l'outil Trello.

Durant cette seconde semaine, j'ai modifié et ajouté des fonctionnalités sur la partie des devis notamment la récupération des données des devis lors de la modification ou encore la possibilité de modifier les dates, j'ai également modifier l'affichage PDF de ce dernier.

Lors de ma troisième semaine, je suis passé sur la partie des contrats, en modifiant leur affichage PDF, en leur ajoutant une fonctionnalité de calcul automatique des prestations (qui étaient jusque là, calculés par la gérante). J'ai également modifier les factures afin de les faire se mettre à jour de manière automatique (calcul du solde, date de règlement, etc...) ainsi que la possibilité d'indiquer quand une facture a été réglée. J'ai aussi ajouté la possibilité d'ajouter des contacts pour les différents partenaires de l'agence, ajouter la possibilité de créer de nouvelles assurances et mis en place des numéros uniques pour les factures, contrat et devis (qui sont obligatoires par la loi). Grâce à ces numéros uniques, j'ai pu mettre en place une fonctionnalité de recherche dans les contrats, les devis et les factures. Suite à l'ensemble de ces modifications, j'ai mis le logiciel de nouveau en ligne afin de permettre à la gérante de pouvoir regarder notre travail.

Ensuite lors de la quatrième semaine, à la suite de la réunion avec la gérante, j'ai mis en place un système de duplication des contrats et des devis pour faciliter la saisie de ces derniers. J'ai aussi pu mettre en place un système d'archivage des contrats/devis/convocations/factures/bons d'échanges afin que les listes ne soit pas surchargées par d'ancien éléments. J'ai ensuite mis en place un système de tableau dynamique afin de pouvoir ajouter des suppléments sur les devis. Durant cette semaine j'ai également fait de la maintenance informatique, en révisant/mettant à jour et analysant les ordinateurs, et en mettant en place un serveur de fichiers commun.

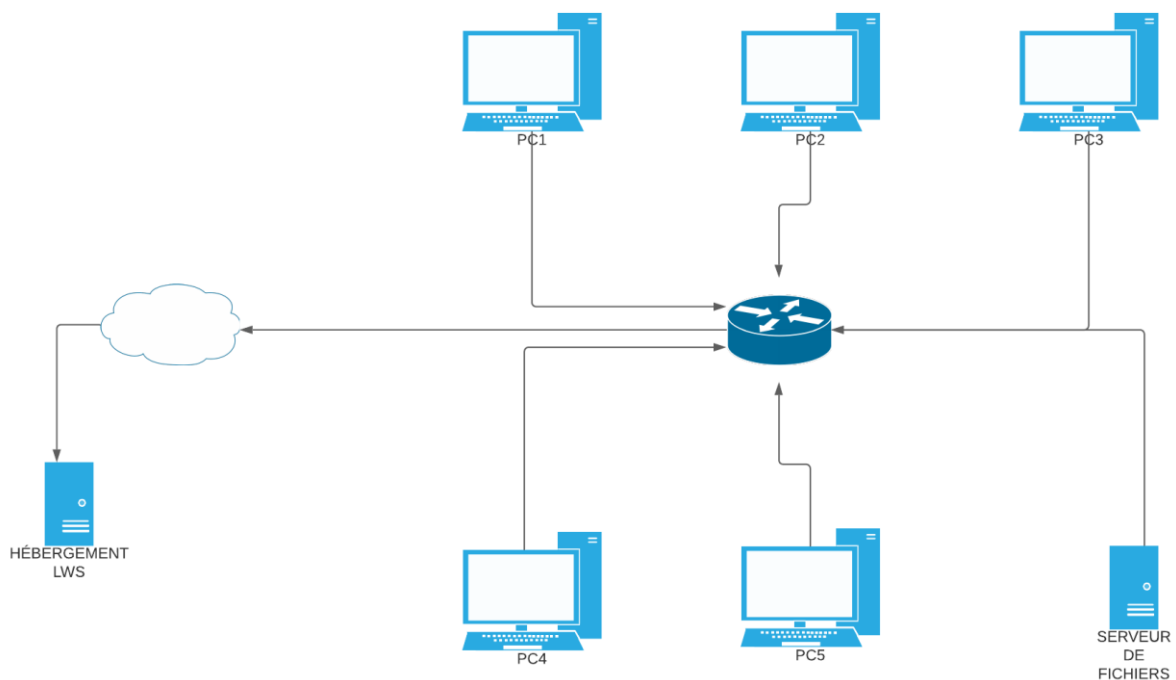
Lors de ma dernière semaine, j'ai pu mettre en place un système d'extraction comptable pour faciliter la gestion administrative et comptable de la gérante, afin d'extraire les différents contrats ainsi que les montants totaux des ventes/marges/TVA. J'ai également corrigé de nombreux petits problèmes présents au sein de logiciel (impossibilité d'ajouter des escales ou encore différents problèmes d'affichage sur les PDF). J'ai également authentifié des domaines sur le logiciel OxiMailing, et installer des systèmes de sauvegardes sur les différents sites WordPress de l'entreprise.

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 l'hébergeur 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 un seul partage qui est commun pour l'ensemble des postes. Un second partage est présent mais sur l'ordinateur de la gérante, il s'agit d'un partage concernant la direction de l'entreprise.

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



L'entreprise MLG Voyages possède au total 5 sites sous WordPress, ainsi qu'un logiciel en PHP et possède au total 12 noms de domaines.

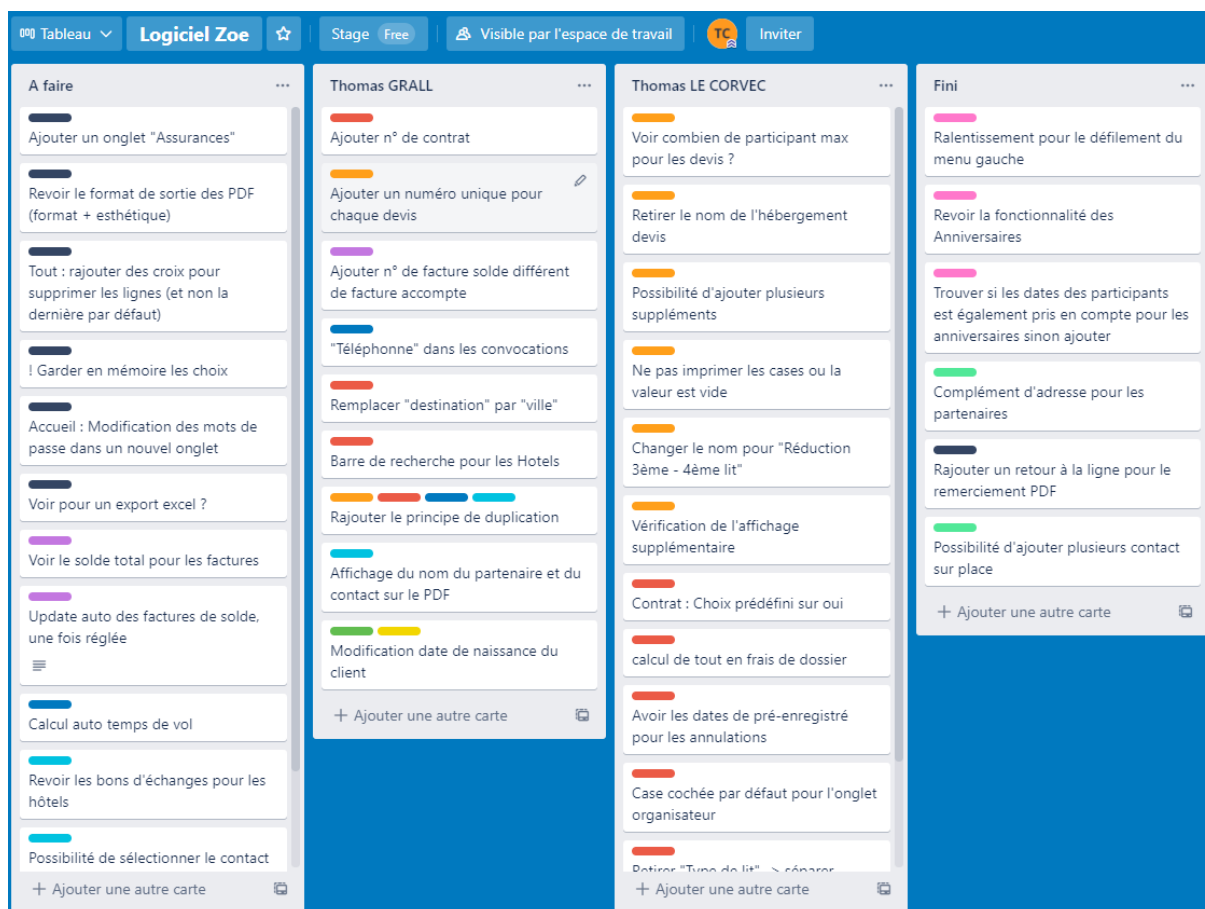
V - ENVIRONNEMENT DE TRAVAIL

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.

J'ai également 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, j'ai eu l'occasion d'utiliser une démarche de gestion de projet avec Trello afin de pouvoir nous répartir les tâches ensemble concernant le logiciel demandé par la gérante.



Ce trello nous a servi à suivre l'avancement du projet et de pouvoir se répartir les tâches. 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 journalières ou hebdomadaires nous avons une meilleure compréhension des besoins ainsi qu'une remontée des problèmes très rapide, ce qui nous a permis de ne pas partir dans de mauvaises directions et de régler les problèmes très rapidement, sans être surchargés.

VII - REALISATION

1 - Contexte

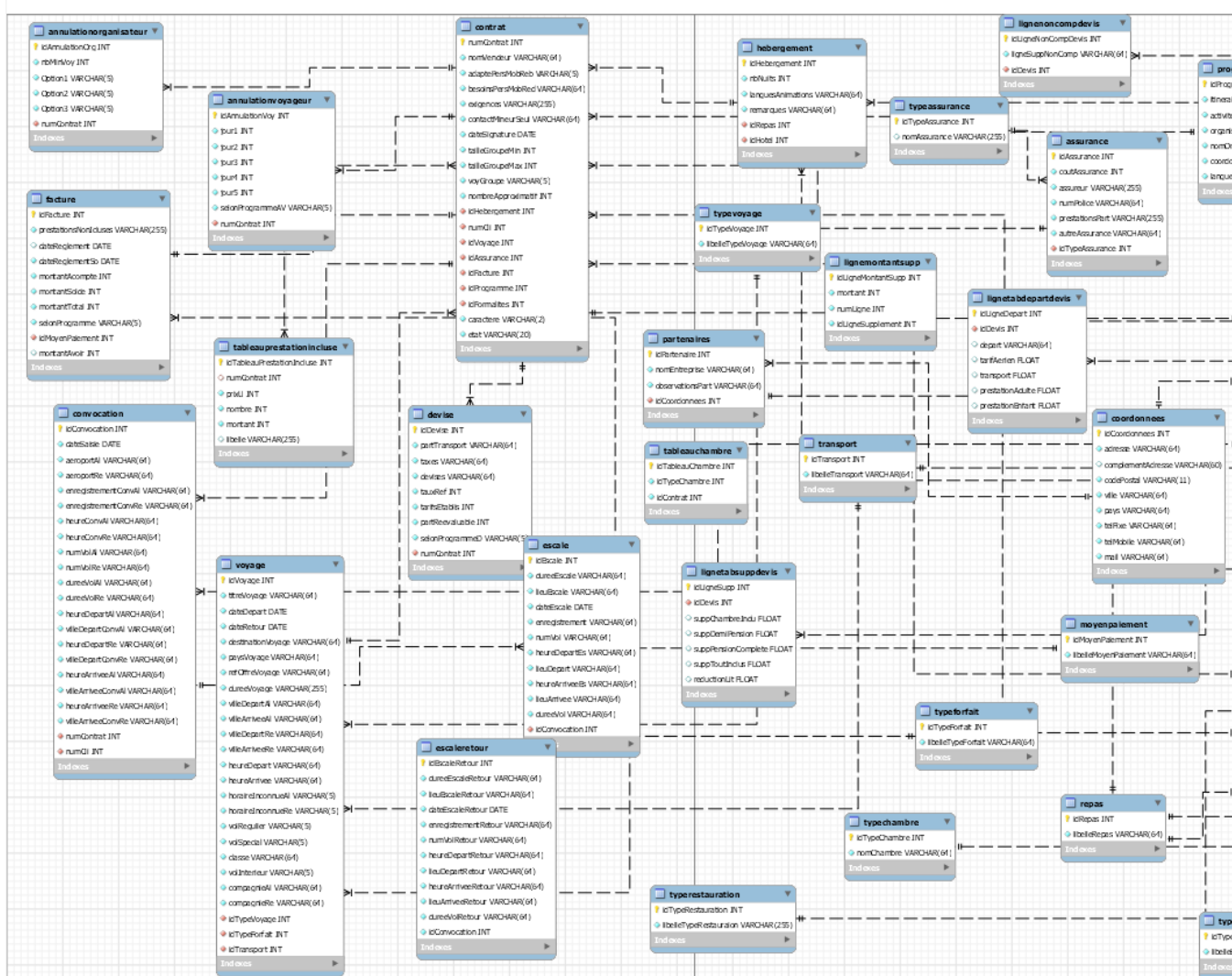
Mon stage s'est concentré sur un projet principal, qu'est le logiciel interne de l'agence, nommé « Zoé ». Ce logiciel a pour but de gérer la saisie, la modification et l'impression des devis/contrats/convocations/factures de l'agence, ainsi que la gestion de ses différents clients, partenaires, contacts, et l'extraction des données comptables.

Ce logiciel a été réalisé par plusieurs stagiaires durant les années passées, et a été repris par nos soins. Nous avons dû mettre en place de nouvelles fonctionnalités telles que des calculs automatiques, des fonctions de recherche, de l'extraction ou encore de l'archivage mais également régler des problèmes existants, comme des problèmes d'affichages de contrats, devis ou autres, et des problèmes de récupération de données.

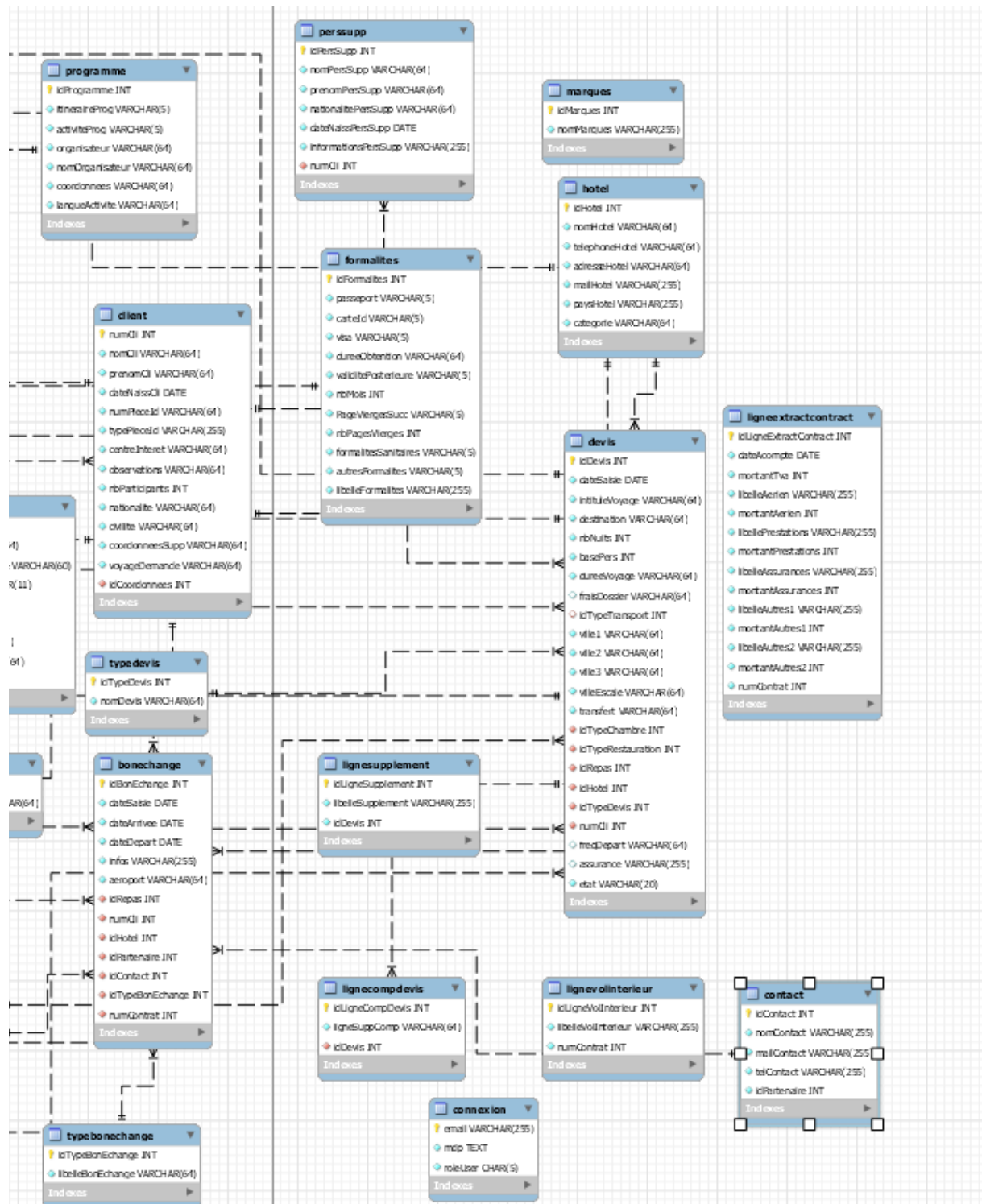
2 - Base de données

Nous avons tout d'abord récupéré la base de données, que nous avons ensuite faite évoluer selon les besoins de la gérante, pour avoir finalement la base de données suivante :

1^{ère} partie :



2^{de} partie :



3 - Arborescence

Le logiciel est basé sur un fichier index principal, où sont ensuite inclus les différentes vues des pages en fonction d'un paramètre « state » passé au travers de l'URL. Une barre de navigation latéral, permet de se déplacer à travers les différentes pages du site, la voici :

Accueil
Clients
Devis
Contrats
Factures
Convocations
Bons d'échanges
Moyens de paiement
Partenaires
Hotels
Assurance
Archives
Extraction
Changer mot de passe
Déconnexion

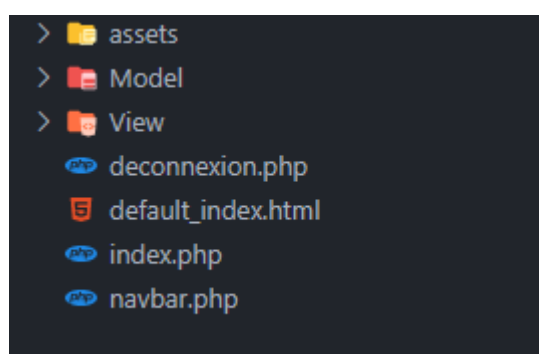
Elle est constituée de différents menus et sous-menus tel que :

- Un onglet pour les clients/participants et les différentes actions associées (Création/Modification)
- Un onglet pour les devis et les différentes actions associées (Création/Modification/Affichage)
- Un onglet pour les contrats et les différentes actions associées (Création/Modification/Affichage)

- Un onglet pour les factures et les différentes actions associées (Règlement et édition d'avoir)
- Un onglet pour les convocations et les différentes actions associées (Création/Modification/Affichage)
- Un onglet pour les bons d'échanges et les différentes actions associées (Création/Modification/Affichage)
- Un onglet pour les moyens de paiement et leur création
- Un onglet pour les partenaires et les différentes actions associées (Création/Modification/Ajout de contacts)
- Un onglet pour les hôtel et les différentes actions associées (Création/Modification)
- Un onglet pour les assurances et les différentes actions associées et leur création
- Un onglet pour les archives et les différentes actions associées (Archivage/Affichage des archives)
- Un onglet pour l'extraction et les différentes actions associées (Extraction/Modification des lignes d'extractions)
- Un onglet pour la modification de mot de passe et un pour la déconnexion

4 - Organisation du code

Voici l'organisation du code :



Le code est séparé dans 3 dossiers distincts, le premier dossier « assets » contient toutes les images, librairies (FPDF, pour la génération des PDF et XLSXWriter pour la génération des fichiers Excel), ainsi que les fichiers CSS. Le second dossier « Model » qui contient le fichier model.php qui est le fichier qui interagit avec la base de données. Le dernier dossier est le dossier « View » qu

contient les vues de l'application, c'est à dire les pages de cette dernière. Il a ensuite 4 fichiers présents dans l'arborescence :

index.php qui est le fichier principal

deconnexion.php qui est le fichier qui exécute le script de déconnexion

navbar.php qui contient la barre latéral de navigation

default_index.html qui est la page par défaut de l'hébergeur

Un autre fichier est présent, mais uniquement sur l'hébergeur, il s'agit du fichier .htaccess, il s'agit d'un fichier de configuration (Apache) qui permet de définir des règles dans ce répertoire, comme la redirection, ou encore la protection par mot de passe du répertoire.

5 - Code développé

Calcul automatique du prix du contrat

Lors de la saisie des contrats le principal problème était le fait que l'utilisateur devait calculer lui même le prix des prestations, et perdait donc un temps considérable. J'ai donc fais en sorte que l'ensemble des prestations soient calculées automatiquement.

Le tout en Javascript :

```
//Fonction qui calcule le montant de chaque ligne de prestation (montant*nombre)
function calculLine(lineNumber){
    var cellPrixUnitaire = document.getElementById("cellPrixUnitaire"+lineNumber);
    var cellNombre = document.getElementById("cellNombre"+lineNumber);
    var cellMontant = document.getElementById("cellMontant"+lineNumber);

    var prixUnitaire = (cellPrixUnitaire.value) ? cellPrixUnitaire.value : 0;
    var nombre = (cellNombre.value) ? cellNombre.value : 0;

    cellMontant.value = prixUnitaire*nombre;

    calculTotal();
}

//Fonction qui calcule le montant de l'ensemble des lignes de prestation
function calculTotal(){
    var montantTotalInput = document.getElementById("montantTotalPrestations");
    var montantTotal = 0;

    for(var i = 1; i <= nbLigneTab; i++){
        var cellMontant = document.getElementById("cellMontant"+i);
        montantTotal += parseInt(cellMontant.value);
    }

    montantTotalInput.value = montantTotal;
}
```

Voici le rendu du code, le montant de chaque ligne est calculée, et l'ensemble de ces montants est additionné pour obtenir le montant total :

PRESTATIONS / TARIFICATIONS			
PRESTATIONS INCLUSES : *	PRIX UNITAIRE (€) : *	NOMBRE : *	MONTANT (€) :
Frais de dossier	15	6	90
Spectacle	54	6	324
Repas	42	6	252
Vol	515	6	3090
Ajouter une ligne Supprimer une ligne			
MONTANT TOTAL À LA DATE DE L'INSCRIPTION		3756	EUR TTC

Fonction pour rechercher dans les <select>

Pour faciliter la vie de l'utilisateur, et lui éviter de chercher parmi de très longue liste déroulante, telle que celles des clients, des contrats ou des hôtels, j'ai mis en place un système de recherche dans ces dernières

Egalement en Javascript à l'aide d'un script externe (Select2.js + JQuery) :

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/css/select2.min.css" rel="stylesheet" />
<script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/js/select2.min.js"></script>

<script>
  $() => {
    $('#contratClient').select2();
    $('#contratHotel').select2();
  });
</script>
```

Une fois les différents scripts et librairies incluses, il suffit d'indiquer l'id du <select> et de lui appliquer une fonction. Voici le rendu :

Client concerné par le contrat : *

Sélectionnez un client

ITINERAIRE :

ACTIVITÉS INCLUSES DANS LE

Pour les visites/excursions/activi

LANGUE PARLEE :

Sélectionnez un client

CARNET DE VOYAGE 31

Coulaud Patricia

JOUHIER MICHEL

LE CORVEC Thomas

MONDORAMAS / GROUPTOUR

Morgant Gauderic

Mise à jour automatique du contenu des <select>

Une des missions qui m'a été confiée, était l'ajout de contact pour les partenaires de l'entreprise. Ces partenaires et ces contacts devaient être sélectionnés lors de la saisie des bons d'échanges pour les clients. Il fallait donc remplir le champ <select> dynamiquement par les noms des contacts du partenaire sélectionné.

Pour ce faire il a fallu passer l'ensemble des contacts, d'un tableau PHP à une variable Javascript, puis filtrer les entrées du <select>.

HTML :

```
<label><strong> Partenaire : *</strong></label>

<select id="lePartenaire" name="lePartenaire" class="inputAddTxt" required onchange="selectPartner()">

    <option value="" hidden selected>Sélectionnez un partenaire</option>

    <?php
    while($dataPartners = $reqGetPartners->fetch())
    {
        ?>

        <option value="<?php echo $dataPartners['idPartenaire'] ?>"><?php echo $dataPartners['nomEntreprise']; ?></option>

        <?php
        }

        $reqGetPartners->closeCursor();

        ?>

</select>

&emsp;&emsp;&emsp;&emsp;&emsp;

<label><strong>Attendu par : *</strong></label>

<select id="leContact" name="leContact" class="inputAddTxt" required>

    <option value="" hidden selected>Sélectionnez le contact </option>

</select>
```

Javascript :

```
<script>

function selectPartner(){
    var contactList = <?php echo json_encode($reqGetContact->fetchAll());?>;

    var contactInnerHTML = contactList.filter(contact => contact.idPartenaire == lePartenaire.value);

    leContact.innerHTML = contactInnerHTML.map(contact => '<option value="'+contact.idContact+'"'>'+contact.nomContact+'</option>');
}

</script>
```

Lorsque l'on sélectionne un partenaire (ex : 365 Travel), seul les contacts (Jeanne, Mehdi et Pierre) de ce dernier apparaissent dans la liste, voici le rendu :

Partenaire : * 365 travel Attendu par : * Jeanne

Transfert pour : * Sélectionnez un hôtel

Jeanne
Mehdi
Pierre

Tableau pour les suppléments pour les devis

L'un des problèmes auquel faisait face la gérante, était la saisie de ses devis, qui était limité à un supplément en plus des suppléments déjà présent, elle ne pouvait donc pas en ajouter plus. Elle souhaitait en plus de pouvoir ajouter des colonnes, pouvoir également ajouter des lignes à ce tableau.

Après quelques heures de réflexion j'ai réussi à trouver une solution pour régler ce problème en mettant en place un tableau ayant la possibilité d'augmenter et diminuer ses lignes, mais également ses colonnes.

HTML :

```
<div class="formAdd">

  <center>

    <table class="tab">
      <tbody id="suppTable">
        <tr id="suppHeader">
          <td class="601"><strong><center>Supplément : </center></strong><br><input name="nameSupp601" type="text" /> </td>
        </tr>
        <tr id="501">
          <td class="601"><input type="text" name="601501"></td>
        </tr>

        <input id="l501" type="hidden" name="suppLines" value="501"/>
        <input id="c601" type="hidden" name="suppColumns" value="601"/>
      </tbody>
    </table>
    <br><br>
    <div>
      <input type="button" value="+ Ligne" onclick="addSuppLine();" />
      <input type="button" value="- Ligne" onclick="removeSuppLine();" />
      <br><br>
      <input type="button" value="+ Colonne" onclick="addSuppColumn();" />
      <input type="button" value="- Colonne" onclick="removeSuppColumn();" />
    </div>

  </center>

</div>
```

La partie en Javascript pour l'ajout de ligne :

```
//TABLEAU SUPPLEMENTS

var nbLines = 501;
var nbColumns = 601;

// Fonction d'ajout de ligne
function addSuppLine(){
    nbLines++;

    var htmlColumns = '';

    for(var i = 1; i <= (nbColumns-600); i++){
        htmlColumns += '<td class="'+(600+i)+'"><input type="text" name="'+((600+i)+++'+nbLines)+'" required/></td>';
    }
    suppTable.insertAdjacentHTML('BeforeEnd', '<tr id="'+nbLines+'">'+htmlColumns+'</tr>');

    document.getElementById('l'+(nbLines-1)).remove();
    suppTable.insertAdjacentHTML("BeforeEnd", "<input id='l"+nbLines+"' type='hidden' name='suppLines' value='"+nbLines+"'>");
}

// Fonction de suppression de ligne
function removeSuppLine(){

    var el = document.getElementById(nbLines);

    if(nbLines != 501){

        el.remove();

        nbLines = nbLines - 1;

        document.getElementById('l'+(nbLines+1)).remove();
        suppTable.insertAdjacentHTML("BeforeEnd", "<input id='l"+nbLines+"' type='hidden' name='suppLines' value='"+nbLines+"'>");
    }
}
```

La partie en Javascript pour l'ajout de colonne :

```
// Fonction d'ajout de colonne
function addSuppColumn(){
    nbColumns++;

    for(var j = 1; j <= (nbLines-500); j++){

        var el = document.getElementById(500+j);

        el.insertAdjacentHTML('BeforeEnd', '<td class="'+nbColumns+'"><input name="'+(nbColumns+++'+(500+j)++)'" type="text" required/>');
    }

    suppHeader.insertAdjacentHTML('BeforeEnd', '<td class="'+nbColumns+'"><strong><center>Supplément : </center></strong><br><input type="text" value=""></td>');

    document.getElementById('c'+(nbColumns-1)).remove();
    suppTable.insertAdjacentHTML("BeforeEnd", "<input id='c"+nbColumns+"' type='hidden' name='suppColumns' value='"+nbColumns+"'>");
}

// Fonction de suppression de colonne
function removeSuppColumn(){
    var allRows = document.getElementsByClassName(nbColumns);

    console.log(allRows);

    if(nbColumns != 601){
        while(allRows[0]) {
            allRows[0].parentNode.removeChild(allRows[0]);
        }

        nbColumns = nbColumns - 1;

        document.getElementById('c'+(nbColumns+1)).remove();
        suppTable.insertAdjacentHTML("BeforeEnd", "<input id='c"+nbColumns+"' type='hidden' name='suppColumns' value='"+nbColumns+"'>");
    }
}
```


Et le rendu final de ce tableau :

Supplément :	Supplément :	Supplément :
supp1	supp2	supp3
120€	40€	50€
200€	50€	30€
150€	10€	0€
220€	50€	40€

+ Ligne - Ligne

+ Colonne - Colonne

Génération du PDF pour les convocations

Durant mon stage, j'ai également du modifier les PDF déjà créés par les précédents stagiaires, et également dû en créer à l'aide de la librairie FPDF, dont voici un exemple de code (celui-ci affiche les escales du vol) :

```
$CustomersConvocation->Cell(190,35,utf8_decode(''),1);

$CustomersConvocation->Ln(3);

$CustomersConvocation->setX(20);

$CustomersConvocation->Cell(20,10,utf8_decode("Durée de l'escale : " . $dataEscaleAl['dureeEscale']));

$CustomersConvocation->SetX(80);

$CustomersConvocation->Cell(20,10,utf8_decode("Lieu de l'escale : " . $dataEscaleAl['lieuEscale']));

$CustomersConvocation->SetX(150);


$CustomersConvocation->Cell(20,10,utf8_decode("Enregistrement : " . $dataEscaleAl['enregistrement']));

$CustomersConvocation->Ln(9);

$CustomersConvocation->setX(20);

$CustomersConvocation->cell(20,10,utf8_decode("Date de l'escale : " . $dataEscaleAl['dateEscaleFormat']));
```

Et le rendu du PDF (Le PDF entier est disponible dans les documents produits) :



CONVOCATION

LE CORVEC Thomas | Mobile : 02010202 | Fixe : 02020202020
1 rue branly
56400 Auray

Important : Ceci est un mémo voyage qui contient des informations importantes pour obtenir vos cartes d'embarquements, nécessaires pour voyager. Veuillez en prendre connaissance dans son intégralité. Nous vous conseillons de le conserver pendant toute la durée de votre voyage.

--- Participants ---

Nom : LE CORVEC	Prénom : Thomas	Age : 22/06/2021	Nationalité : Française
Nom : Tes	Prénom : Test	Age : 23/06/2021	Nationalité : Française
Nom : Henri	Prénom : Henri	Age : 19/02/2021	Nationalité : FR
Nom : ytryt	Prénom : yrttrt	Age : 23/06/2021	Nationalité : rtytyt

--- Trajet Aller ---

Convoqué le 08/06/2021	À l'aéroport : Charles de Gaulle	Pour : 10h
Comptoir n°56	Numéro de vol : TF87988	Durée du vol : 1h
Départ de Paris à 12h		Arrivée à Nice à 13h

--- Etape Aller ---

Durée de l'escale : 2h	Lieu de l'escale : Nice	Enregistrement : 253
Date de l'escale : 28/06/2021	Numéro de vol : ET76678	Durée du vol : 6h
Départ de Nice à 15h		Arrivée à Moscou à 21h

Archivage des éléments

Durant le stage, la gérante s'est rendu compte que les contrats et les devis allaient très vite s'accumuler dans le logiciel, elle a donc souhaité une fonctionnalité permettant d'archiver ses anciens contrats et devis.

Pour se faire j'ai tout d'abord ajouter un état au devis et au contrat en base de données, afin de savoir si ces derniers sont d'actualités ou s'ils sont archivés.

J'ai ensuite rajouter un onglet dans le menu pour archiver les contrats et devis ainsi que de pouvoir les consulter.

Voici la page permettant d'archiver (en fonction du type d'élément et de l'année) :



The screenshot shows a web form titled "Éléments à archiver". It contains two input fields: "Élément à archiver : *" which is a dropdown menu with "Devis" and "Contrat" as options, and "Année à archiver : *" which is a text input field. Below these fields is a button labeled "Archiver".

Également le code permettant d'appeler les fonctions du fichier Model.php (permettant d'interagir avec la base de données) :

```
function archiveCreate(){

    $elementType = $_POST['elementType'];
    $elementYear = $_POST['elementYear'];

    if($elementType == "quotation"){

        archivedQuotation($elementYear);

        header('Location: index.php?state=seeArchivedQuotation');

    }

    elseif($elementType == "contract"){

        archivedContract($elementYear);

        header('Location: index.php?state=seeArchivedContract');

    }

}
```

Les fonctions qui interagissent avec la base de données :

```
function archivedQuotation($year){  
    $bdd = dataBaseConnection();  
  
    $reqArchivedQuotation = $bdd->prepare("UPDATE devis SET etat='archived' WHERE YEAR(dateSaisie) = :elementYear");  
  
    $reqArchivedQuotation->bindParam(':elementYear', $year);  
  
    $reqArchivedQuotation->execute();  
}  
  
function archivedContract($year){  
    $bdd = dataBaseConnection();  
  
    $reqArchivedContract = $bdd->prepare("UPDATE contrat SET etat='archived' WHERE YEAR(dateSignature) = :elementYear");  
  
    $reqArchivedContract->bindParam(':elementYear', $year);  
  
    $reqArchivedContract->execute();  
}
```

Une fois archivés, les contrats et les devis sont accessibles depuis une page, où il est possible de les consulter et de les supprimer :

N° Contrat	Nom du client	Date d'édition	Dates du voyage	Destination	Nombre de Participants	PDF	Supprimer
C000071	LE CORVEC Thomas	14/06/2021	Du 18/06/2021 au 00/00/0000	errtrtrtre	5	PDF	Supprimer
C000056	LE CORVEC Thomas	11/06/2021	Du 08/06/2021 au 01/06/2021	ter	5	PDF	Supprimer
C000058	LE CORVEC Thomas	10/06/2021	Du 18/06/2021 au 00/00/0000	errtrtrtre	5	PDF	Supprimer
C000046	LE CORVEC Thomas	09/06/2021	Du 20/12/2023 au 00/00/0000	Paris	5	PDF	Supprimer
C000055	LE CORVEC Thomas	09/06/2021	Du 10/07/2023 au 00/00/0000	Paris	5	PDF	Supprimer
C000052	LE CORVEC Thomas	08/06/2021	Du 08/11/2022 au 00/00/0000	Paris	5	PDF	Supprimer
C000054	LE CORVEC Thomas	08/06/2021	Du 05/08/2021 au 00/00/0000	Paris	5	PDF	Supprimer
C000047	LE CORVEC Thomas	08/06/2021	Du 10/12/2023 au 00/00/0000	Paris	5	PDF	Supprimer
C000043	LE CORVEC Thomas	04/06/2021	Du 00/00/0000 au 00/00/0000	Paris	5	PDF	Supprimer
C000045	LE CORVEC Thomas	03/06/2021	Du 10/10/2023 au 00/00/0000	Paris	5	PDF	Supprimer
C000031	Coulaud Patricia	31/05/2021	Du 02/06/2021 au 09/06/2021	gfgf	1	PDF	Supprimer
C000030	Yaouanq Titouan	26/05/2021	Du 15/07/2021 au 09/05/2022	Québec	1	PDF	Supprimer
C000029	JOUIER MICHEL	18/02/2021	Du 25/02/2021 au 28/02/2021	Marrakech	2	PDF	Supprimer

[Extraction comptable \(génération Excel\)](#)

La dernière mission qui m'a été confiée, était l'ajout d'une fonctionnalité d'extraction comptable, afin d'extraire les données des contrats pour en calculer les différents prix et les différentes marges.

Avant de commencer la génération des fichiers Excel, il me fallait déjà trouver une librairie permettant de le faire. Après plusieurs essais plus ou moins concluant, mon choix s'est porté sur la librairie XLSXWriter, qui est composé d'un seul et unique fichier PHP.

Cette dernière permettait de générer des fichiers au format XLSX assez rapidement et facilement.

Après avoir ajouté cette dernière, j'ai tout d'abord étudié le rendu qu'attendait la gérante. Puis j'ai tout d'abord commencé par ajouter une fonctionnalité au logiciel, lui permettant de définir et modifier certaines informations apparaissant sur le tableau d'extraction (Montant de la TVA, libellé et montant des prestations ou encore les dates d'acompte), le tout stocké en base de données.

Informations majeures

N° contrat : C000092 Date d'édition : 23/06/2021

Client concerné par le contrat : LE CORVEC Thomas

INFORMATIONS GÉNÉRALES

Date d'acompte * : 09 / 07 / 2021 ☐ Montant TVA * : 263 €

LIBELLÉ / MONTANT

Aérien	Libellé	Vol de Paris vers Moscou
	Montant	2200
Prestations	Libellé	Spectacle
	Montant	518
Assurances	Libellé	Aucune
	Montant	0
Autres 1	Libellé	Repas
	Montant	253
Autres 2	Libellé	Hotel + Transfert
	Montant	358

Modifier les lignes d'extraction

Une fois cela fait, j'ai ajouté une page permettant d'extraire les contrats, entre 2 dates sélectionnées :

Date d'extraction

Date de début d'extraction : * 14 / 06 / 2021 ☐ Date de fin d'extraction : * 01 / 07 / 2021 ☐

Extraire

Voici le code permettant cette extraction :

```

function getContractBetweenDate($startDate, $endDate)
{
    $bdd = dataBaseConnection();

    $reqGetContract = $bdd->prepare("SELECT *, DATE_FORMAT(dateSignature , '%d/%m/%Y') AS dateSignatureFormat, DATE_FORMAT(dateReglement , '%d/%m/%Y') AS dateReglementFormat, DATE_FORMAT(dateReglementSo , '%d/%m/%Y') AS dateReglementSoFormat, DATE_FORMAT(dateDepart , '%d/%m/%Y') AS dateDepartFormat, DATE_FORMAT(dateRetour , '%d/%m/%Y') AS dateRetourFormat, DATE_FORMAT(dateNaissCli , '%d/%m/%Y') AS dateNaissCliFormat, DATE_FORMAT(dateAcompte , '%d/%m/%Y') AS dateAcompteFormat
FROM contrat
INNER JOIN client ON client.numCli = contrat.numCli
INNER JOIN assurance ON assurance.idAssurance = contrat.idAssurance
INNER JOIN hebergement ON contrat.idHebergement = hebergement.idHebergement
INNER JOIN voyage ON contrat.idVoyage = voyage.idVoyage
INNER JOIN typevoyage ON voyage.idTypeVoyage = typevoyage.idTypeVoyage
INNER JOIN facture ON contrat.idFacture = facture.idFacture
INNER JOIN programme ON contrat.idProgramme = programme.idProgramme
INNER JOIN formalites ON contrat.idFormalites = formalites.idFormalites
INNER JOIN annulationorganisateur ON contrat.numContrat = annulationorganisateur.numContrat
INNER JOIN annulationvoyageur ON contrat.numContrat = annulationvoyageur.numContrat
INNER JOIN devise ON contrat.numContrat = devise.numContrat
INNER JOIN typeforfait ON voyage.idTypeForfait = typeforfait.idTypeForfait
INNER JOIN transport ON voyage.idTransport = transport.idTransport
INNER JOIN hotel ON hebergement.idHotel = hotel.idHotel
INNER JOIN repas ON hebergement.idRepas = repas.idRepas
INNER JOIN typeassurance ON assurance.idTypeAssurance = typeassurance.idTypeAssurance
INNER JOIN moyenpaiement ON facture.idMoyenPaiement = moyenpaiement.idMoyenPaiement
INNER JOIN ligneextractcontract ON ligneextractcontract.numContrat = contrat.numContrat
WHERE dateSignature BETWEEN :startDate AND :endDate
ORDER BY dateSignature DESC, contrat.numContrat DESC");

$reqGetContract->bindParam(':startDate', $startDate);

$reqGetContract->bindParam(':endDate', $endDate);

$reqGetContract->execute();
}

```

Une fois les différents contrats récupérés, il faut ensuite les afficher dans le fichier Excel et effectuer les différents calculs. Tout d'abord on définit le nom des colonnes et leur style :

```

function extractContractsForFile(){
    if(isset($_POST['extractData'])){
        $reqGetContractsBetweenDate = getContractBetweenDate($_POST['startDate'], $_POST['endDate']);

        $header = array(
            'Réservation faite le' => 'DD/MM/YYYY',
            'Nom' => 'string',
            'Destination' => 'string',
            'Date de départ' => 'DD/MM/YYYY',
            'Prix de vente' => 'euro',
            'Acompte' => 'euro',
            'Moyen de paiement de l\'acompte' => 'string',
            'Date d\'acompte' => 'DD/MM/YYYY',
            'Solde' => 'euro',
            'Remboursement' => 'euro',
            'Date de solde' => 'DD/MM/YYYY',
            'Moyen de paiement du solde' => 'string',
            'TVA' => 'euro',
            'Nom Aérien' => 'string',
            'Montant Aérien' => 'integer',
            'Nom Prestations' => 'string',
            'Montant Prestations' => 'integer',
            'Nom Assurances' => 'string',
            'Montant Assurances' => 'integer',
            'Nom Autres 1' => 'string',
            'Montant Autres 1' => 'integer',
            'Nom Autres 2' => 'string',
            'Montant Autres 2' => 'integer',
            'Marge' => 'integer'
        );

        $styleHeader = array(
            "fill" => "#ebae34",
            "widths" => [20,20,30,20,15,15,20,20,15,15,20,20,15,20,20,20,20,20,20,20,15],
            "halign" => "center",
            "valign" => "center",
            "border" => "top,left,right,bottom",
            "border-style" => "thin",
            "border-color" => "#000000"
        );
    }
}

```

Puis on va itérer sur les différents contrats récupérés afin de créer de nouvelle ligne au fichier Excel, puis calculer les différentes marges et tarifs de l'ensemble de ces derniers.

```
$totalTva = 0;
$totalAerien = 0;
$totalPrestations = 0;
$totalAssurances = 0;
$totalAutres = 0;
$totalMarge = 0;
$total = 0;

$writer = new \PHPExcel_Writer();
$writer->writeSheetHeader('Sheet 1', $header, $styleHeader);

while($dataContract = $reqGetContractsBetweenDate->fetch()){
    $dateSolde = ($dataContract['dateReglementSo'] == '0000-00-00') ? $dataContract['dateReglement'] : $dataContract['dateReglementSo'];

    $marge = $dataContract['montantTotal'] - ($dataContract['montantTva'] + $dataContract['montantAerien'] + $dataContract['montantPrestations']);

    $totalTva += $dataContract['montantTva'];
    $totalAerien += $dataContract['montantAerien'];
    $totalPrestations += $dataContract['montantPrestations'];
    $totalAssurances += $dataContract['montantAssurances'];
    $totalAutres += ($dataContract['montantAutres1'] + $dataContract['montantAutres2']);
    $totalMarge += $marge;
    $total += $dataContract['montantTotal'];

    $data = array(
        $dataContract['dateSignature'],
        ($dataContract['nomCli'] . ' ' . $dataContract['prenomCli']),
        $dataContract['destinationVoyage'],
        $dataContract['dateDepart'],
        $dataContract['montantTotal'],
        $dataContract['montantAcompte'],
        $dataContract['libelleMoyenPaieement'],
        $dataContract['dateAcompte'],
        $dataContract['montantSolde'],
        $dataContract['montantAvair'],
        $dateSolde,
        $dataContract['libelleMoyenPaieement'],
        $dataContract['montantTva'],
        $dataContract['libelleAerien'],
        $dataContract['montantAerien'],
        $dataContract['libellePrestations'],
        $dataContract['montantPrestations'],
        $dataContract['libelleAssurances'],
        $dataContract['montantAssurances'],
        $dataContract['libelleAutres1'],
        $dataContract['montantAutres1'],
        $dataContract['libelleAutres2'],
        $dataContract['montantAutres2'],
        $marge
    );

    $writer->writeSheetRow('Sheet 1', $data, $styleData);
}
```

Puis afficher ces différents tarifs et marges, puis télécharger le fichier sur l'ordinateur de l'utilisateur :

```
$styleFinal = array(
    "halign" => "center",
    "valign" => "center"
);

$writer->writeSheetRow('Sheet 1', array());
$writer->writeSheetRow('Sheet 1', array());
$writer->writeSheetRow('Sheet 1', array("", "TOTAL DES VENTES", ($total.'€')), $styleFinal);
$writer->writeSheetRow('Sheet 1', array());
$writer->writeSheetRow('Sheet 1', array("", "TOTAL"), $styleFinal);
$writer->writeSheetRow('Sheet 1', array("", ($totalAerien.'€'), "AERIEN"), $styleFinal);
$writer->writeSheetRow('Sheet 1', array("", ($totalPrestations.'€'), "PRESTATIONS"), $styleFinal);
$writer->writeSheetRow('Sheet 1', array("", ($totalAssurances.'€'), "ASSURANCES"), $styleFinal);
$writer->writeSheetRow('Sheet 1', array("", ($totalAutres.'€'), "AUTRES"), $styleFinal);
$writer->writeSheetRow('Sheet 1', array());
$writer->writeSheetRow('Sheet 1', array("", (($totalAerien+$totalPrestations+$totalAssurances+$totalAutres).€'), "TOTAL TOUTES PRESTATIONS"), $styleFinal);
$writer->writeSheetRow('Sheet 1', array());
$writer->writeSheetRow('Sheet 1', array("", "TOTAL MARGE", ($totalMarge.'€')), $styleFinal);
$writer->writeToFile('test.xlsx');

header('Location: test.xlsx');
```

Voici le rendu de ce fichier (Ce fichier est disponible entièrement dans les documents produits) :

[illegible]

Et également les différentes informations renseignées plutôt (Montant de la TVA, etc...)

M	N	O	P	Q	R	S	T	U	V	W	X
TVA	Nom Aérien	Montant Aérien	Nom Prestations	Montant Prestations	Nom Assurances	Montant Assurances	Nom Autres 1	Montant Autres 1	Nom Autres 2	Montant Autres 2	Marge
263 00 €	Vol de Paris vers Moscou	2200	Spectacle	518	Aucune	0	Repas	253	Hotel + Transfert	358	56558
10 00 €	misterfly	10	bedonlines	0	valeur assurances	20		0		0	3362
0 00 €	Tunisair	150	allyssa	30		0	allyssa	50	imed	40	752
273 00 €	TOTAL AÉRIEN	2360	TOTAL PRESTATIONS	548	TOTAL ASSURANCES	20	TOTAL AUTRES	701			60672

6 - Sécurité

Lors du développement du logiciel j'ai remarqué une principale faille de sécurité, très importante qui plus est, il était possible à l'aide des requêtes envoyées vers la base de données de faire de l'injection SQL, car ces dernières concaténaient simplement les paramètres avec la requête, sans utiliser les fonctions prévues à cet effet. Pour résoudre ce problème, j'ai modifier l'ensemble des requêtes SQL à l'aide des fonctions prévues, afin d'éviter toutes injections SQL.

Example :

Requête SQL concaténée :

```
$requestPayWay = $bdd->query("INSERT INTO moyenpaiement (idMoyenPaiement,libelleMoyenPaiement)
VALUES (null, ".$libelle.")");
```

Requête SQL avec utilisation de fonction :

```
$requestPayWay = $bdd->prepare("INSERT INTO moyenpaiement (idMoyenPaiement,libelleMoyenPaiement)
VALUES (null, :libelle)");
$requestPayWay->bindParam(':libelle',$libelle);
$requestPayWay->execute();
```


7 - Gestion de version

Durant la réalisation du projet, étant principalement tout seul, j'ai préféré éviter l'utilisation d'une solution versioning tel 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, nous avons pu finir et réaliser entièrement ce projet, qui je l'espère a donné entière satisfaction à la gérante. Malgré les problèmes et les fonctionnalités apparues au cours de projet, nous avons eu le temps de finir entièrement ce dernier dans les temps.

Ce stage m'aura apporté beaucoup sur le plan professionnel mais également sur le plan personnel, en me permettant d'évoluer techniquement, d'acquérir de nouvelles compétences, tel 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
PROCEDURE – Créer une cagnotte.docx	Procédure expliquant la création d'une cagnotte afin de permettre aux futurs collaborateurs de pouvoir les créer facilement
PROCEDURE – Mise en forme email.docx	Procédure expliquant la modification de la mise en forme de l'email, afin de permettre à la gérante de le modifier à sa guise.
PROCEDURE – Valider un don par cheque.docx	Procédure expliquant la validation et la prise en compte d'un don effectué par chèque pour les cagnottes cadeaux.