



IT TELEFAUNE

Annuaire en ligne pour professionnels

Par Maxence Vandekerckhove

Sommaire

I \ Conception du fonctionnement de l'application.	3
II \ Organisation des données et Modèles Pré-base.	4
III \ Architecture de l'application et création du GIT.....	6
1 \ Application de développement.....	6
2 \ Base de données	7
3 \ GitHub et partage du projet.....	8
IV \ Technologies utilisées.....	8
1 \ Razor.	8
2 \ Entity Framework et MySQL	9
V \ Problèmes rencontrées.	10
VI \ Améliorations possible.	11
VII \ Conclusion	11

I \ Conception du fonctionnement de l'application.

Dans le cadre de notre projet, nous réaliserons une application web permettant aux utilisateurs de consulter un annuaire d'utilisateurs pour la société IT TeleFaune.

Cette entreprise est basée dans l'agroalimentaire et s'étend sur l'ensemble du territoire français. De cela découle le besoin de pouvoir obtenir les coordonnées des salariés des différents sites afin de faciliter la communication.

Notre cahier des charges se décompose alors de cette manière :

Tout d'abord en tant que visiteur, vous aurez accès à plusieurs options à savoir :

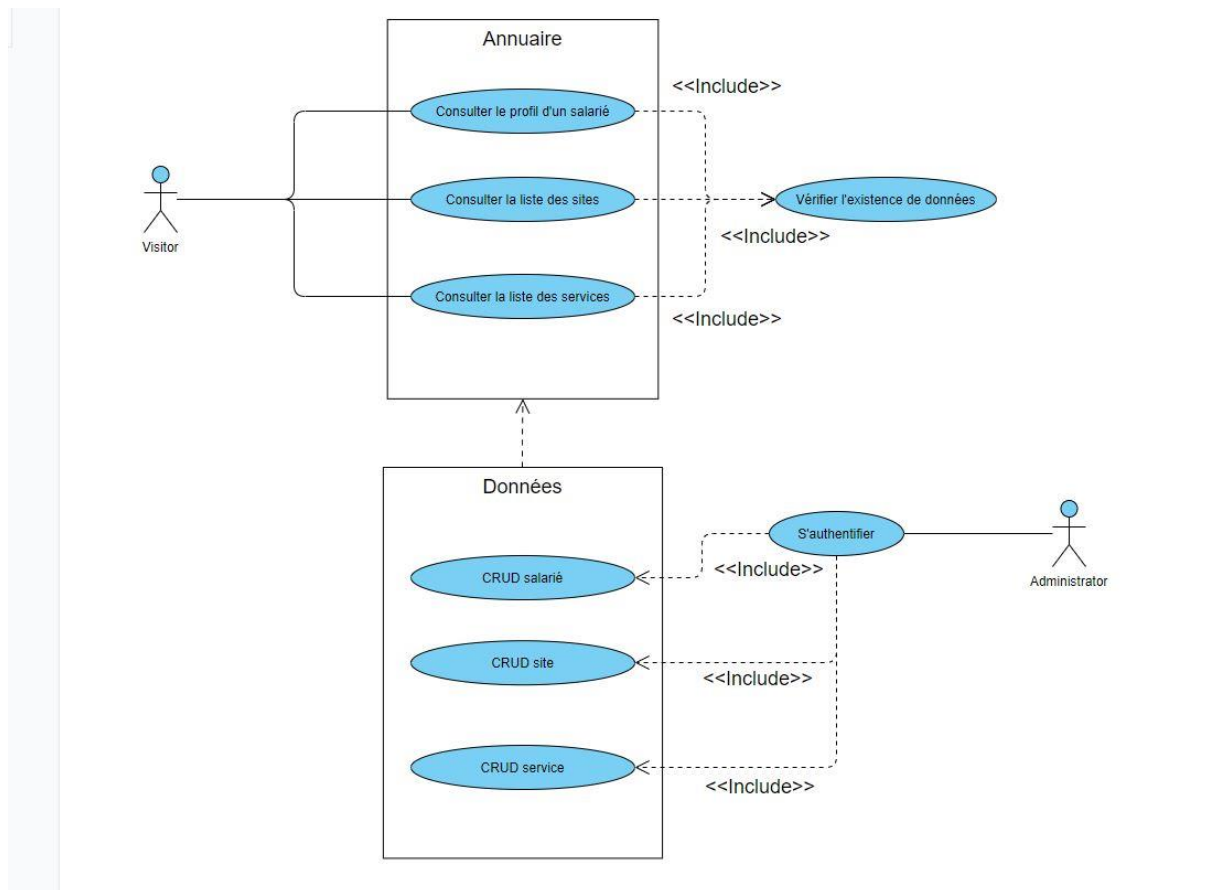
- La consultation du profil d'un salarié à partir d'une barre de recherche
- La consultation de la liste des salariés d'un site
- La consultation de la liste des salariés d'un service
- La consultation de la liste des sites
- La consultation de la liste des services

Une seconde partie présente sur le site permettra la connexion à un espace privé nommé espace administrateur.

L'administrateur pourra se connecter via une interface d'authentification avec son adresse électronique ainsi que son mot de passe.

En tant qu'administrateur vous aurez accès aux droits suivants :

- Création d'une fiche profil de salarié
- Modification d'une fiche profil de salarié
- Suppression d'une fiche profil de salarié
- Création d'un site
- Modification d'un site
- Suppression d'un site
- Création d'un service
- Modification d'un service
- Suppression d'un service
- Création d'un compte pour administrateur (à déterminer)
- Modification d'un compte pour administrateur (à déterminer)
- Suppression d'un compte pour administrateur (à déterminer)



Use case Diagram of the website

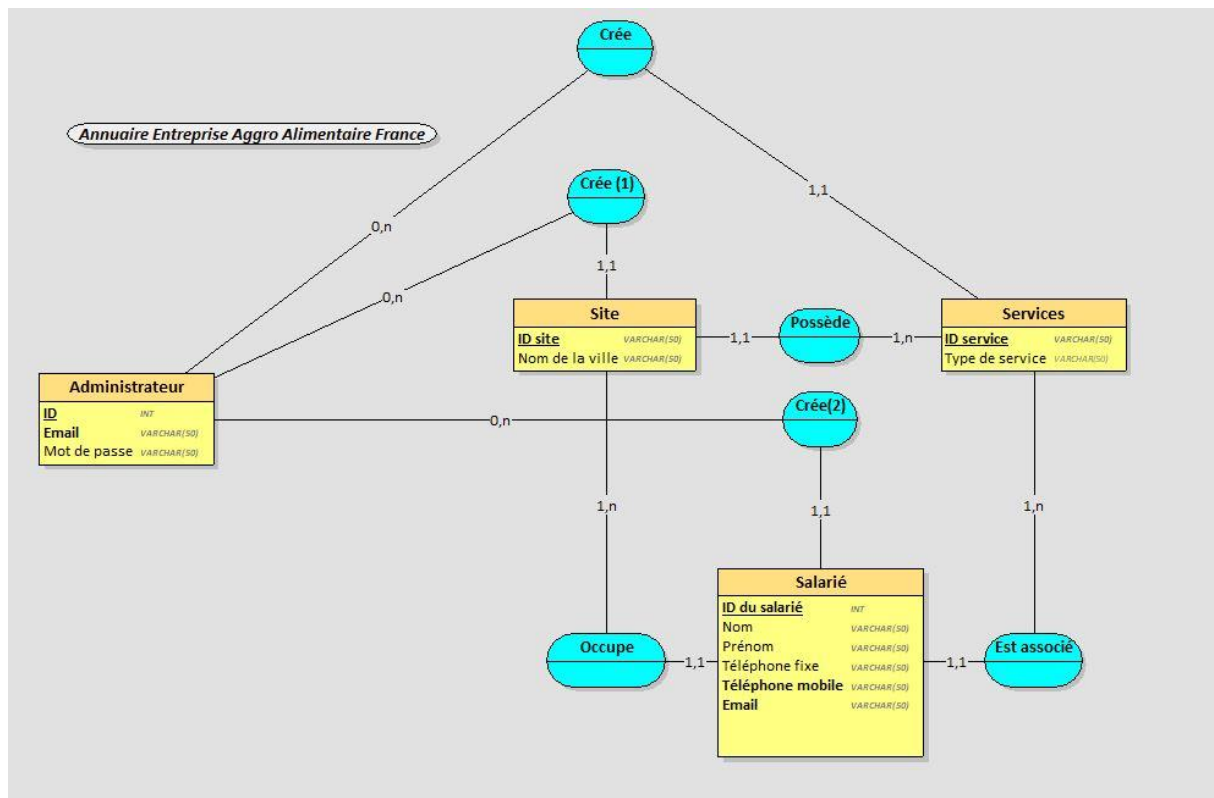
Le schéma ci-dessus aussi appelé « Diagramme de cas d'utilisations » indique de manière abstraite les différentes fonctionnalités qui seront présente lors du rendu final de l'application.

II \ Organisation des données et Modèles Pré-base.

Afin de stocker et d'afficher nos différentes données, nous avons commencé à réaliser différents schémas permettant d'identifier les tables qui seront nécessaire au bon fonctionnement de notre application web.

Un modèle conceptuel de donnée ou MLD est une représentation abstraite qui permet de facilement comprendre comment les différents éléments de notre application seront liés entre eux.

A la suite de ce document, vous trouverez un modèle conceptuel de données des tables potentiel de notre future base de données.



Conceptual Data Model of the website

Sur le schéma ci-dessus, nous avons disposés les différentes tables de données ainsi que leurs propriétés.

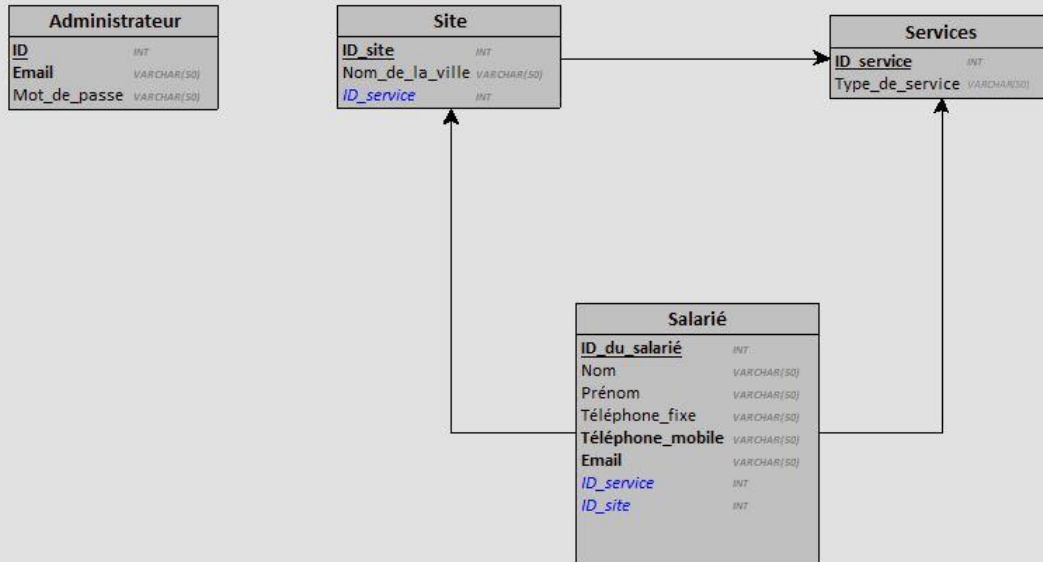
Ainsi nous pouvons réaliser que notre table contenant les différents sites de l'entreprise contiendra pour chaque site l'ID unique du site ainsi que le nom de la ville où se situe le site.

Chaque donnée présente dans la table de service contiendra un ID unique ainsi que le type de service.

Les administrateurs seront identifiés par un ID unique ainsi que par leur adresse mail et leur mot de passe.

Concernant les Salariés, de nombreuses informations seront renseignés car elles composeront leur fiche de contact (nom, prénom, mail, téléphone...).

De plus ce schéma permet aussi de comprendre que certaines tables seront liées entre elles, par exemple on sait que le site, le service et le salarié contiendront des informations communes car un salarié occupe un site et un site possède un service (exemple).



Logical Data Model of the website

Sur le schéma ci-dessus appelé modèle logique de données ou MLD nous nous sommes approchés encore un peu plus de la mise en forme d'une vraie base de données.

En effet, on constate que les liaisons initialement présentes entre site, salarié et service ont pris la forme de clés. De plus, ces clés nous serviront à identifier un salarié selon son service ou son site, ou encore à savoir le service d'un site.

De plus, nous avons supprimé la liaison présente entre la table salariée, site et service avec la table administrateur.

En effet, l'administrateur sera celui qui créera les données des différentes autres tables, mais ces données créées n'ont pas besoin d'être liées à leur créateur. Cette table est donc indépendante des autres.

III \ Architecture de l'application et création du GIT.

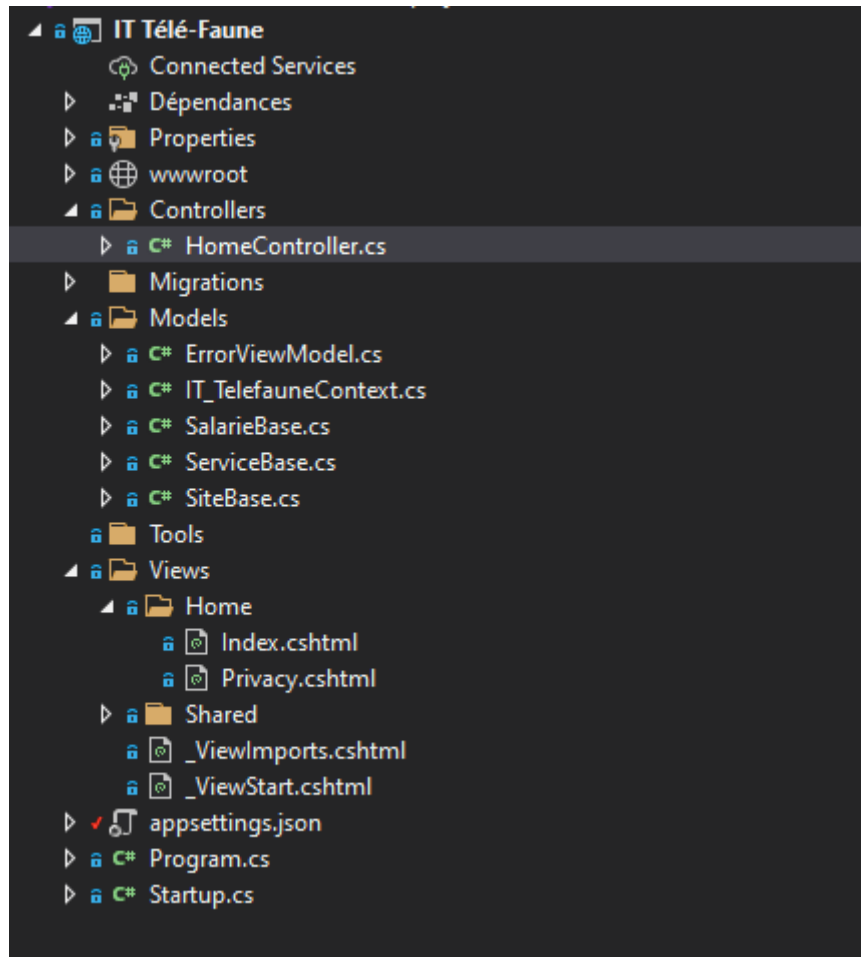
Dans cette partie nous ferons la présentation des différents outils de travail permettant la réalisation du projet.

1 \ Application de développement.

Tout d'abords nous réaliseront cette application sur le logiciel Visual Studio 2019 sous ASP Net core.

Afin d'organiser plus simplement notre code, nous avons choisis la simplicité d'une architecture MVC.

Le modèle MVC ou Models / Views / Controllers permet la séparation de l'application entre la logique métier et l'affichage pur.



MVC Model in the code of the Website

Sur cette capture d'écran, on peut directement vérifier la présence de trois différents fichiers principaux : Controllers / Models / Views.

2 \ Base de données

Concernant la base de données du logiciel, cette dernière sera hébergée sur MySQL Server Développer en local et on pourra la consulter via le logiciel Microsoft SQL Server Manager.

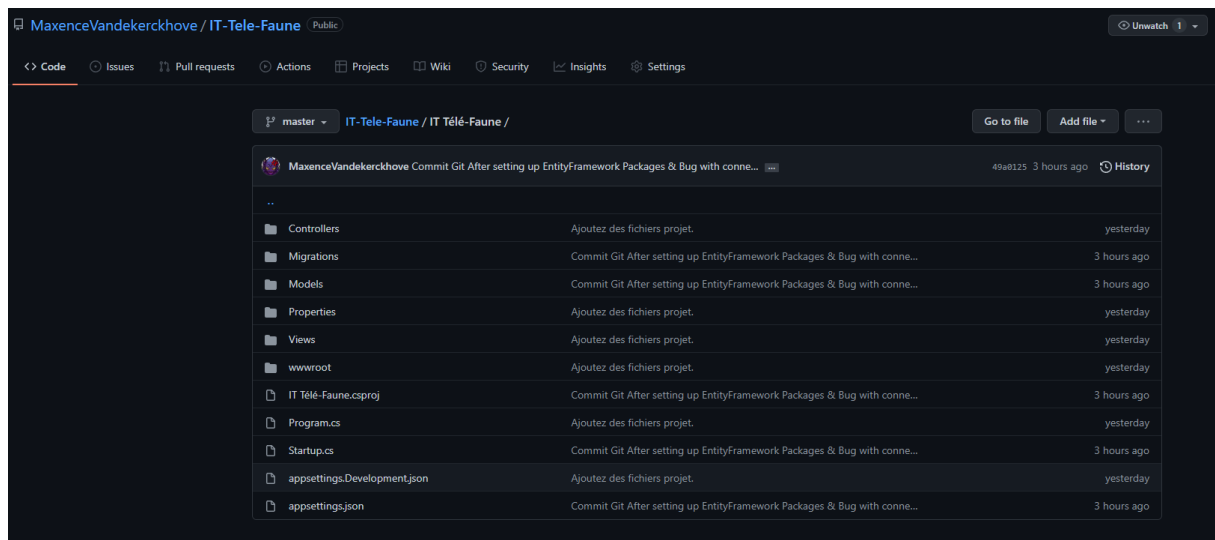
Les données y seront éditées via les différents CRUD présents sur le site.

3 \ GitHub et partage du projet.

Pour retrouver le projet sur internet, ce dernier est régulièrement mis à jour sur GitHub et est disponible sur le site via ce lien :

<https://github.com/MaxenceVandekerckhove/IT---Telefaune>

GitHub permet un partage plus simple des solutions informations ainsi qu'une possibilité de backup en cas de perte.



The folder and the code of the application available on GitHub

IV \ Technologies utilisées.

1 \ Razor.

Dans mon modèle-vue-contrôleur, j'ai décidé d'utiliser les vues Razor afin de découvrir une nouvelle technologie.

Razor permet de faciliter la conception des pages HTML présentes dans les Views en introduisant une nouvelle syntaxe, de gagner du temps et de gagner en lisibilité.

L'extension des fichiers Razor est : Fichier.cshtml.

Ci-dessous un code permettant la visualisation d'une page en fonction d'une session, on peut voir que ce code contient des parties en HTML pour la visualisation ainsi que des parties en C# :


```

<tbody>
    <foreach (var site in Model)>
    {
        <tr class="align-items-lg-center">
            <td class="text-center" width="20%"><a asp-controller="Site" asp-route-Id="@site.SiteId" asp-action="consultUsers">Consulter</a></td>
            <td class="text-center" width="20%">@site.NomSite</td>
            <td class="text-center" width="15%">@site.Ville</td>
            <td class="text-center" width="15%">@site.TypeServiceWrong</td>
            <if (ViewBag.Sessionnv == null)>
            {
                <td width="15%">
                    <div class="col-7">
                        <button type="button" class="btn btn-primary mx-1 disabled">Editer</button>
                    </div>
                </td>
                <td class="text-center" width="15%">
                    <div class="col-7 justify-content-md-center">
                        <button type="button" class="btn btn-danger mx-1 disabled">Supprimer</button>
                    </div>
                </td>
            }
            <if (ViewBag.Sessionnv != null)>
            {
                <td width="15%">
                    <div class="col-7">
                        <a asp-controller="Site" asp-route-Id="@site.SiteId" asp-action="Update" class="btn btn-primary mx-2">Editer</a>
                    </div>
                </td>
                <td class="text-center" width="15%">
                    <div class="col-7 justify-content-md-center">
                        <a asp-controller="Site" asp-route-Id="@site.SiteId" asp-action="DeleteGet" class="btn btn-danger mx-1">Supprimer</a>
                    </div>
                </td>
            }
        }
    }
</tbody>

```

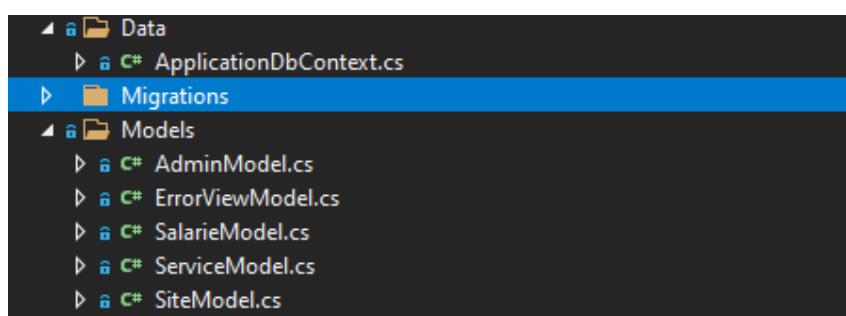
cshhtml file containing html and C# code parts using Razor

2\ Entity Framework et MySQL

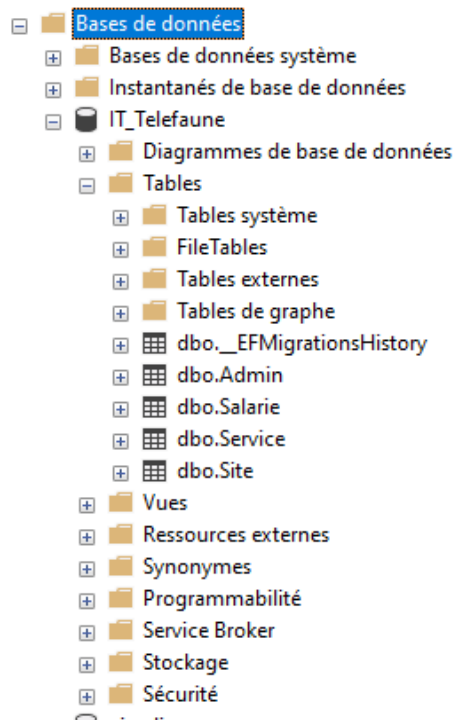
En plus d'utiliser Razor en tant que nouvelle technologie dans mon projet, j'ai décidé d'utiliser le package Entity Framework.

Microsoft Entity Framework est un outil qui permet la génération automatique d'une base de données en fonction de modèles directement intégrés dans le code.

En implémentant directement mes modèles de donnée dans mon code, je peux générer ou modifier mes tables de donnée à ma guise à l'aide de migrations.



The list of my models in code.



The Database generated by Entity Framework (in SQL Server Manager)

V\ Problèmes rencontrés.

Lors de mon projet, j'ai fait face à deux problèmes majeurs qui m'ont bloqué dans la conception de fonctions de bases :

Tout d'abord j'ai essayé d'intégrer une barre de recherche à la page principale avec pour objectif de pouvoir filtrer un tableau contenant à la fois les salariés, les sites et les services. Ce qui permettrait d'afficher n'importe quel élément en fonction de la recherche

Le problème était un problème de base de données dans le contrôleur Index.

Par rapport à ce problème j'aurais dû uniquement réaliser une barre de recherche sur la liste des salariés. Cela aurait été plus simple tout en remplissant la fonctionnalité demandée dans le cahier des charges de pouvoir effectuer une recherche sur les salariés.

Le second problème que j'ai rencontré est un problème d'attribution de donné dans la création de mes sites et de mes salariés.

En effet, lorsque je crée un salarié ou un site, j'ai une liste de valeur comme celle-ci :

Select Site ▼

Select Service ▼

Cependant, cette liste ne contient pas les valeurs des tables sites et services mais des données entrées manuellement afin de pouvoir tout de même faire fonctionner le reste de l'application.

Avec un peu plus de temps, j'aurais certainement trouvé un moyen pour afficher les données des tables sites et services dans les listes.

VI\ Améliorations possible.

Concernant les améliorations possibles, j'aurais aimé accorder plus de temps à la correction du problème d'affichage de donnée dans les listes, problème devenu bloquant pour l'application web.

De plus, il aurait été sage d'ajouter une barre de recherche dans la liste des utilisateurs mais aussi potentiellement dans les autres CRUD (Selon l'augmentation du nombre de données).

Pour finir, avec plus de temps, j'aurais pu améliorer le visuel de l'application en général afin de le rendre plus agréable lors de la navigation.

VII\ Conclusion

Même si j'ai rencontré différentes difficultés dans ce projet ainsi que des soucis de timing.

J'ai réussi à réaliser les différents points demandés dans le cahier des charges.

Ce projet m'a apporté beaucoup de connaissances quant à l'utilisation du Framework ASP.NET ainsi que de nouvelles connaissances sur des technologies tel que Razor et Entity Framework que je n'avais pas eu l'occasion d'utiliser auparavant.

M'ouvrir à de nouvelles choses a été aussi instructif que dangereux car cela m'a permis d'évoluer mais en contrepartie cela demandait une charge de travail décuplé car beaucoup de choses étaient à apprendre.

Dans le futur je serais heureux de réaliser un projet dans le même thème afin d'améliorer ou réaliser les points que j'aurais voulu traiter.