

Celma Adrien

BTS SIO SLAM

Campus Saint-Aspais Melun

18 Rue Louis Beaunier, 77000 Melun



Rapport de Stage

08/01/2024 – 09/02/2024

ticketmaster[®]

Ticketmaster France
Immeuble CityLife – 28 allée d'Aquitaine – 92000 Nanterre

Sommaire

1	Introduction	2
2	Etude de l'existant	
3	Cahier des charges	3
4	Présentation approfondie des tâches effectués.....	4
4.1	Première tâche.....	4
4.2	Deuxième tâche	5
5	Conclusion	6
6	Annexe	6

1 Introduction

From January 8 to February 9, 2024, I completed an internship at Ticketmaster France. During this internship within the development team, I was able to focus on maintaining an application solution, fixing bugs or adding features to the company's internal software. The main source of this report was the various lessons learned from the daily practice of the tasks to which I was assigned. In order to faithfully and analytically report the month spent within the Ticketmaster France company, it seems logical to first present a presentation of the company, its activity, the service that welcomes me and the IT context in which I operate. . Then to describe what my tutor expected of me in the company, then it will be explained in more depth the different tasks that I was able to carry out within the department. Finally, I will conclude with what I was able to bring to the company and what I learned during my internship.

2 Etude de l'existant

Ticketmaster est une marque américaine mondialement reconnue qui opère dans plus de 30 pays à travers le monde et qui offre un service de vente de billets pour une grande variété d'événements, allant des concerts aux spectacles sportifs en passant par les festivals et les pièces de théâtre.

Ticketmaster France est une filiale de Ticketmaster, tout comme celle-ci elle est spécialisée dans la vente de billets pour des événements, c'est l'une des principales entreprises de billetterie et de divertissement en France. Ils exploitent un site web où les clients peuvent rechercher des événements en fonction de leur lieu, de leur catégorie et de leurs préférences. Ils peuvent acheter des billets en ligne de manière sécurisée et les recevoir par courrier électronique, par la poste, ou les récupérer sur place lors de l'événement. Les clients peuvent également se rendre à des points de ventes partenaires (les clients peuvent acheter leurs billets dans des magasins comme E. Leclerc, Carrefour, ...). L'entreprise travaille en collaboration avec des organisateurs d'événements, des salles de concert, des stades, des théâtres et d'autres lieux pour offrir une large sélection de billets à ses clients. Un service client est entièrement dédié pour répondre aux questions, résoudre les problèmes et fournir une assistance tout au long du processus d'achat de billets.

A l'inverse de ses concurrents, Ticketmaster France tente de se différencier d'eux en proposant par exemple des plans de salle interactifs (laisse la possibilité aux clients de pouvoir se placer là où il le souhaite lors de l'événement), une application mobile, la possibilité d'acheter des billets collector et d'acheter des places de catégories "Platinum". Les billets collector sont des billets présentés sous la forme d'une carte avec une illustration aux couleurs de l'artiste ou du spectacle. Lorsque vous achetez une place vous devez choisir une catégorie de place, chacune correspond à un prix statué et à un emplacement précis. Il existe plusieurs sortes de catégories comme Standard, Or, Diamant, ... La catégorie "Platinum" est l'une de ses catégories, elle fournit un emplacement privilégié et donne parfois accès à la loge de l'artiste avant une représentation. Sur certains événements les clients ont même le choix entre différents packages leurs offrant divers souvenirs et avantages du moment.

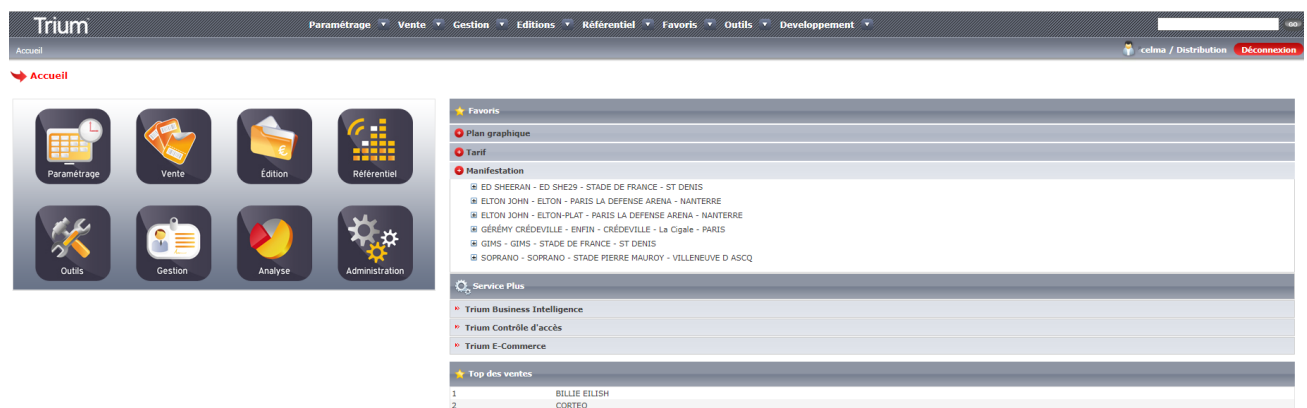
Cette entreprise se compose de plusieurs équipes et l'une d'elle est l'équipe de développement dans laquelle j'ai fait partie durant ses 5 semaines de stage. Cette équipe se forme d'un responsable, d'un responsable adjoint et de ses développeurs. Dans l'entreprise cette équipe est par exemple chargée

de maintenir à jour les logiciels utilisés par l'entreprise comme l'application Trium permettant le paramétrage de vente des événements. L'équipe se charge aussi du développement de webservices.

Je suis intervenu dans ce département informatique pour des améliorations et des corrections de bugs, ajouter des fonctionnalités sur des paramètres préexistants sur le logiciel interne de l'entreprise et faire de la maintenance de la solution applicative "Trium"

3 Cahier des charges

L'entreprise possède un logiciel collaboratif nommés « Trium », c'est un site web interne développé en CSharp. Cette plateforme permet à n'importe quel employé d'avoir accès aux différentes informations disponibles sur les événementiels. Cette même plateforme nous permet par exemple de rajouter de nouvelles données sur un événement, d'en supprimer, d'en créer ou bien d'annuler la commande d'un client. L'entreprise mets à disposition le logiciel à ses points de ventes en magasin pour que les clients des enseignes puissent acheter des places de spectacle. Chacun peut y accéder, mais chacun possède des droits différents. Ainsi les points de ventes en magasin n'ont accès qu'à la partie qui leurs permet de vendre, en revanche l'équipe de développement peuvent accéder à l'entièreté du projet. Cela permet de centraliser et simplifier toutes leurs actions sur un même endroit.



(Accueil du logiciel Trium)

Il m'a été demandé d'ajouter dans l'application métier de la société Ticketnet, de nouveaux écrans de paramétrages, ainsi les tarifs proposés dans les manifestations (événement, spectacle, concert ...) sont constitués d'une nature de clientèle, d'une catégorie, d'un montant pour un mode de commercialisation (Web, Mobile, Magasin...), les natures de clientèle permettent de définir des tarifs spécifiques ainsi que des comportements en fonction des besoins. Il est par exemple possible de définir des tarifs pour les porteurs de carte magasin (Auchans, Leclerc, ...) dans ce cas le code de la carte fidélité sera demandé lors du choix des places d'un événement, il est aussi possible d'avoir des avantages pour d'autre porteur de carte comme par exemple les clients de la carte Revolut qui pouvait bénéficier d'une prévente à tarif préférentielle pour les places du match de NBA à Paris qui a eu lieu en 2024.

Toutes ces fonctionnalités sont disponibles dans [l'écran de paramétrages des natures de clientèle](#). Il m'a été demandé d'ajouter dans [cet écran](#) un bouton d'information permettant la saisie des

commentaires en Français et en Anglais pour une nature de clientèle, ces informations seront ensuite visibles sur le site web de la société dans [une info-bulle](#) au survole d'un tarif. Cette information permet d'expliquer à qui s'adresse les différents tarifs, exemple : un « Tarif étudiant » est un tarif valable sur présentation d'un justificatif ou un « tarif enfant » accessible pour les enfants de 3 à 12 ans.

Il m'a été aussi demandé de pouvoir modifier ces informations au niveau de la manifestation, ainsi le paramétrage fait au niveau de la nature de clientèle peut être surchargé au niveau de la manifestation, c'est-à-dire, que les commentaires peuvent être supprimés ou modifiés. Et enfin effectué tous les tests unitaires afin de s'assurer du bon déroulement des modifications.

Pour finir on m'a demandé d'ajouter dans la [fiche manifestation](#) un bouton radio pour le tarif en temps réel, cette fiche contient l'ensemble des informations liées à un événement. Et de faire les modifications nécessaires pour son bon fonctionnement.

4 Présentation approfondie des tâches effectuées

4.1 Première tâche

Pour ce développement, j'ai ajouté le bouton « informations » dans [l'écran de paramétrages des natures de clientèle](#), qui permet l'ouverture d'un nouvel écran pour la saisie des commentaires. L'application métier (Trium) est un site web interne, pour identifier les pages web à modifier, il suffit de visualiser le contenu de l'URL du navigateur. Les pages web sont développées en CSharp, j'ai donc ajouté un [bouton ASP DotNet dans la page détail des natures de clientèle](#) qui permet l'affichage du nouvel écran.

La nouvelle page web créée se nomme [« AssociationInformationsNatureClientele »](#),

Elle comporte :

- [un entête contenant l'identifiant, la référence et le libellé court d'une nature de clientèle](#)
- [les deux zones de saisie des commentaires](#)
- [deux boutons d'action, l'un pour la validation de la saisie et l'autre pour l'abandon des modifications.](#)

[Pour m'aider à la réalisation du projet, mon tuteur de stage m'a présenté une maquette de ce à quoi cela devait ressembler.](#)

Lors du chargement de la page d'information (événement [Page_Load](#)) on vérifie si la valeur de la propriété « IsPostBack » est à False, si tel est le cas cela correspond à l'initialisation de la page, on vérifie donc les paramètres en entrant via la méthode [« getParameters »](#) celle-ci récupère les données transmises par la page détail des natures de clientèle et construit l'objet « nature » de type « NatureClientele ». La [méthode « getData »](#) permet la mise à jour de l'entête grâce à l'objet « nature » (identifiant, référence et libellé court) puis récupère les données de la base SQL Server grâce à l'identifiant de la nature de clientèle et de [la méthode « GetInformationsNatureClientele »](#) qui elle-même appelle une procédure stockée.

Pour alimenter la page d'information nature de clientèle, et pouvoir stocker les commentaires j'ai donc dû créer une nouvelle table « info_natcli ». Cette table se compose des champs suivants :

- [idnatcli : identifiant de la nature de clientèle de type INT](#)
- [info_fr et info_en : commentaires en français et en anglais des tarifs de type varchar](#)
- [utilmodi : utilisateur ayant fait une modification, de type varchar](#)
- [datmodi : date à laquelle ça a été modifié, de type datetime](#)

Pour la récupération des informations SQL utile à l'affichage j'ai créé la procédure stockée suivante :

- [InformationNatureClientelSelect](#).

Pour la sauvegarde des données dans SQL server j'ai créé une autre procédure stockée :

- [« InformationsNatureClienteleInsertOrUpdate »](#)

Elle permet la gestion en création et en modification, ainsi si l'enregistrement n'existe pas en base de données c'est une insertion, dans l'autre cas c'est une modification des données.

Pour pouvoir modifier ou supprimer les commentaires de la nature de clientèle au niveau de la manifestation, j'ai dû modifier une [procédure SQL préexistante. Elle permet d'insérer les informations de la table de nature de clientèle dans celle de la manifestation.](#)

4.2 Deuxième tâche

La dernière tâche qu'on m'a demandé de faire était l'ajout d'un bouton radio permettant l'affichage du tarif en temps réel. La variable a déjà été créée dans une table, elle est de type Booléen nul. Même procédé qu'auparavant, je retrouve le nom de la page à modifier dans l'url du navigateur internet. Puis [j'y ajoute mon bouton radio ASP](#). Lors de l'activation de l'un des deux boutons, il faut la possibilité que si l'on appuie sur « Non » lorsque le bouton « Oui » est actionné, que le bouton « Oui » ne le soit plus. [Ainsi avec une boucle if, on va pouvoir bloquer l'une des deux options lorsque l'autre est activé.](#)

Pour la récupération des informations SQL utile à l'affichage il faut rajouter cette nouvelle valeur à la procédure SQL [« ManifestationSelect »](#) pour qu'elle puisse sélectionner cette valeur par rapport au bon identifiant de la manifestation. Pour permettre d'enregistrer cette nouvelle valeur dans la base de données et qu'elle puisse également se mettre à jour, on va devoir dans la méthode [« UpdateManifestation »](#) de la bibliothèque de classe ajouter le nouveau paramètre qui sera appelé dans la procédure SQL. On y voit la classe « ExecuteSQL » qui permet d'exécuter une procédure stockée, elle précède le nom de la procédure à modifier, elle se nomme [« ManifestationUpdate »](#). Dans cette méthode, il y a deux parties. [La première est pour vérifier si la valeur a été modifier, la deuxième va mettre à jour la donnée si elle a été modifier.](#) Pour se faire on va [déclarer la valeur « @tariftempsreel »](#) de type Booléen Nul correspondant à la nouvelle valeur, et on va également [déclarer la valeur « @oldtariftempsreel »](#) qui correspond à la valeur qui est incrémenté dans la base.

Tout travail doit à la fin passer une série de tests pour remarquer d'éventuelles problèmes, l'un d'entre eux était de s'assurer du bon fonctionnement du bouton. Ainsi avec une simple requête SQL je pu vérifier selon l'identifiant d'une manifestation spécifique, si la valeur du tarif en temps réel était

bien enregistrée dans la base de données. Si en activant le bouton Oui, le tarif en temps réel est égal à 1 alors la donnée a bien été répertoriée.

5 Conclusion

Lors de ce stage, j'ai pu continuer à voir les différents aspects du travail de développeur, c'est-à-dire, le développement, les tests de recette et la livraison en production, j'ai pu aussi continuer à m'adapter au mieux à cet environnement. L'ensemble des modifications que j'ai apporté à l'application Trium, va être validé par le responsable qualité avant d'être mis en production. A l'issue de cela, mes modifications serviront aux gestionnaires de billetterie et faciliteront leurs prochaines tâches. J'ai eu de nombreux contacts avec les développeurs afin de comprendre comment était codé le logiciel, et j'y ai compris l'importance du travail d'équipe en informatique. Leurs aides avec leurs compétences ainsi que leurs expériences m'ont permis de mener à bien mes projets que je devais développer. J'ai pu prolonger mon expérience de développer en CSharp sur leur solution applicative, cette expérience m'a permis de me familiariser un peu plus avec ce langage de développement. J'ai continué à voir les procédures stockées ou j'ai pu renforcer mes connaissances en SQL en y travaillant dessus.

6 Annexe

Ecran de paramétrages de la nature de clientèle

The screenshot shows the 'Nature de clientèle' configuration screen in the Trium application. The interface includes a breadcrumb trail: Accueil > Référentiel > Vente > Natures de clientèle > Nature de clientèle. The main form contains the following fields and controls:

Identifiant	921
Référence	<input type="text" value="25J"/>
Libellé long	<input type="text" value="JEUNE 12-25 ANS"/>
Libellé court	<input type="text" value="12-25 ANS"/>
Type d'adhésion	<input type="text" value="(Aucun)"/>
Activer bip contrôle d'accès	<input type="checkbox"/>
Identifier le client	<input type="checkbox"/>
Places échangées	<input type="checkbox"/>
Promotion	<input type="checkbox"/>
Code Message associé	<input type="text"/>
Tarif de référence	<input checked="" type="checkbox"/>
Tarif éligible au calcul du prix minimum	<input checked="" type="checkbox"/>
Texte explicatif	<input type="checkbox"/>

On the right side of the form, there are three red buttons: **Confirmer modif.**, **Supprimer**, and **Abandonner**.

At the bottom of the screen, there are five red buttons: **Traduire**, **Nouveau**, **Profils**, **Populations**, and **Informations**.

Ecran Information de la nature de clientèle

Trium

Paramétrage

Accueil

Référentiel

Vente

Natures de clientèle

Nature de clientèle

Nature de clientèle

Identifiant	Référence	Libellé court
921	25J	12-25 ans

Commentaire tarif FR

Commentaire tarif EN


Confirmer modif.

Abandonner

Info Bulle

L'ODYSSÉE
LUMINEUSE

Une aventure familiale à ciel ouvert



Horaires

Tous les jours
du 16 février au 3 mars 2024 de 18h à 22h.
Toutefois, il est recommandé de venir à 20h
pour éviter la forte affluence.

Durée moyenne de visite:
1h10

1. Sélectionnez une date

FÉVRIER 2024

L	M	M	J	V	S	D
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29			

CATEGORIE UNIQUE

Dès 10.00 €

PlacementLIBRE

TARIF ADULTE

13.00 €

0

+

TARIF ENFANT

11.00 €

0

+

T

10.00 €

0

+

Tarif enfant accessible aux enfants de 3 à 12 ans inclus.

Votre séance

vendredi 16 février 2024 à 18:00

CATEGORIE UNIQUE

Dès 10.00 €

PlacementLIBRE

TARIF ADULTE

13.00 €

0

+

TARIF ENFANT

11.00 €

0

+

TARIF ETUDIANT

10.00 €

0

+

Fiche manifestation

RAPPEL : Le taux de TVA sélectionné est différent de celui préconisé selon les règles de TVA applicables sur la billetterie de spectacles.

Identifiant	Référence	Libellé court	Libellé long	Statut	Début	Fin	-
539194	CIG-ELM23	ELMER23	ELMER FOOD BEAT	En vente	07/12/2023	07/12/2023	

Création	Vente	Annulation	Site	Lieu	Serveur	Localisation
27/10/2022	27/10/2022		LA CIGALE - PARIS	La Cigale	0	TKN-DISTR-DISTRIB

Général

Comptabilité

Contrôle d'accès

Informations

Artiste

Mots clés

Planning

Revente

Référence

CIG-ELM23

Libellé long

ELMER FOOD BEAT

Commentaire billet

ARSENAL PRODUCTIONS PRESENTE

Genre

CONCERT

Début

07/12/2023

Durée des séances (minutes)

Date d'annonce

Heure

Date d'ouverture des ventes

Heure

Organisateur national

Organisateur

ARSENAL PRODUCTIONS

Manifestation à confirmer

☐ Non ☒ Oui

Placement libre uniquement

☐ Non ☒ Oui

Choix de la zone

☐ Non ☒ Oui

Chemin pour la vue à la place

Configuration

LI-CIGAL01

Identification client

☐ Non ☒ Oui

Contrôle d'identité

☒ Non ☐ Oui

Exclusion dans l'édition VAD

☒ Non ☐ Oui

Exclusion des tops des ventes / TM1 Reports

☒ Non ☐ Oui

Export TM International

☒ Non ☐ Oui

Organisateur national (BI)

Gestionnaire

TICKETNET-DISTRIBUTION

19/06 : Tarif MASTERDAY du 20/06 au 02/07 : Quota 30

Libellé court

ELMER23

Type manifestation

Standard

Sous-genre

POP ROCK FOLK

Fin

07/12/2023

Saison

Aucune

Référence externe

Fermeture anticipée

☒ Non ☐ Oui

Statut

En vente

N° de Licence

L2-R-21-9524/L3-R-21-9515

Délai de confirmation

Activation de l'ISM

☒ Non ☐ Oui

Tarif en temps réel

☒ Non ☐ Oui

Maquette billet

MAQMANIF

Identification bénéficiaire

☒ Non ☐ Oui

Annulations Autorisées

☐ Non ☒ Oui

Calendrier de séances

☒ Non ☐ Oui

Garantie fan

☒ Non ☐ Oui

Utilisateur gestionnaire

julienm

Séances

Plan manifestation

Tarifs

Importer Plan

Mode de Paiement

Mode de retrait

Limite de vente

Générer contingents

Autorisation PV

Adhésion

Co-gestionnaire

Nature de clientèle

Catégories places

Contingents

Générer formules

Questionnaires

Produits

Maquettage e-ticket

Éditions

Feuille de contrôle

Places par tarif

Type catégorie places

Ouv./Farm.

Déclaration Sibil

Platinum

Rapport planifié

Visuels

Confirmer modif.

Annulation

Abandonner

Suppr. favoris

Bouton « Informations »

```
<asp:Content ID="Content2" ContentPlaceHolderID="cphButtons" Runat="Server">
    <asp:Button runat="server" ID="btnProfils"PostBackUrl="AssociationProfilsNatureClientele.aspx" OnClick="btnProfils_Click" Text="<%% Resources: GlobalLocalization, Profils %>" CausesValidation="false" />
    <asp:Button runat="server" ID="btnPopulations"PostBackUrl="AssociationNatureClientelePopulations.aspx" OnClick="btnPopulations_Click" Text="<%% Resources: GlobalLocalization, Populations %>" CausesValidation="false" />
    <asp:Button runat="server" ID="btnInformations"PostBackUrl="AssociationInformationsNatureClientele.aspx" OnClick="btnInformations_Click" Text="<%% Resources: GlobalLocalization, Informations %>" CausesValidation="false" />
</asp:Content>
```

Création de la nouvelle page

```
<siteMapNode url="Pages/Referentiel/Vente/NaturesClienteleListe.aspx" title="$resources:SiteMapLocalizations,RefVenteNaturesClienteleListe" description="$resources:SiteMapLocalizations,RefVenteNaturesClienteleListeDesc">
<siteMapNode url="Pages/Referentiel/Vente/NatureClienteleDetail.aspx" title="$resources:SiteMapLocalizations,RefVenteNatureClienteleDetail" description="$resources:SiteMapLocalizations,RefVenteNatureClienteleDetailDesc">
<siteMapNode url="Pages/Referentiel/Vente/AssociationProfilsNatureClientele.aspx" title="$resources:SiteMapLocalizations,RefVenteAssociationProfilsNatureClientele" description="$resources:SiteMapLocalizations,RefVenteAsso
<siteMapNode url="Pages/Referentiel/Vente/AssociationNatureClientelePopulations.aspx" title="$resources:SiteMapLocalizations,RefVenteAssociationNatureClientelePopulations" description="$resources:SiteMapLocalizations,RefV
<siteMapNode url="Pages/Referentiel/Vente/AssociationInformationsNatureClientele.aspx" title="$resources:SiteMapLocalizations,RefVenteAssociationInformationsNatureClientele" description="$resources:SiteMapLocalizations,Re
</siteMapNode>
```

Code de la page « AssociationInformationNatureClientele »

```

% Page Language="C#" MasterPageFile="../Master/DesignMasterListe.master" AutoEventWireup="true" CodeFile="AssociationInformationsNatureClientele.aspx.cs" Ir
asp:Content ID="Content1" ContentPlaceHolderID="cphInformation" Runat="Server">
    <asp:Label ID="lblResultat" runat="server" SkinID="lblUpResultat" Visible="false" EnableViewState="false"></asp:Label>

    <table class="collapse" style="padding-bottom:10px;">
        <tr class="bgGrid">
            <td class="tdGridView borderB">
                <asp:Label runat="server" SkinID="lblTitreFavoris" ID="lblTitreIdentifiant" Text="<% Resources: GlobalLocalization, Identifiant %>" />
            </td>
            <td class="tdGridView borderB">
                <asp:Label runat="server" SkinID="lblTitreFavoris" ID="lblTitreRefNatureClientele" Text="<% Resources: GlobalLocalization, Reference %>" />
            </td>
            <td class="tdGridView borderB">
                <asp:Label runat="server" SkinID="lblTitreFavoris" ID="lblTitreLibelleCourt" Text="<% Resources: GlobalLocalization, LibelleCourt %>" />
            </td>
        </tr>
        <tr>
            <td class="tdGridView border">
                <asp:Label runat="server" SkinID="lblBoldFalse" ID="lblIdentifiant" />
            </td>
            <td class="tdGridView border">
                <asp:Label runat="server" SkinID="lblBoldFalse" ID="lblReference" />
            </td>
            <td class="tdGridView border">
                <asp:Label runat="server" SkinID="lblBoldFalse" ID="lblLibelleCourt" />
            </td>
        </tr>
    </table>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="cphFiltre" Runat="Server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="cphListe" Runat="Server">
    <table class="PaddingTab">
        <tr>
            <td>
                <asp:Label runat="server" ID="lblCommentaireTariffR" Text="<% Resources: GlobalLocalization, CommentaireTariffR %>" />
            </td>
            <td>
                <asp:TextBox runat="server" ID="txtInfoFr" MaxLength="1000" Columns="40" />
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label runat="server" ID="lblCommentaireTariffEN" Text="<% Resources: GlobalLocalization, CommentaireTariffEN %>" />
            </td>
            <td>
                <asp:TextBox runat="server" ID="txtInfoEn" MaxLength="1000" Columns="40" />
            </td>
        </tr>
    </table>
</asp:Content>
<asp:Content ID="Content4" ContentPlaceHolderID="cphButtons" Runat="Server">
    <asp:Button ID="btnOK" runat="server" Text="<% Resources: GlobalLocalization, ConfirmerModif %>" CausesValidation="false" OnClick="OnConfirmerModifClick" />
    <asp:Button ID="btnAbandonner" runat="server"PostBackUrl="NatureClienteleDetail.aspx" Text="<% Resources: GlobalLocalization, Abandonner %>" CausesValidation="false" OnClick="OnButtonAbandonnerClick" />
</asp:Content>

```

Méthode GetData

```

protected void getData()
{
    try
    {
        // mise a jour de l'entete
        lblIdentifiant.Text = nature.idnatcl.ToString();
        lblReference.Text = nature.refnatcl;
        lblLibelleCourt.Text = nature.lignatcl;

        DataTable dt = service.GetInformationsNatureClientele(nature.idnatcl.Value);
        UpdateData(dt);
    }
    catch (ExceptionResultat ex)
    {
        SetResultat(ex.Message);
    }
}

```

Page_Load et GetParameters

```
protected void Page_Load(object sender, EventArgs e)
{
    service = new ServiceRefVente();
    if (!IsPostBack)
    {
        nature = new NatureClientele();

        if (!getParameters())
        {
            Response.Redirect("NaturesClienteleListe.aspx");
        }
        getData();
    }
}

protected bool getParameters()
{
    if (PreviousPage != null)
    {
        BasePage pageListe = (BasePage)Page.PreviousPage;

        if (pageListe.OParams.ContainsKey("idnatcl"))
            nature.idnatcl = Convert.ToInt32(pageListe.OParams["idnatcl"].ToString());
        if (pageListe.OParams.ContainsKey("llgnatcl"))
            nature.llgnatcl = pageListe.OParams["llgnatcl"].ToString();
        if (pageListe.OParams.ContainsKey("refnatcl"))
            nature.refnatcl = pageListe.OParams["refnatcl"].ToString();

        return true;
    }
    return false;
}
```

Méthode GetInformationsNatureClientele

```
public DataTable GetInformationsNatureClientele(int idnatcli)
{
    string msgTraceDebut = "GetInformationsNatureClientele(idnatcli=" + idnatcli.ToString() + ")";
    Trace log = new Trace(HttpContext.Current.User.Identity.Name, HttpContext.Current.Session["guid"].ToString());
    DateTime dateTimeDebut = log.TraceDebutSousMethod(application, msgTraceDebut);

    try
    {
        return Ventes.GetInformationsNatureClientele(idnatcli, log);
    }
    catch (Exception ex)
    {
        if (ex is SqlException)
        {
            int CodeErreur = 5700600;
            if (((SqlException)ex).Number > 50000)
                CodeErreur = ((SqlException)ex).Number;

            ExceptionResultat exResultat = new ExceptionResultat();
            exResultat.Message = Localization.EasyResourceManager.getMessageErreur(CodeErreur);
            exResultat.Code = CodeErreur;
            log.TraceError(application, "GetInformationsNatureClientele", exResultat);
            throw exResultat;
        }
        log.TraceError(application, "GetInformationsNatureClientele", ex);
        throw ex;
    }
    finally
    {
        log.TraceFinSousMethod(application, "GetInformationsNatureClientele", dateTimeDebut);
    }
}
```

Création de la table info_natcli

```
IF NOT EXISTS (SELECT * FROM syscolumns AS sc INNER JOIN sysobjects AS so ON (so.id=sc.id) WHERE so.name = 'info_natcli')
BEGIN
    CREATE TABLE info_natcli
    (
        idnatcl INT NOT NULL,
        info_fr VARCHAR(1000) NULL,
        info_en VARCHAR(1000) NULL,
        utilmodi VARCHAR(10) NOT NULL,
        datmodi DATETIME NOT NULL,
        CONSTRAINT PK_info_natcli PRIMARY KEY CLUSTERED
        (
            idnatcl ASC
        )
    )
END
GO
```

Procédure SQL : InformationsNatureClienteleSelect

```
CREATE PROCEDURE InformationsNatureClienteleSelect
    @idnatcl INT
AS
    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
    SET NOCOUNT ON

    -- Récupération de l'enregistrement
    SELECT
        i.idnatcl,
        n.refnatcl,
        n.llcnatcl,
        n.llgnatcl,
        i.info_fr,
        i.info_en
    FROM
        info_natcli AS i
        INNER JOIN natcli AS n ON (i.idnatcl = n.idnatcl)
    WHERE
        i.idnatcl = @idnatcl

    IF (@@ERROR <> 0)
        GOTO ABORT

    RETURN 0

    -- cas d'echec lors de la transaction
ABORT:
    RETURN 1
GO
```

Procédure SQL : InformationsNatureClienteleInsertOrUpdate

```
CREATE PROCEDURE InformationsNatureClienteleInsertOrUpdate
    @idnatcl INT,
    @info_fr VARCHAR(1000),
    @info_en VARCHAR(1000),
    @utilmodi VARCHAR(10)
AS
    BEGIN
        SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
        SET NOCOUNT ON

        -- nettoyage des paramètres
        SET @info_fr = LTRIM(RTRIM(@info_fr))
        SET @info_en = LTRIM(RTRIM(@info_en))

        IF EXISTS(SELECT 0 FROM info_natcli WHERE idnatcl = @idnatcl)
        BEGIN
            -- mise à jour de la table
            UPDATE
                info_natcli
            SET
                info_fr = CASE
                    WHEN ISNULL(@info_fr, '') = '' THEN NULL
                    ELSE @info_fr
                END,
                info_en = CASE
                    WHEN ISNULL(@info_en, '') = '' THEN NULL
                    ELSE @info_en
                END,
                utilmodi = CASE
                    WHEN ISNULL(@utilmodi, '') = '' THEN NULL
                    ELSE @utilmodi
                END,
                datmodi = GETDATE()
            WHERE
                idnatcl = @idnatcl
        END
        ELSE
        BEGIN
            -- création de l'enregistrement
            INSERT INTO info_natcli (idnatcl, info_fr, info_en, utilmodi, datmodi) VALUES (@idnatcl, @info_fr, @info_en, @utilmodi, GETDATE())
        END

        IF @@ERROR <> 0
            GOTO ABORT

        EXECUTE InformationsNatureClienteleSelect @idnatcl = @idnatcl
        RETURN 0
    ABORT:
        RETURN 1
    END
GO
```

Import

```
DECLARE @idmanif INT
SELECT @idmanif = m.idmanif FROM manif as m INNER JOIN grille as g ON g.idmanif = m.idmanif WHERE g.idgril = @idgril
IF NOT EXISTS (SELECT 0 FROM info_natcli_manif as i WHERE i.idmanif = @idmanif AND i.idnatcl = @idnatcl)
BEGIN
    IF EXISTS (SELECT 0 FROM info_natcli WHERE idnatcl = @idnatcl)
    BEGIN
        INSERT INTO info_natcli_manif
        (
            idmanif,
            idnatcl,
            info_fr,
            info_en,
            utilmodi,
            datmodi
        )
        SELECT
            @idmanif,
            @idnatcl,
            i.info_fr,
            i.info_en,
            @utilmodi,
            @dtDate
        FROM
            info_natcli as i
        WHERE
            idnatcl = @idnatcl
    END
END
```

Affichage du code de la fiche manifestation

```
<asp:TableCell>
    <asp:Label ID="lblTarifTempsReel" runat="server" Text="<%= $Resources: GlobalLocalization, TarifTempsReel %>"></asp:Label>
</asp:TableCell>
<asp:TableCell>
    <asp:RadioButton ID="rbTarifTempsReelNon" runat="server" GroupName="TarifTempsReel"
        Text="<%= $Resources: GlobalLocalization, Non %>" />
    <asp:RadioButton ID="rbTarifTempsReelOui" runat="server" GroupName="TarifTempsReel"
        Text="<%= $Resources: GlobalLocalization, Oui %>" />
</asp:TableCell>
```

Autre code de la fiche manifestation

```
// Tarif Temps Reel
if (manifestation.TarifTmpReel.HasValue && manifestation.TarifTmpReel.Value)
{
    rbTarifTempsReelOui.Checked = true;
    rbTarifTempsReelNon.Checked = false;
}
else
{
    rbTarifTempsReelNon.Checked = true;
    rbTarifTempsReelOui.Checked = false;
}
```

ManifestationSelect

```
CREATE Procedure dbo.ManifestationSelect
    @idmanif INT,
    @codutil VARCHAR(10),
    @action BIT = 0
AS
    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
    SET NOCOUNT ON

    DECLARE @datcur DATETIME = GETDATE()

    SELECT
        m.idmanif,
        RTRIM(m.refmanif) as refmanif,
        m.tariftempsreel
    FROM
        manif AS m
    LEFT OUTER JOIN saison_manif AS sm ON (m.idmanif = sm.idmanif)
    LEFT OUTER JOIN maniffavorite AS mf ON (m.idmanif = mf.idmanif AND mf.codutil = @codutil)
    LEFT OUTER JOIN acompteAutoManif AS acm ON (m.idmanif = acm.idmanif)
    INNER JOIN tva AS t ON (t.codtva = m.codtva AND t.codpays = m.codpays AND @datcur BETWEEN t.debval AND ISNULL(t.finval,@datcur))
    WHERE
        m.idmanif = @idmanif
```

Méthode UpdateManifestation

```
public static Manifestation UpdateManifestation(Manifestation manifestation, Trace log)
{
    parameters.Add(new Parameter("@ordfact", SqlDbType.Int, manifestation.ordfact, 4));
    parameters.Add(new Parameter("@nbmseanc", SqlDbType.Int, manifestation.nbmseanc, 4));
    parameters.Add(new Parameter("@envoi", SqlDbType.Int, manifestation.envoi, 4));
    parameters.Add(new Parameter("@acompteAutoManif", SqlDbType.Bit, manifestation.acompteAuto));
    parameters.Add(new Parameter("@datouvte", SqlDbType.DateTime, manifestation.datouvte));
    parameters.Add(new Parameter("@datannonce", SqlDbType.DateTime, manifestation.datannonce));
    parameters.Add(new Parameter("@indism", SqlDbType.Bit, manifestation.indism));
    parameters.Add(new Parameter("@indcalseanc", SqlDbType.Bit, manifestation.indcalseanc));
    parameters.Add(new Parameter("@exclutop", SqlDbType.Bit, manifestation.exclutop));
    parameters.Add(new Parameter("@exporttmintl", SqlDbType.Bit, manifestation.exporttmintl));
    parameters.Add(new Parameter("@cheminvueplace", SqlDbType.VarChar, manifestation.cheminvueplace));
    parameters.Add(new Parameter("@garantiefan", SqlDbType.Bit, manifestation.garantiefan));
    parameters.Add(new Parameter("@orgnat_bi", SqlDbType.Int, manifestation.orgnat_bi));
    parameters.Add(new Parameter("@attridistphy", SqlDbType.Bit, manifestation.attridistphy));
    parameters.Add(new Parameter("@paiementautorise", SqlDbType.Bit, manifestation.paiementAutorise));
    parameters.Add(new Parameter("@zoning", SqlDbType.Bit, manifestation.zoning));
    parameters.Add(new Parameter("@qrcode", SqlDbType.Bit, manifestation.qrcode));
    parameters.Add(new Parameter("@ctrlident", SqlDbType.Bit, manifestation.ctrlident));
    parameters.Add(new Parameter("@codecomptable", SqlDbType.VarChar, manifestation.codecomptable, 20));
    parameters.Add(new Parameter("@datedispobill", SqlDbType.DateTime, manifestation.datedispobill));
    parameters.Add(new Parameter("@fermanticipe", SqlDbType.Bit, manifestation.fermanticipe));
    parameters.Add(new Parameter("@billetdigital", SqlDbType.Bit, manifestation.billetdigital));
    parameters.Add(new Parameter("@transfert", SqlDbType.Int, manifestation.transfert, 4));
    parameters.Add(new Parameter("@authotp", SqlDbType.Int, manifestation.authotp, 4));
    parameters.Add(new Parameter("@cabdiff", SqlDbType.Int, manifestation.cabdiff, 4));
    parameters.Add(new Parameter("@tariftempsreel", SqlDbType.Bit, manifestation.TarifImpReel));

    DataTable dt = new DataTable();
    ExecuteSQL.ExecuteReader("ManifestationUpdate", ref parameters, Constantes.ConnectionString, Constantes.TimeoutEnSeconde, ref dt, "Parametres", "UpdateManifestation", log);
    if (dt.Rows.Count > 0)
    {
        Manifestation manif = ConvertElement.ConvertToManifestation(dt.Rows[0]);
        dt.Dispose();
        return manif;
    }
    return null;
}
```

« ManifestationUpdate » : va vérifier si la valeur a été modifier

```
SELECT
    @gest = gestmanif,
    @orgmnf = orgmanif,
    @ordfct = ISNULL(ordfact, -1),
    @oldticket = ISNULL(oticket,0),
    @oldmticket = ISNULL(mticket,0),
    @olddatedispobill = datedispobill,
    @oldfermantcipe = ISNULL(fermantcipe,0),
    @oldbilletdigital = ISNULL(billetdigital,0),
    @oldllgmanif = llgmanif,
    @olddebmanif = debmanif,
    @oldfinmanif = finmanif,
    @oldcommbill = commbill,
    @oldauthotp = ISNULL(authotp,0),
    @oldcabdiff = ISNULL(cabdiff,0),
    @oldqrcode = ISNULL(qrcode,0),
    @oldtariftempsreel = ISNULL(tariftempsreel,0)
FROM
    manif
WHERE
    idmanif = @idmanif
```

« ManifestationUpdate » : Partie Mise à jour

```
-- Mise-à-jour de la manifestation
UPDATE
    manif
SET
    tariftempsreel = CASE WHEN @tariftempsreel IS NULL THEN tariftempsreel ELSE @tariftempsreel END
```

« ManifestationUpdate » : Déclaration variable

```
DECLARE @oldtariftempsreel BIT = NULL, @tariftempsreel BIT = NULL
```

Test de validation SQL pour le bouton radio

```
SELECT idmanif, utilmodi, tariftempsreel FROM manif WHERE idmanif=539194 ;
```

Résultats		
	idmanif	utilmodi
1	562945	lcelma