LIMONGIELLO LUCAS

Lycée Technique Marie Curie
16 Boulevard Jeanne d'Arc
13005 Marseille

FONDATION MEDITERRANEE INFECTION

Réalisation d'une application Web PARATRACKING

Stage de 1° Année de BTS SIO
du 22 mai au 30 juin 2023

BTS Services Informatiques aux Organisations Option SLAM

Année scolaire 2022-2023

SOMMAIRE

I. STAGE

CONTEXTE

Dans le cadre de ma première année en BTS SIO au Lycée Marie-Curie, je dois réaliser un stage de 6 semaines pour valider mon passage en 2ème année.

Il me permet d'être formé au sein d'une entreprise dans le but d'acquérir des connaissances informatiques, tout en mettant en pratique les connaissances théoriques que j'ai acquises lors de mon année scolaire.

J'ai réalisé au sein du service informatique de l'IHU, une application web nommée PARATRACKING qui doit permettre de tracer les demandes de modifications de paramétrages des automates d'analyses médicales.

PRÉSENTATION DE L'ENTREPRISE

L'AP-HM (Assistance Publique - Hôpitaux de Marseille)

L'AP-HM est l'ensemble hospitalier public qui regroupe les hôpitaux de Marseille.

Il s'agit du plus grand centre hospitalier universitaire (CHU), et est composé de plusieurs Hôpitaux, dont l'Hôpital de la Timone, l'Hôpital Nord, celui de la Conception, l'Hôpital Sainte-Marguerite ou encore l'Hôpital Salvator, ainsi que d'autres structures spécialisées.

L'AP-HM joue un rôle essentiel dans la fourniture de soins médicaux avancés, l'enseignement et la recherche.

Hôpitaux Universitaires de Marseille | ap hm

Présentation de l'IHU (Institut Hospitalo-Universitaire)

L'IHU Méditerranée Infection est un institut spécialisé dans la recherche, le traitement et la prévention contre les maladies infectieuses.

Il collabore avec des partenaires nationaux et internationaux pour mener des recherches et lutter contre les maladies infectieuses.

Il joue un rôle actif dans la formation et l'éducation des professionnels de santé.

II. PRÉSENTATION DU PROJET

PRÉSENTATION DES LOGICIELS

J’ai utilisé le logiciel Balsamiq afin de créer une maquette rapide de l'application afin d’avoir une vue d’ensemble de l’application.

Visual Studio 2019 est un environnement de développement intégré (IDE). Nous l'utiliserons ici afin de développer l'application Web en VB.NET.

J’ai utilisé SQL Server pour créer la base de données de l'application. C’est un système de gestion de base de données relationnelles il nous a permis de stocker, gérer et récupérer efficacement des données.

GESTION DU PROJET

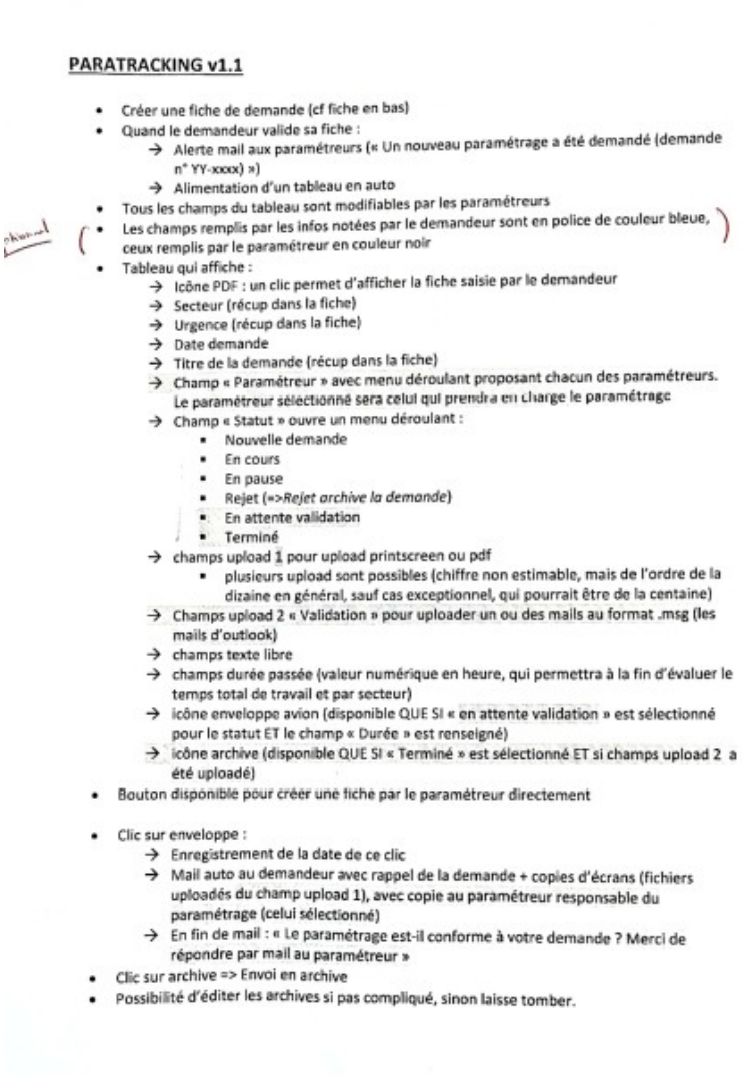
Méthode AGILE

La méthode prend la forme de réunions hebdomadaires durant lesquelles nous devons valider les développements réalisés et fixer les objectifs de la semaine à venir. L'espace entre les deux réunions est appelé un sprint car c'est le moment où on doit développer les fonctionnalités attendues.

Besoin fonctionnel de l'application web

Lors de la première réunion nous avons défini les besoins de l'utilisateur. Il s'agissait du responsable qualité en charge du suivi du paramétrage des automates. L'application attendue doit donc lui permettre de suivre précisément les demandes, la réalisation et le suivi des modifications réalisées.

Parmi les éléments nécessaires au suivi, on doit calculer le temps pris par la modification. L'ensemble des besoins de l'utilisateur ont été définis dans le cahier des charges ci-dessous.

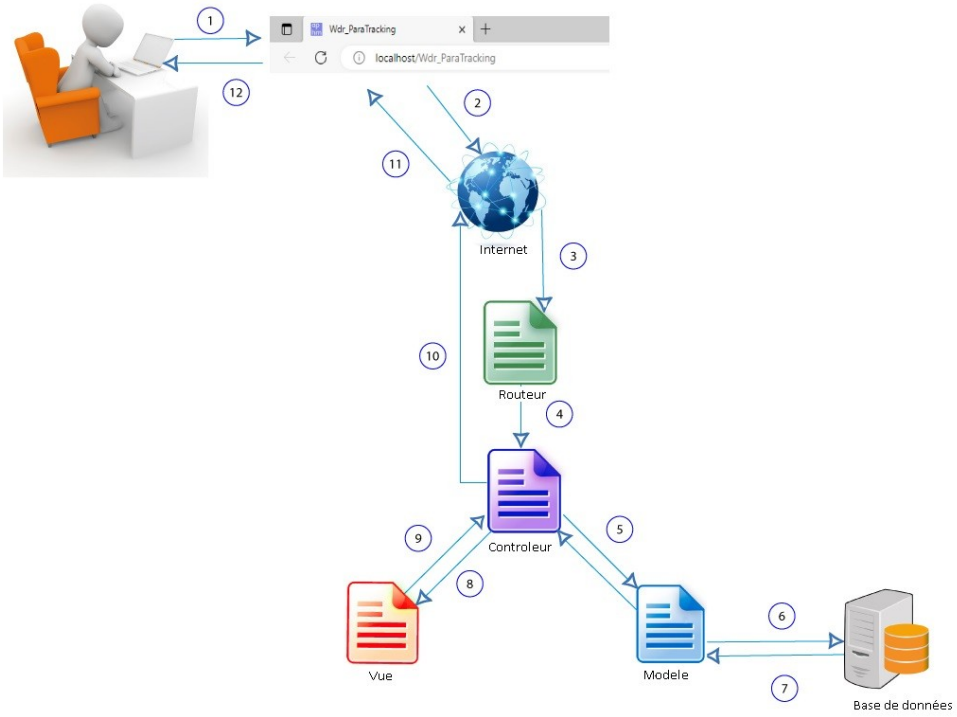


PARATRACKING v1.1

- Créer une fiche de demande (cf fiche en bas)
- Quand le demandeur valide sa fiche :
  - → Alerte mail aux paramétreurs (« Un nouveau paramétrage a été demandé (demande n° YY-xxxx) »)
  - → Alimentation d'un tableau en auto
- Tous les champs du tableau sont modifiables par les paramétreurs
- Les champs remplis par les infos notées par le demandeur sont en police de couleur bleue, ceux remplis par le paramétreur en couleur noir
- Tableau qui affiche :
  - → Icône PDF : un clic permet d'afficher la fiche saisie par le demandeur
  - → Secteur (récup dans la fiche)
  - → Urgence (récup dans la fiche)
  - → Date demande
  - → Titre de la demande (récup dans la fiche)
  - → Champ « Paramétreur » avec menu déroulant proposant chacun des paramétreurs. Le paramétreur sélectionné sera celui qui prendra en charge le paramétrage
  - → Champ « Statut » ouvre un menu déroulant :
    - Nouvelle demande
    - En cours
    - En pause
    - Rejet (=>*Rejet archive la demande*)
    - En attente validation
    - Terminé
  - → champs upload 1 pour upload printscreen ou pdf
    - plusieurs upload sont possibles (chiffre non estimable, mais de l'ordre de la dizaine en général, sauf cas exceptionnel, qui pourrait être de la centaine)
  - → Champs upload 2 « Validation » pour uploader un ou des mails au format .msg (les mails d'outlook)
  - → champs texte libre
  - → champs durée passée (valeur numérique en heure, qui permettra à la fin d'évaluer le temps total de travail et par secteur)
  - → icône enveloppe avion (disponible QUE SI « en attente validation » est sélectionné pour le statut ET le champ « Durée » est renseigné)
  - → icône archive (disponible QUE SI « Terminé » est sélectionné ET si champs upload 2 a été uploadé)
- Bouton disponible pour créer une fiche par le paramétreur directement
- Clic sur enveloppe :
  - → Enregistrement de la date de ce clic
  - → Mail auto au demandeur avec rappel de la demande + copies d'écrans (fichiers uploadés du champ upload 1), avec copie au paramétreur responsable du paramétrage (celui sélectionné)
  - → En fin de mail : « Le paramétrage est-il conforme à votre demande ? Merci de répondre par mail au paramétreur »
- Clic sur archive => Envoi en archive
- Possibilité d'éditer les archives si pas compliqué, sinon laisse tomber.

III. MVC

SCHÉMA DU MVC



LES DIFFÉRENTS LANGAGES UTILISÉS

Étapes 1, 2, 3 :	Http
Étapes 4, 5, 8 :	VisualBasic.Net
Étapes 6, 7:	Linq
Étapes 9 :	Razor
Étapes 10 et 11 :	HTML/CSS/JavaScript
Étapes 12 :	Page web

EXPLICATION DU MVC

Le client via une requête http demande l'affichage d'une page. Cette demande est interceptée par le routeur qui redirige la demande vers le contrôleur demandé. Si aucun contrôleur n'est précisé dans la demande alors le routeur retourne le contrôleur par défaut.

```
Public Module RouteConfig
    1 référence
    Public Sub RegisterRoutes(ByVal routes As RouteCollection)
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}")

        routes.MapRoute(
            name:="Default",
            url:="{controller}/{action}/{id}",
            defaults:=New With {.controller = "Demandes", .action = "IndexEnAttente", .id = UrlParameter.Optional}
        )
    End Sub
```

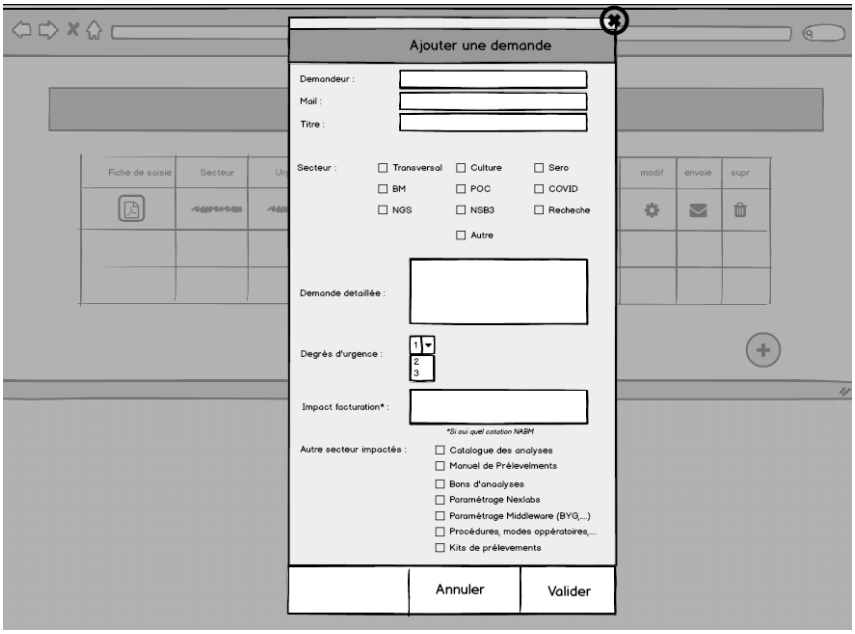
Le contrôleur, codé en VB.NET fait appel au modèle, pour récupérer les données dans la base de données grâce à une requête Linq. Les données de la base sont retournées au contrôleur. Il fait ensuite appel à la vue codée en Razor contenant du HTML/CSS pour la présentation des données. Le contrôleur fusionne ensuite les données et la présentation grâce au render et retourne le résultat au format HTML/CSS au client.

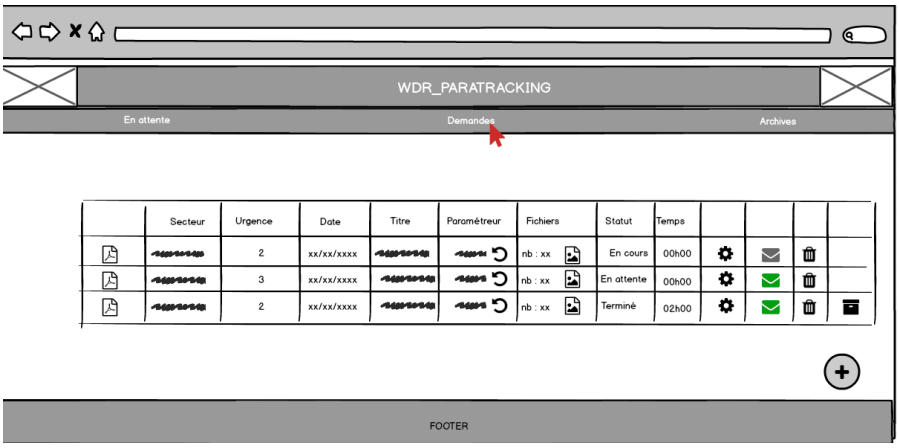
IV. RÉALISATION DU PROJET

MAQUETTE

J'ai réalisé une maquette avec le logiciel Balsamiq lors de la conception de l'application afin qu'elle corresponde au cahier des charges.

La maquette a pour but de donner un aperçu graphique de l'application à l'utilisateur et couvre la totalité des fonctionnalités. Voici un exemple :

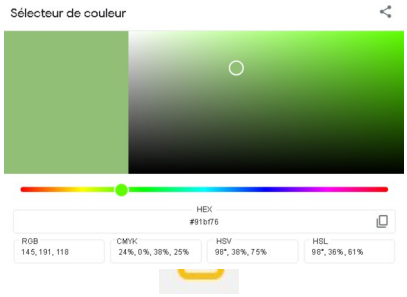




CHARTE GRAPHIQUE

Une charte graphique doit être respectée et a été mise en place par mon maître de stage sous forme de Template, son utilisation étant prévue exclusivement sur ordinateur il n'est pas nécessaire que l'application soit responsive.

Couleur de l'application :



Update :

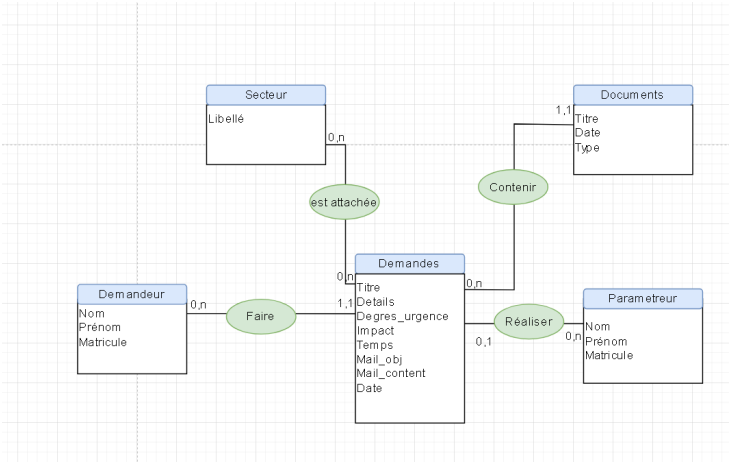
Delete :

Add :

Header/Footer :

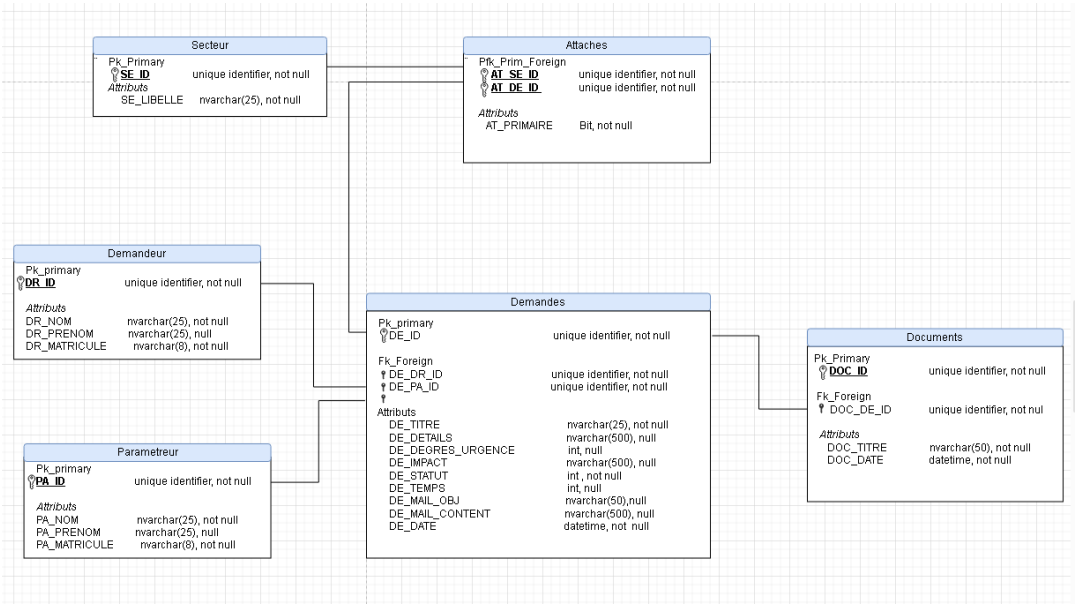
LA BASE DE DONNÉES (BDD)

Modèle conceptuel des données (MCD)



Le MCD est une description graphique simplifiée qui permet de représenter des modèles de données sous la forme de diagrammes. Ceux-ci contiennent des entités représentées sous forme de carrés et des associations représentées sous forme d'ellipse contenant un verbe décrivant la nature de la relation.

Modèle physique des données (MPD)



Le MPD est la transformation physique du MCD. Les entités sont transformées en tables, les relations disparaissent. En fonction des cardinalités les clés primaires de certaines tables deviennent des clés étrangères d'autres tables.

Requêtes de création de la BDD

Table PT_DEMANDES

```
CREATE TABLE [dbo].[PT_DEMANDES](
    [DE_ID] [uniqueidentifier] NOT NULL,
    [DE_DEMANDEUR_NOM] [nvarchar](50) NOT NULL,
    [DE_DEMANDEUR_MAIL] [nvarchar](50) NULL,
    [DE_PARAMETREUR_ID] [nvarchar](8) NULL,
    [DE_TITRE] [nvarchar](50) NOT NULL,
    [DE_DETAILS] [nvarchar](500) NULL,
    [DE_DEGRES_URGENCE] [int] NOT NULL,
    [DE_STATUT] [int] NOT NULL,
    [DE_TEMPS] [int] NULL,
    [DE_MAIL_OBJ] [nvarchar](50) NULL,
    [DE_MAIL_CONTENT] [nvarchar](500) NULL,
    [DE_MAIL_DATE] [datetime] NULL,
    [DE_DATE] [datetime] NOT NULL,
    [DE_REMARQUE] [nvarchar](500) NULL,
    [DE_MOTIF_ARRET] [nvarchar](500) NULL,
 CONSTRAINT [PK_PT_DEMANDES] PRIMARY KEY CLUSTERED
(
    [DE_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Table PT_DOCUMENTS

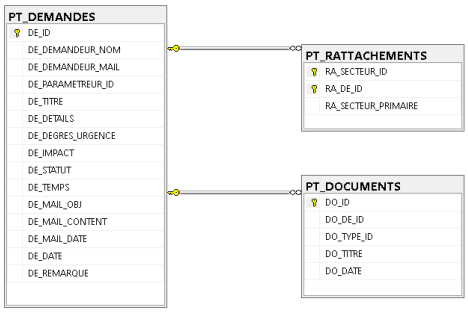
```
CREATE TABLE [dbo].[PT_DOCUMENTS](
    [DO_ID] [uniqueidentifier] NOT NULL,
    [DO_DE_ID] [uniqueidentifier] NOT NULL,
    [DO_TYPE_ID] [uniqueidentifier] NOT NULL,
    [DO_FICHIER] [text] NOT NULL,
    [DO_TITRE] [nvarchar](50) NOT NULL,
    [DO_DATE] [datetime] NOT NULL,
    [DO_POIDS] [int] NOT NULL,
 CONSTRAINT [PK_PT_DOCUMENTS] PRIMARY KEY CLUSTERED
(
    [DO_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

Table PT_DOCUMENTS

```
CREATE TABLE [dbo].[PT_RATTACHEMENTS](
    [RA_SECTEUR_ID] [uniqueidentifier] NOT NULL,
    [RA_DE_ID] [uniqueidentifier] NOT NULL,
    [RA_SECTEUR_PRIMAIRE] [bit] NOT NULL,
 CONSTRAINT [PK_PT_RATTACHEMENTS] PRIMARY KEY CLUSTERED
(
    [RA_SECTEUR_ID] ASC,
    [RA_DE_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Résultat

Voici donc à quoi ressemble la base de données :



Les tables paramétreurs, demandeurs, secteurs ont disparu car ces données sont stockées dans une autre base de données et seront récupérées via un web service de type SOAP.

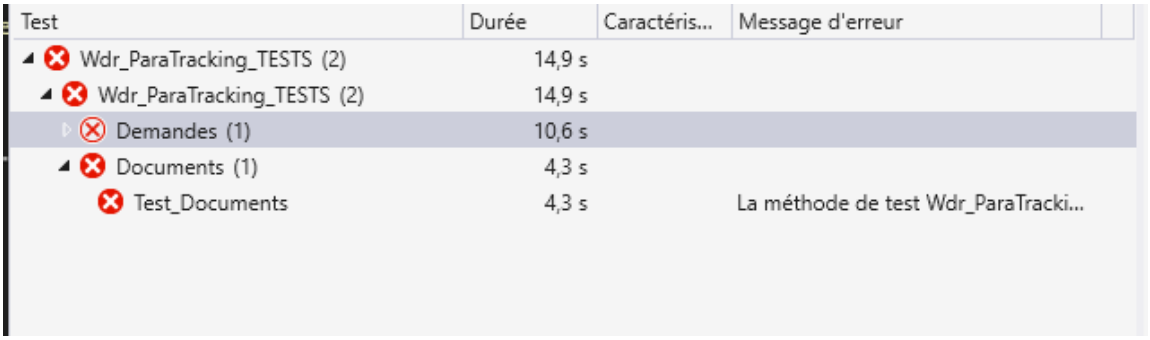
La base de données finale ne correspond pas exactement à ce schéma car on a dû y ajouter certaines informations pour s'adapter aux évolutions des demandes de notre client.

Programmation des Tests (TDD)

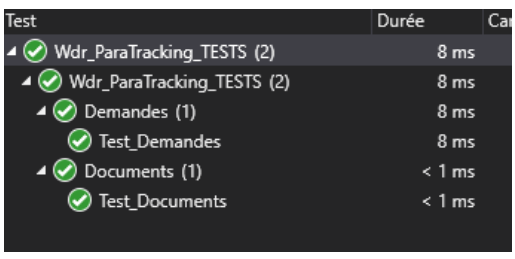
Philosophie du TDD

La méthode TDD (Test Driving Development) vise à développer les tests en premier avant même le développement de la couche d'accès aux données.

Une fois les tests développés ils seront planifiés régulièrement. Au fur et à mesure du développement des fonctionnalités les tests en erreurs passent au statut OK. L'avantage de cette méthode est de permettre au chef de projet de suivre l'avancement du développements très simplement.



Une fois toutes les fonctions développées l'ensemble des tests unitaires passent en vert.



Test Unitaire

Le Test Unitaire est composé d'un Select, d'un Insert, d'un Update et d'un Delete.

Il a pour but de tester les différentes fonctions codées afin de vérifier qu'il n'y a aucun bug. Il doit pouvoir interagir avec la base de données et la laisser dans son état initial une fois le test terminé.

Exemple pour la table des demandes

```
Dim MyGlobalProvider As New GlobalProvider
Dim MyDocuments As List(Of Document)
Dim MyUniqueId As Guid
Dim MyDocument As Document
'
MyDocuments = MyGlobalProvider.Documents.GetAll()
Assert.IsNull(MyDocuments)
'
MyUniqueId = MyGlobalProvider.Documents.InsertOne(Guid.NewGuid, Guid.Parse("AA897531-08BD-4D4D-8CD6-38E90B196436"),
MyDocument = MyGlobalProvider.Documents.GetOne(MyUniqueId)
Assert.IsNotNull(MyDocument)
Assert.AreEqual("titre de ma demande", MyDocument.Titre)

MyGlobalProvider.Documents.UpdateOne(MyUniqueId, Guid.Parse("AA897531-08BD-4D4D-8CD6-38E90B196436"), Guid.Parse("AA8
MyDocument = MyGlobalProvider.Documents.GetOne(MyUniqueId)
Assert.IsNotNull(MyDocument)
Assert.AreEqual("titre2", MyDocument.Titre)
'
MyDocuments = MyGlobalProvider.Documents.GetAll()
Assert.AreEqual(1, MyDocuments.Count)

MyGlobalProvider.Documents.DeleteOne(MyUniqueId)
MyDocument = MyGlobalProvider.Documents.GetOne(MyUniqueId)
Assert.AreEqual(Nothing, MyDocument)
'
```

Dans l'exemple ci-dessus on teste le Select, l'Insert, l'Update et le Delete, grâce aux fonctions du même nom qui seront créés par la suite.

Couche D'accès Aux Données (Datalayer)

Le Provider est crée afin de regrouper l'ensemble des fonctions permettant l'accès aux données.

Pour le provider des documents une fonction **GetAll()** a été développée. Elle permet de récupérer l'ensemble des documents de la base sous forme de liste parcourable d'objet document. Pour cela on réalise une requête Linq qui permet de récupérer les données dans la base. Celle-ci est l'équivalent de la requête SQL *« SELECT * FROM PT_DOCUMENTS ».*

```
Public Function GetAll() As List(Of Document)
    Try
        '
        Return BuildAll(From obj In MyDataContext.PT_DOCUMENTS)
        '
    Catch ex As Exception
        Throw ex
    End Try
End Function
```

Elle fait appel à la fonction **BuildAll()** qui se charge de transformer une liste. parcourable de type PT_DOCUMENTS en liste parcourable de type document.

```
Private Function BuildAll(pValues As IEnumerable(Of PT_DOCUMENTS)) As List(Of Document)
    Try
        Dim MyReturn As List(Of Document)
        If Not pValues Is Nothing AndAlso pValues.Count > 0 Then
            MyReturn = New List(Of Document)
            '
            For Each pValue As PT_DOCUMENTS In pValues
                MyReturn.Add(BuildOne(pValue))
            Next
        End If
        '
        Return MyReturn
```

Pour chaque élément contenu dans la liste parcourable de PT_DOCUMENTS la fonction **BuildAll()** appelle la fonction **BuildOne()**. Celle-ci a pour objectif de transformer un élément de type PT_DOCUMENTS en 1 élément de type document.

```
Private Function BuildOne(pValue As PT_DOCUMENTS) As Document
    Try
        If Not pValue Is Nothing Then
            Return New Document(pValue)
        Else
            Return Nothing
        End If
    Catch ex As Exception
        Throw ex
    End Try
```

La fonction **BuildOne()** utilise le constructeur de la classe document pour transformer un PT_DOCUMENTS en une instance d'un objet document.

```
2 rererences
Public Sub New(pDocument As PT_DOCUMENTS)
    '
    _UniqueId = pDocument.DO_ID
    _DemandeId = pDocument.DO_DE_ID
    _TypeId = pDocument.DO_TYPE_ID
    _Titre = pDocument.DO_TITRE
    _DateCreation = pDocument.DO_DATE
    '
End Sub
```

Le constructeur crée une nouvelle instance de la classe document et affecte toutes les valeurs reçues dans le paramètre de type PT_DOCUMENTS aux propriétés de la classe Documents. Pour cela j'ai dû développer des accesseurs de type GET et SET.

```
Private _UniqueId As Guid
1 référence
Public Property UniqueId As Guid
    Get
        Return _UniqueId
    End Get
    Set(value As Guid)
        _UniqueId = value
    End Set
End Property
```

DESCRIPTION DU FONCTIONNEMENT MVC

Exemple pour l'affichage de la liste des demandes en attente

Lorsque le routeur reçoit une demande d'affichage des demandes en attente il se redirige donc vers le contrôleur des demandes et sur la fonction *IndexEnAttente*. Celle-ci retourne la vue Index qui se trouve dans le sous répertoire en attente du dossier demande.

```
0 références
Function IndexEnAttente() As ActionResult
    Return View("En_Attente/Index")
End Function
```

La vue index contient une zone HTML dans laquelle on va afficher la liste des demandes en attente.

```
<div class="px-2 px-xl-4 " style="margin-bottom: 5rem">
    <br />

    <div Class="d-flex flex-column justify-content-center DataTables_Container mb-10" style="position:absolute;top:-9999px;left:-9999px">
        <div id="Display_Results" Class="table-responsive">
            @*Remplissage dynamique avec la fonction Javascript "Display_Results"*@
        </div>
    </div>

    <div class="divButtonsFixed justify-content-end">
        <Button Class="btnRound btnAdd btnMedium" onclick="ShowModalCreate()" title="Nouvelle demande"><i class="fa-solid fa-plus"></i></Bu
    </div>
</div>
```

Lorsque le document est chargé une fonction Javascript permet de réaliser un appel Ajax vers le contrôleur des demandes pour obtenir la liste des demandes en attente. Celles-ci seront affichées dans la zone prévue grâce à la fonction ShowDataTables.

```
20  @section scripts
21      <script>
22
23          $(document).ready(function () {
24              /**/
25              Display_Results();
26              /**/
27          });
28
29          function Display_Results() {
30              $.ajax({
31                  url: '@Url.Action("Display_List_EnAttente", "Demandes")',
32                  type: 'GET',
33                  data: {  },
34                  success: function (result) {
35                      ShowDataTables(result);
36                  },
37                   error: function (request, error) { ShowPopupOnError(request, error) }
38              });
39          }
```

Le contrôleur des demandes contient une fonction Display_List_EnAttente qui retourne les données obtenues par le modèle via le contrôleur des demandes fusionnées avec la présentation contenue dans le fichier Display_List présent dans le sous répertoire en attente.

```
Function Display_List_EnAttente() As ActionResult
    '
    Try
        ViewBag.StepBack = "IndexEnAttente"
        Return PartialView("En_Attente/Display_List", MyGlobalProvider.Demandes.GetAllEnAttente)
        '
    Catch ex As Exception
        Throw ex
    End Try
    '
End Function
```

Le Provider des demandes contient la méthode **GetAllEnAttente()**

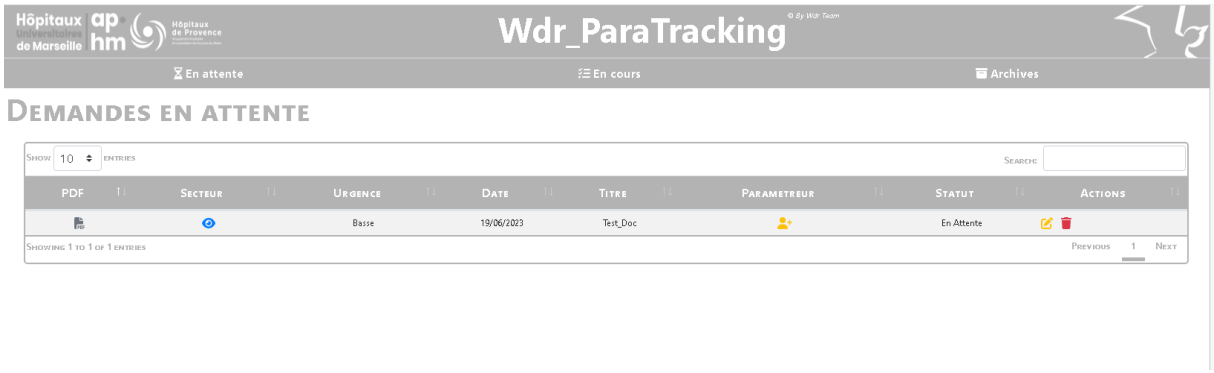
```
Public Function GetAllEnAttente() As List(Of Demande)
    Try
        '
        Return BuildAll(From obj In MyDataContext.PT_DEMANDES_EN_ATTENTE)
        '
    Catch ex As Exception
        Throw ex
    End Try
End Function
```

La vue *Display_List* permet d'organiser la présentation des données grâce a langage HTML/CSS mixée avec du code VB. Le passage de l'un à l'autre se fait grâce au @.

```
<thead>
    <tr Class="h6" value="NA">
        <th>
            @" PDF "@ PDF
        </th>
        <th>
            @" Secteur "@ Secteur
        </th>
        <th>@Html.DisplayNameFor(Function(model) model.Degres)</th>
        <th>@Html.DisplayNameFor(Function(model) model.DeDate)</th>
        <th>@Html.DisplayNameFor(Function(model) model.Titre)</th>
        <th>@Html.DisplayNameFor(Function(model) model.ParametreurId)</th>
        <th>@Html.DisplayNameFor(Function(model) model.Statut)</th>
        <th>Actions</th>
    </tr>
</thead>
<tbody>
    @If Not Model Is Nothing Then
        @For Each item In Model
            @<tr class="inputChange" style="height:2.5em" value="@item.UniqueID">
                <td>
                    @" PDF "@ <Button type="button" onclick="Display_PDF('@item.UniqueId')" Class="btn"><i class="fa-solid fa-file-pdf text-secondary"></i></Button
                </td>
                <td>
                    <Button type="button" onclick="Display_Secteurs('@item.UniqueId')" Class="btn"><i class="fa-solid fa-eye text-primary"></i></Button>
                </td>
                <td>@item.Degres_Display</td>
                <td>@item.DeDate_Display</td>
                <td>@item.Titre</td>
                <td>
                    @item.ParametreurId
                    <Button type="button" onclick="Add_Parametreur('@item.UniqueId')" Class="btn"><i class="fa-solid fa-user-plus text-warning"></i></Button>
                </td>
                <td>@item.Statut_Display</td>

                <td style="width:10px;text-align:left" class="pl-2">
                    <Button type="button" onclick="ShowModalEdit('@item.UniqueId')" Class="btn"><i class="fa-solid fa-pen-to-square text-warning"></i></Button>
                    <Button type="button" onclick="ShowModalDelete('@item.UniqueId')" Class="btn"><i Class="fa-solid fa-trash text-danger"></i></Button>
                </td>
```

La page HTML obtenue est ensuite retournée au client en réponse à sa demande http.



Exemple pour l'ajout d'un parametreur

Lorsque l'utilisateur clique sur le bouton pour ajouter un parametreur.


```
<td>
    @item.ParametreurId
    <Button type="button" onclick="Add_Parametreur('@item.UniqueId')" Class="btn"><i class="fa-solid fa-user-plus text-warning"></i></Button>
</td>
```

Il déclenche une fonction JavaScript qui fait un appel Ajax.

```
<script>
    function Add_Parametreur(uniqueID) {
        $.ajax({
            url: '@Url.Action("Add_Parametreur", "Utilisateurs")',
            type: 'GET',
            data: { pUniqueID: uniqueID },
            success: function (result) { ShowPopupOnSuccess(result) },
            error: function (request, error) { ShowPopupOnError(request, error) }
        });
    }
```

La fonction AddParametreur demande au modèle de récupérer tous les paramétreurs grâce à la fonction GetAllParametreurs.

```
    ' GET: Utilisateurs
    <HttpGet()>
    0 références
    Function Add_Parametreur(pUniqueId As Guid) As ActionResult
        Try
            ViewBag.MyDemandeId = pUniqueId
            ViewBag.Parametreurs = MyGlobalProvider.Utilisateurs.GetAllParametreurs()
            Return PartialView("Add_Parametreur")
        Catch ex As Exception
            Throw ex
        End Try
```

Celle-ci fait un appel au Webservices WS_Matrices qui permet de récupérer l'ensemble des utilisateurs de l'application qui possèdent un profil parametreur identifié par la chaîne passée en paramètre de la fonction **GetAllByProfilId()**.

```
1 référence
Public Function GetAllParametreurs() As List(Of Ws_Matrices.Matrice)
    Try
        '
        Return GetAllByProfilId(Guid.Parse("2F2AD8FD-77DC-4867-AD3A-39FFEF49389A"))
        '
    Catch ex As Exception
        Throw ex
    End Try
```

Cette vue affiche une Liste déroulante et contiendra les noms des paramétreurs sous la forme d'une pop-up.

```
@*<dt>@Html.LabelFor(Function(model) model.SystemeId, htmlAttributes:=New With {.Class = "form-label-for-edit"})</dt>*@
<dd id = "MyParametreurs" >@Html.DropDownList("Parametreur", New SelectList(CType(ViewBag.Parametreurs, IEnumerable), "CompteNT", "nomAffiche"), New With
@*@Html.ValidationMessageFor(Function(model) model.SystemeId, "", New With {.class = "text-danger"})*@
```

Le résultat sera affiché de la manière suivante :

Lorsque l'utilisateur clique sur le bouton sauvegarder il soumet le formulaire grâce à une méthode en post.

```
<Button type = "submit" Class="btn @("My-PopupButtonSave" + ViewData("ThemeSuffixe"))">Sauvegarder</Button>
```

Le contrôleur des demandes contient une méthode **Add_Parametreur()** spécifique aux appels de type post. Celle-ci reçoit en paramètre l'identifient unique de la demande ainsi que le nom du parametreur.

```
        <HttpPost()>
        0 références
        Function Add_Parametreur(pUniqueId As Guid, Parametreur As String) As ActionResult
            Try
                Dim MyStatut As Integer = IIf(Not Parametreur = "", 2, 1)
                MyGlobalProvider.Demandes.Update_Parametreur(pUniqueId, Parametreur, MyStatut)
                Return RedirectToAction("IndexEnAttente", "demandes")
            Catch ex As Exception
                Throw ex
            End Try
        End Function

    End Class
```

Le contrôleur appelle dans le modèle des demandes la fonction **Update_Parametreur()** qui met à jour le parametreur ainsi que le statut de la demande. Et applique les changements dans la base de données.

```
1 référence
Public Function Update_Parametreur(pUniqueId As Guid, pParametreurId As String, pStatut As Integer)
    '
    Try
        Dim MyObject As IEnumerable(Of PT_DEMANDES) = From obj In MyDataContext.PT_DEMANDES Where obj.DE_ID = pUniqueId
        '
        If Not MyObject Is Nothing AndAlso MyObject.Count = 1 Then
            '
            MyObject.First.DE_PARAMETREUR_ID = pParametreurId
            MyObject.First.DE_STATUT = pStatut
            '
            MyDataContext.SubmitChanges()
            '
            Return True
        Else
            Return False

        End If

    Catch ex As Exception

        Throw ex
```

L'utilisateur est ensuite redirigé vers la page des demandes en attente.

```
Return RedirectToAction("IndexEnAttente", "demandes")
```

V. REMERCIEMENTS

Durant ma première année de BTS SIO, j'ai effectué un stage dans le service informatique de l'institut de la fondation Méditerranée Infection 19-21 boulevard Jean Moulin 13005 Marseille.

Je tiens tout d'abord à remercier tout le personnel de la fondation Méditerranée Infection pour son accueil chaleureux, son soutien tout au long de mon stage et les diverses connaissances qu'ils ont partagées avec moi durant cette période.

Je tiens ensuite à remercier tout particulièrement mon tuteur M. STOUPAN Didier du temps qu'il a consacré à ma formation, qui m'a épaulé et conseillé et qui m'a surtout transmis son expertise dans le domaine de l'informatique. L'aide dont il a fait preuve à mon égard m'a permis de pleinement mettre à profit mon stage à l'institut.

Je tiens également à remercier Kévin stagiaire de première année de Master en Informatique pour sa disponibilité et son aide.

Enfin, je remercie les enseignants du Lycée Marie Curie M. DEMEDES et M. MICHAUD professeurs d'informatique qui m'ont permis de compléter ma formation avec leurs cours. Ces connaissances m'ont permises d'être performant lors de mon stage en entreprise.

VI. CONCLUSION

Ce stage m'est apparu comme une expérience très satisfaisante et enrichissante. J'ai pu grâce à ce stage, améliorer mes connaissances en informatique avec notamment la notion de programmation objet en VB.NET et la méthode Agile. En étant confronté à des professionnels j'ai pu élargir mes connaissances.

Le projet en lui-même a également représenté une bonne expérience pour moi. En effet la programmation de l'application Patratacking est une tendance de plus en plus adoptée par les entreprises de par ses nombreux avantages.

Le résultat de ce stage est donc une application réalisée dans le respect de plusieurs règles de bonnes pratiques de programmation web, qui permettent d'optimiser l'application.

De plus cela m'a permis de conforter mon choix de parcours d'étude.

VII. ATTESTATION DE STAGE

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS **SESSION 2023**

Logo de l'organisme d'accueil

ATTESTATION DE STAGE
1° année BTS SIO

ORGANISME D'ACCUEIL

Nom ou dénomination sociale : Fondation de Coopération Scientifique IHU MÉDITERRANÉE INFECTION

Adresse : 19-21 bd Jean Moulin 13005 Marseille - France

N° Siret : 501 980 882 00028

TVA - FR 04 501 980 882

☏ : Tél. : 04 13 73 22 11

certifie que

LA OU LE STAGIAIRE

Nom : Limongiello Prénom : Lucas

Né(e) le : 04 / 03 / 2004 Sexe : F ☐ M ☒

Adresse : 1 Rue Georges Guynemer Plan-de-cuques 13380

☏ : 07 60 41 09 33 Mél : Lucaslimongiello@gmail.com

ÉTUDIANT(E) EN BTS Services informatiques aux organisations

Option ☐ **SISR** ☒ **SLAM**

AU SEIN DE (nom de l'établissement d'enseignement supérieur ou de l'organisme de formation) :

a effectué un stage prévu dans le cadre de ses études

DURÉE DU STAGE

Dates de début et de fin du stage : **du 22/05/2023 au 30 /06 /2023**.

Représentant une **durée totale de 6 semaines.**

La durée totale du stage est appréciée en tenant compte de la présence effective de la ou du stagiaire dans l'organisme, sous réserve des droits à congés et autorisations d'absence prévus à l'article L.124-13 du code de l'éducation (art. L.124-18 du code de l'éducation). Chaque période au moins égale à 7 heures de présence consécutives ou non est considérée comme équivalente à un jour de stage et chaque période au moins égale à 22 jours de présence consécutifs ou non est considérée comme équivalente à un mois.

MONTANT DE LA GRATIFICATION VERSÉE Á LA OU AU STAGIAIRE

La ou le stagiaire a perçu une gratification de stage pour un **montant total** de ..... euros.

L'attestation de stage *est indispensable pour pouvoir, sous réserve du versement d'une cotisation, faire prendre en compte le stage dans les droits à retraite. La législation sur les retraites (loi n°2014-40 du 20 janvier 2014) ouvre aux étudiants* ***dont le stage a été gratifié*** *la possibilité de faire valider celui-ci dans la* ***limite de deux trimestres****, sous réserve du versement d'une cotisation.* ***La demande est à faire par l'étudiant(e) dans les deux années*** *suivant la fin du stage et sur* ***présentation obligatoire de l'attestation de stage*** *mentionnant la durée totale du stage et le montant total de la gratification perçue. Les informations précises sur la cotisation à verser et sur la procédure à suivre sont à demander auprès de la Sécurité sociale (code de la Sécurité sociale art. L.351-17 – code de l'éducation art. D.124-9)*

Fait à ..... **le** ... / ... / .....

Nom, fonction et signature de la personne représentant de l'organisme d'accueil

Fondation de Coopération Scientifique
IHU
MÉDITERRANÉE INFECTION
19-21 bd Jean Moulin
13005 Marseille - France
N° Siret : 501 980 882 00028
TVA - FR 04 501 980 882
Tél. : 04 13 73 22 11