

Rapport de stage

Développement

LE NY Clara

3 janvier 2023 – 10 février 2023

Responsable du stage : Romain Jamet

Professeur référent : Nouredine Djabou

Etablissement : Lycée technologique Jean Chaptal, Quimper

Entreprise d'accueil : Guelt : ZAC Kervidanou 1, Quimperlé

Table des matières

Introduction.....	3
I-Présentation de l'entreprise	4
I-Présentation de Guelt	4
II-Présentation du service informatique	5
II-Nature de la mission, l'existant et sa critique	5
La mission	5
L'existant	5
Sa critique.....	6
III-Solution présenté.....	6
IV-Réalisation et mise en œuvre	7
La mise en place du projet	7
API	7
MVVM.....	8
Les modèles.....	8
Les vues	9
Les modèles de vues.....	9
La création de la base de donnée.....	10
Les tables.....	10
Les procédures stockées.....	11
Interface utilisateur.....	12
Page de login	12
Page de pointage.....	13
Page d'administration des utilisateurs.....	14
Ce que fait mon programme	14
V-Conclusion.....	15

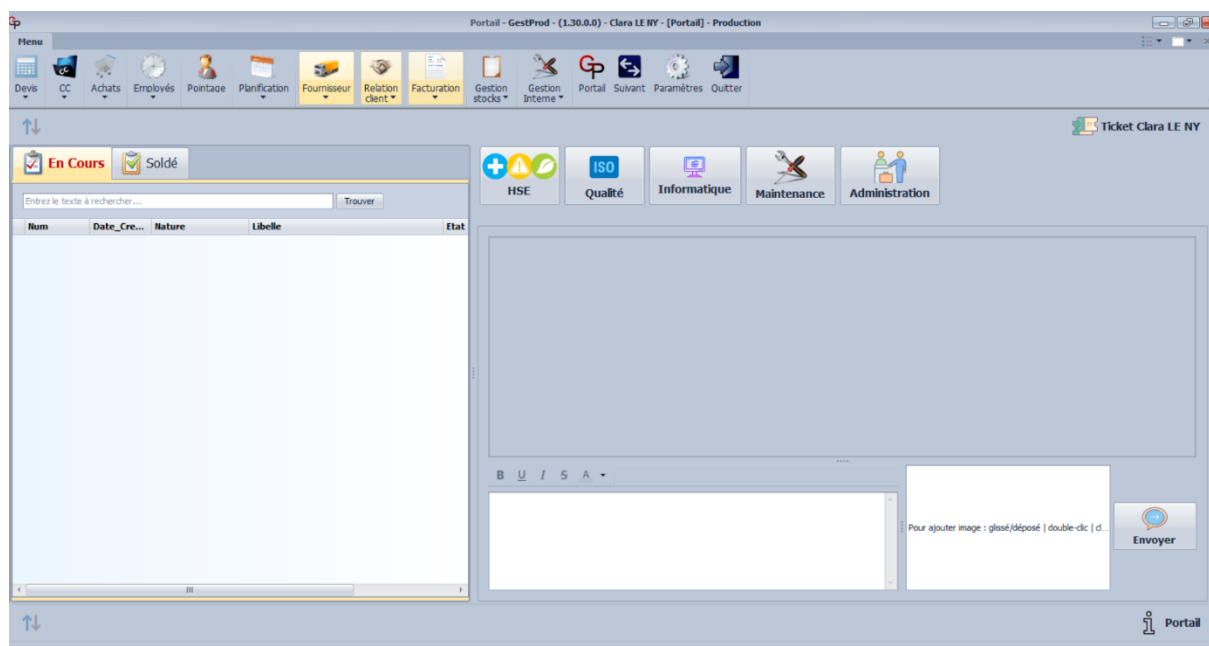
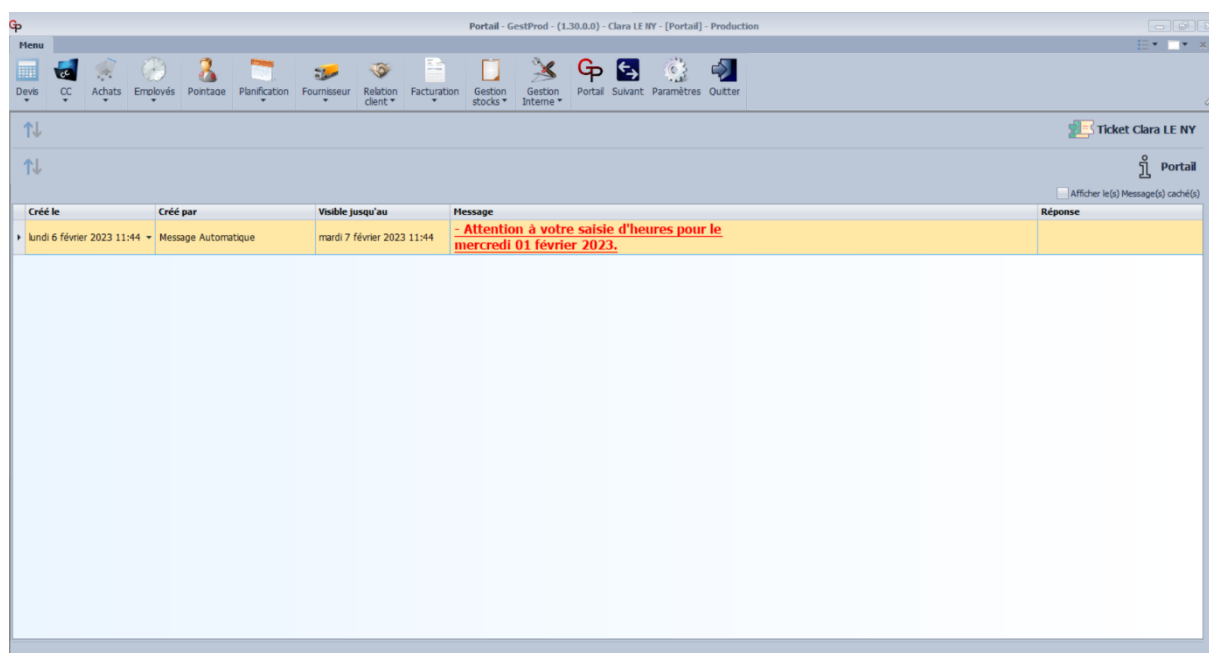
Introduction

En BTS SIO, j'ai effectué mon stage au sein de la société Guelt, située à Quimperlé du 3 janvier au 10 février 2023.

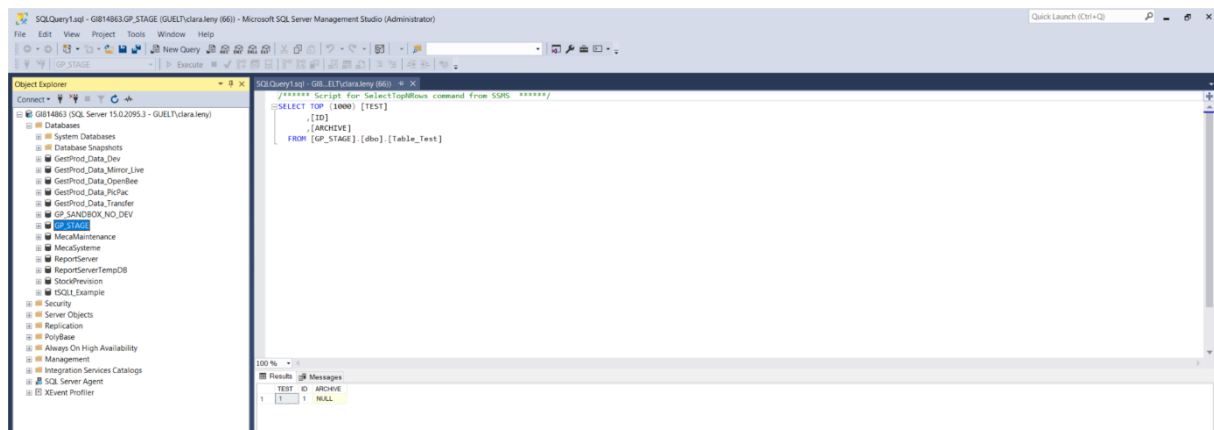
J'ai choisi cette entreprise car j'avais la possibilité d'apprendre de nouvelles choses et d'approfondir mes compétences aussi bien en développement qu'en administration réseau.

Pendant ce stage j'ai utilisée différents outils qu'utilisais l'entreprise tels que

- GestProd : ERP de l'entreprise développer en interne permettant de tout gérer dans l'entreprise et aussi les tickets d'incidents



- SQL Server Management : Où j'avais accès à toutes les bases de données de l'entreprise



- Visual Studio 2022 : Pour la réalisation du projet
- DevExpress : Licence qui permet d'accéder à différents composants de programmation.

I-Présentation de l'entreprise

I-Présentation de Guelt

L'entreprise a été créée en 1978 par à Yves Guelt. L'entreprise est spécialisée dans la conception d'équipements pour l'industrie agroalimentaire et propose des services d'usinage, de maintenance, de réparation de pièce et de machines pour l'industrie.

GUELT

Concepteur / Constructeur / Intégrateur

FILLPACK

Le dosage, notre métier

Foodbizz

Distribution de matériel industriel

**MECA
SYSTEME**

La mécanisation de l'emballage

**PicPac
Systems**

Solutions robotiques

L'INTELLIGENCE TECHNIQUE PARTAGÉE

Ses missions consistent à la création de machine dans différents secteurs d'activités tels que l'agroalimentaire pour répondre au besoin en équipement de process et d'emballage, les GMS (grande et moyenne surface) pour l'emballage des produits dans les laboratoires boucherie, traiteur et poissonnerie. La création de machines de conditionnement automatisée qui répondent aux normes sanitaires des cuisines collectives.

Plusieurs services sont disponibles chez Guelt comme Guelt Techni-service qui proposent un suivi technique et l'usinage et fabrication de pièces à façon, le reconditionnement de pièce et de machines et l'intervention technique sur site. Ainsi que Guelt nautic qui s'occupe de la fabrication et la rénovation de pièces pour le nautisme tel que les pièces d'accastillage, quilles, bulbes, pièces mécaniques de contrôle de foils ou encore des pieds de mâts.

L'entreprise aujourd'hui est une SAS (Société par action simplifiée), son siège principal se situe à Quimperlé et possède 5 sites et plus de 400 employés au service des projets des clients.

II-Présentation du service informatique

Le service informatique est composé du 11 personnes. Les missions des développeurs consistent en la maintenance de l'ERP GestProd, de la base de donnée, à la résolution de ticket coter dev et de participer aux nouveaux projets de l'entreprise. Les missions des réseaux sont la maintenance du système informatique et la résolution de ticket coté réseaux.

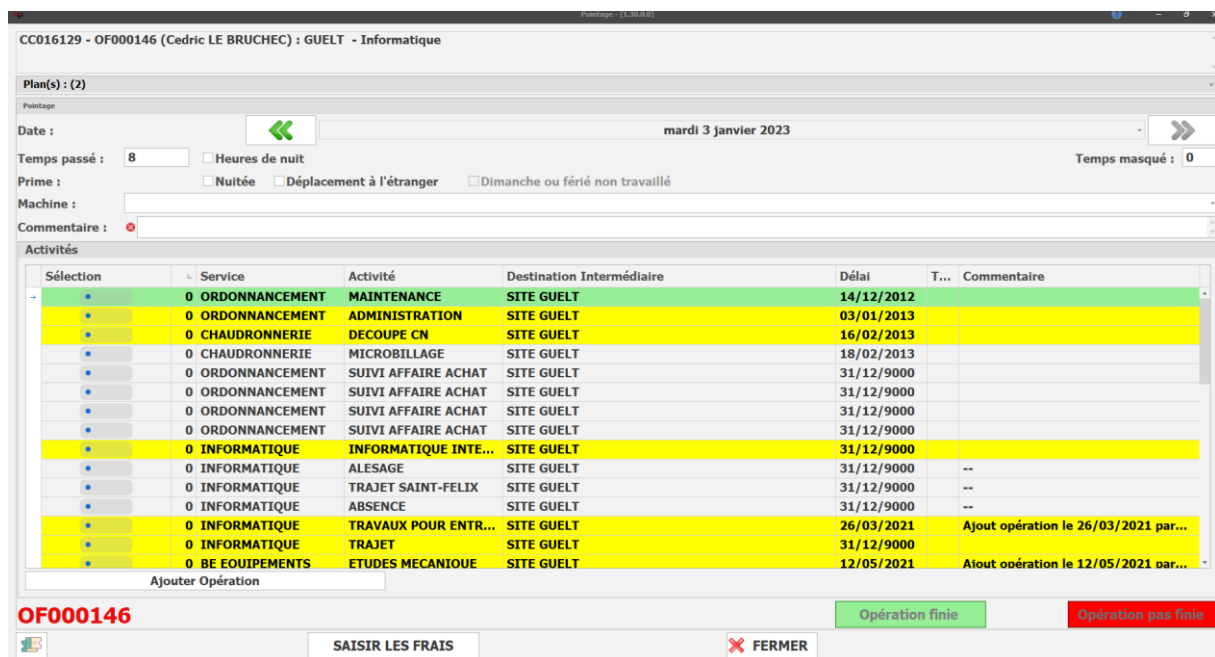
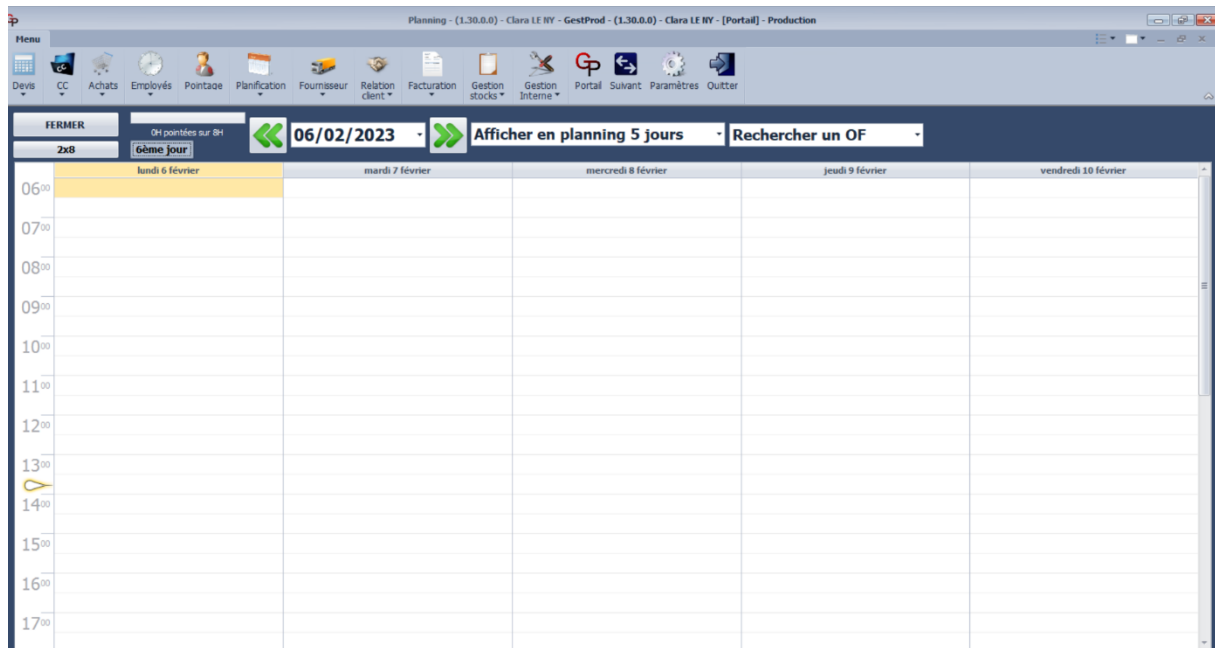
II-Nature de la mission, l'existant et sa critique

La mission

Ma mission tout au long de ces 6 semaines de stage a été de programmer une application en CSharp pour faire une étude de faisabilité et de future mise en place d'un Web Service permettant d'externaliser un système de pointage ressemblant à celui présent dans l'application GestProd. Ce qui me permet de participer à l'évolution d'un site Web exploitant les données de l'organisation.

L'existant

Le système de pointage existe déjà dans le portail GestProd, il permet à chaque salarié de pointer ces heures pour chaque opération qu'il a fait au court de la journée grâce à un ordre de fabrication (OF) précis.



Sa critique

Bien que le système soit déjà existant dans GestProd, il est absolument nécessaire de lancer GestProd pour accéder à ce service et son lancement peut être plus au moins long en fonction de la connexion de chacun notamment lorsque les personnes sont en déplacement où accède à leur bureau à distance. Pour que ce service soit optimiser il faudrait que ces personnes puissent y accéder en instantané où qu'elles soient sans charger les autres informations de GestProd dont elles n'ont pas besoin.

III-Solution présenté

La solution présenté est un Web Service http en WPF codé en CSharp qui sera MVVM (Model-View-ViewModel) et regroupera les informations nécessaires au pointage.

IV-Réalisation et mise en œuvre

Pour mettre à bien cette mission je me suis renseigné sur les différentes méthodes pour créer un Web Service en CSharp avec, WPF ou minimal api. Comment tester le Web Service avec Postman ou Swagger. J'ai pu m'appuyer sur un tuto YouTube nommé « TimCo Retail Manager Course » pour commencer à créer de 0 ce Web Service.

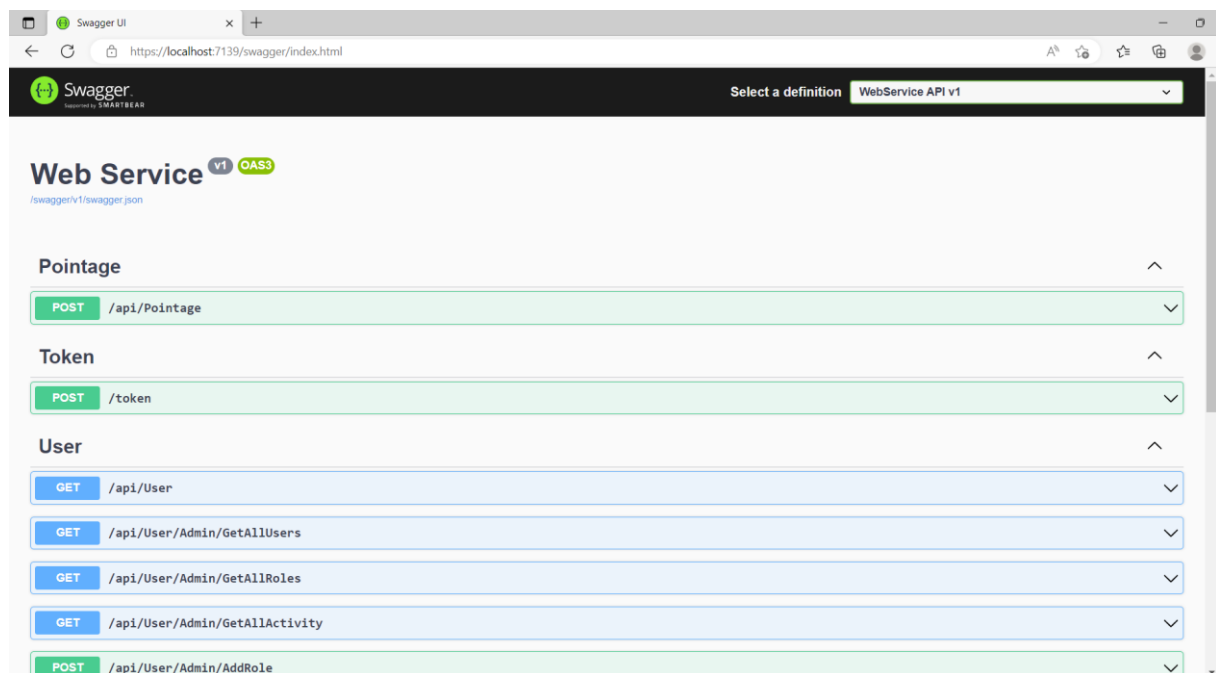
La mise en place du projet

Pour commencer j'ai créé le projet, configuré la route, créé le premier utilisateur puis à installer Swagger et toute sa configuration

API

Swagger est un outil permettant de concevoir et de modéliser une API dans n'importe quel langage de programmation, il offre aussi la possibilité de faire des tests simples directement dans l'API et possède une documentation interactive.

Voici mon API avec Swagger, qui reprend différentes fonctions dans mon programme pour pouvoir les tester et afficher les informations.



POST /token

Parameters

Try it out

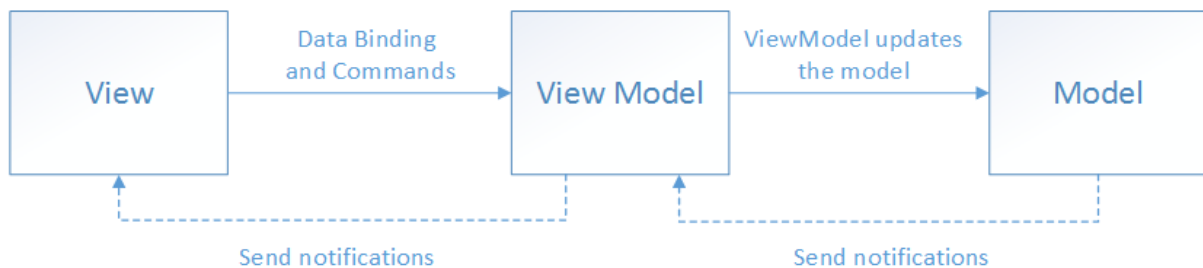
Name	Description
username string (query)	<input type="text" value="username"/>
password string (query)	<input type="text" value="password"/>
grant_type string (query)	<input type="text" value="grant_type"/>

Responses

Code	Description	Links
200	Success	No links

MVVM

Le modèle MVVM comporte 3 composants principaux : le modèle, la vue et le modèle de vue. Chacun sert un objectif distinct. Le diagramme ci-dessous montre les relations entre les 3 composants. Avec ce modèle MVVM, j'ai pu participer à la conception de l'architecture d'une solution applicative.



Les modèles

Les classes de modèle sont des classes non visuelles qui encapsulent les données de l'application. Chaque modèle de l'application est créé pour des données précises.

Exemple d'un modèle, EmployeeModel :


```

8 références
public class EmployeModel
{
    2 références
    public string Empl_ID { get; set; }
    1 référence
    public string _CreateByUser { get; set; }
    2 références
    public Dictionary<string, string> Roles { get; set; } = new Dictionary<string, string>();

    0 références
    public string RoleList
    {
        get
        {
            return string.Join(", ", Roles.Select(x => x.Value));
        }
    }
}

```

Exemple d'une utilisation dans un modèle du vue :

```

var userList = await _employeEndpoint.GetAll();
Users = new BindingList<EmployeModel>(userList);

```

Les vues

Les vues sont responsables de la définition de la structure, de la disposition et de l'apparence de ce que l'utilisateur voit à l'écran. Chacune des vues sont définies en XAML.

Exemple d'une vue définie en XAML, LoginView :

```

<RowDefinition Height="auto" />
<RowDefinition Height="*" />
</Grid.RowDefinitions>
<TextBlock Grid.Row="0" Grid.Column="1" Grid.ColumnSpan="2" Margin="0 0 0 20"
    HorizontalAlignment="Center" FontSize="64" FontFamily="Segoe UI Light">
    Connexion
</TextBlock>

<!-- Error Message row -->
<TextBlock x:Name="ErrorMessage" Visibility="{Binding IsErrorVisible, Converter={StaticResource BooleanToVisibilityConverter},
    FallbackValue=Collapsed}"
    Grid.Row="1" Grid.Column="1" Grid.ColumnSpan="2"
    Foreground="Red" MaxWidth="300"
    TextWrapping="Wrap" Margin="0 0 0 20"/>

<!-- Username row -->
<TextBlock Grid.Row="2" Grid.Column="1" Margin="0 0 5 10">
    Nom d'utilisateur :
</TextBlock>
<TextBox x:Name="UserName" Grid.Row="2" Grid.Column="2" MinWidth="350"
    Margin="0 0 0 10" />

<!-- Password row -->
<TextBlock Grid.Row="3" Grid.Column="1" Margin="0 0 5 20">
    Mot de passe :
</TextBlock>
<PasswordBox x:Name="Password" Grid.Row="3" Grid.Column="2" MinWidth="350"
    Margin="0 0 0 20" />

<!-- Log In row -->
<Button Grid.Row="4" Grid.Column="1" Grid.ColumnSpan="2" Padding="20 5"
    HorizontalAlignment="Center" x:Name="Login">
    S'identifier
</Button>

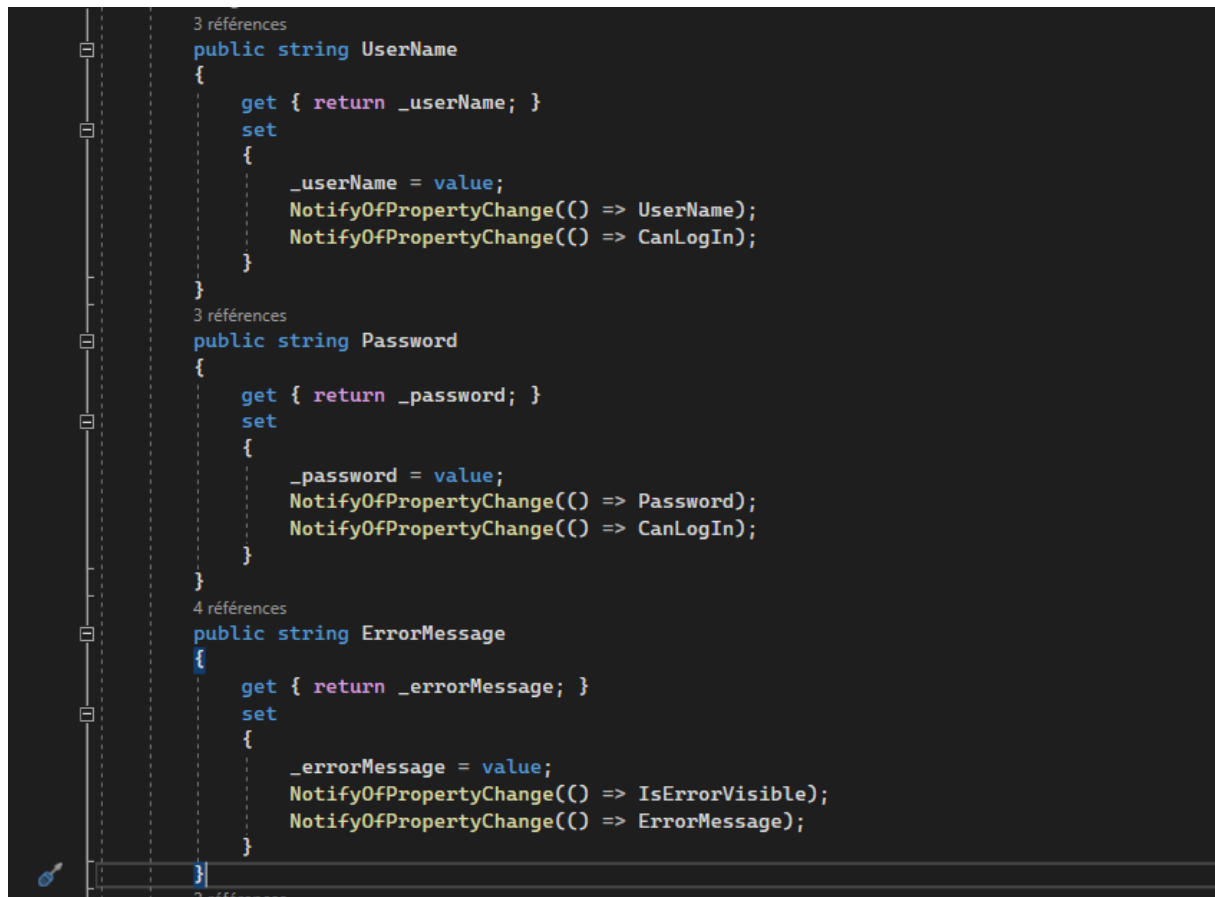
```

Les modèles de vues

Les modèles de vues implémentent des propriétés et des commandes auxquelles les vues peuvent lier les données et informent les vues de tout changement d'état par le biais d'événements de notification de modification.

Les modèles de vues sont également responsables de la coordination des interactions des vues avec les classes de modèles requises. Chaque modèle de vue fournit des données à partir d'un modèle sous une forme que la vue peut facilement consommer. Pour ce faire, des conversions de données sont parfois nécessaires dans les modèles de vues.

Exemple d'un modèle de vue, LoginViewModel :



```
3 références
public string UserName
{
    get { return _userName; }
    set
    {
        _userName = value;
        NotifyOfPropertyChange(() => UserName);
        NotifyOfPropertyChange(() => CanLogin);
    }
}

3 références
public string Password
{
    get { return _password; }
    set
    {
        _password = value;
        NotifyOfPropertyChange(() => Password);
        NotifyOfPropertyChange(() => CanLogin);
    }
}

4 références
public string ErrorMessage
{
    get { return _errorMessage; }
    set
    {
        _errorMessage = value;
        NotifyOfPropertyChange(() => IsErrorVisible);
        NotifyOfPropertyChange(() => ErrorMessage);
    }
}

2 références
```

La création de la base de donnée

Ce point permet de remplir les critères concernant la conception ou l'adaptation d'une base de donnée, de l'exploitation des données à l'aide d'un langage de requêtes, son administration et son déploiement ainsi que l'utilisation des composants d'accès aux données.

Les tables

Pour stocker les informations sur les utilisateurs et celles du pointage, j'ai eu besoin de créer une base de données locale qui contient 3 tables.

Employe

Elle contient les Id haché, le prénom, le nom et le login de chaque utilisateur utilisant le service

Fichier de script : Employe.sql

Nom	Type de données	Autoriser les valeurs NULL	Par défaut
Empl_ID	nvarchar(128)	<input type="checkbox"/>	
Empl_Nom	nvarchar(128)	<input type="checkbox"/>	
Empl_Prenom	nvarchar(128)	<input type="checkbox"/>	
_CreateByUser	nvarchar(128)	<input type="checkbox"/>	

Clés (1)
 <sans nom> (Clé primaire, Clustered: Empl_ID)
Contraintes de validation (0)
Index (0)
Clés étrangères (0)
Déclencheurs (0)

PointageStatutCommentaires

Elle contient l'Id et le Libelle des opérations possible aux pointages

Fichier de script : PointageStatutCommentaires.sql

Nom	Type de données	Autoriser les valeurs NULL	Par défaut
PtAg_ID	int	<input type="checkbox"/>	
PtSc_Libelle	nvarchar(128)	<input type="checkbox"/>	

Clés (1)
 <sans nom> (Clé primaire, Clustered: PtAg_ID)
Contraintes de validation (0)
Index (0)
Clés étrangères (0)
Déclencheurs (0)

Pointage

Elle contiendra pour chaque pointage, l'Id du pointage, l'Id de l'employé, le Libelle de l'opération, le nombre d'heure passer dessus, le possible commentaire du salarié ainsi que la date du jour préremplie et le nom d'utilisateur de l'employé.

Fichier de script : Pointage.sql

Nom	Type de données	Autoriser les valeurs NULL	Par défaut
PtAg_ID	int	<input type="checkbox"/>	
Empl_ID	nvarchar(128)	<input type="checkbox"/>	
PtSc_Libelle	nvarchar(128)	<input type="checkbox"/>	
PtAg_NombreHeures	decimal(18,2)	<input type="checkbox"/>	
PtSc_Commentaires	nvarchar(128)	<input checked="" type="checkbox"/>	
PtAg_Date	datetime2(7)	<input type="checkbox"/>	
_CreateByUser	nvarchar(128)	<input type="checkbox"/>	

Clés (1)
 PK_Pointage (Clé primaire, Clustered: PtAg_ID)
Contraintes de validation (0)
Index (0)
Clés étrangères (1)
 FK_Pointage_ToEmployee (Empl_ID)
Déclencheurs (0)

Les procédures stockées

Pour faciliter les changements dans la base de donnée, j'ai créés 2 procédures stockées qui seront appelé dans mon programme.

spEmploye_Lookup

Elle permet de récupérer les données de l'employé grâce à son Id

```
CREATE PROCEDURE [dbo].[spEmploye_Lookup]
    @Id nvarchar(128)
AS
begin
    set nocount on;

    SELECT Empl_ID, Empl_Nom, Empl_Prenom, _CreateByUser
    from [dbo].[Employe]
    Where Empl_ID = @Id;
end
```

spPointage_Insert

Elle va permettre d'insérer les données dans la table pointage après la validation du programme

```

CREATE PROCEDURE [dbo].[spPointage_Insert]
    @PtAg_ID int output,
    @Empl_ID nvarchar(128),
    @PtSc_Libelle nvarchar(128),
    @PtAg_NombreHeures decimal(18,2),
    @PtSc_Commentaires nvarchar(128),
    @PtAg_Date datetime2,
    @_CreateByUser nvarchar(128)
AS
begin
    set nocount on;

    insert into dbo.Pointage(Empl_ID, PtSc_Libelle, PtAg_NombreHeures, PtSc_Commentaires, PtAg_Date, _CreateByUser)
    values(@Empl_ID, @PtSc_Libelle, @PtAg_NombreHeures, @PtSc_Commentaires, @PtAg_Date, @_CreateByUser);

    select @PtAg_ID = Scope_Identity();
end

```

Interface utilisateur

Pour que les utilisateurs puissent interagir avec le Web Service de pointage, il faut créer des vues afin qu'ils puissent renseigner les informations nécessaires. Ces vues vont permettre d'envoyer les informations saisies aux modèles de vues afin de les traiter. Ce point permet de modéliser une solution applicative, de gérer les accès et les privilèges appropriés avec différents rôles qui autorise ou non l'accès aux différentes vues ainsi que d'identifier, développer, utiliser ou adapter des composants logiciels. En effet, pour la page de pointage j'ai utilisé un composant de DevExpress.

Page de login

ShellView

Fichier

Connexion

Nom d'utilisateur :

test@imatest.com

Mot de passe :

●●●●●●●●

S'identifier

Page de pointage

ShellView

— □ ×

Fichier Compte Utilisateurs

Page de Pointage

Date :

Activités :

Commentaire :

Temps passé :

Pointer

ShellView

— □ ×

Fichier Compte Utilisateurs

Page de Pointage

Date :

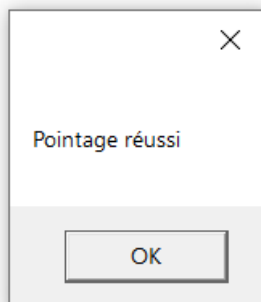
Activités :

Commentaire :

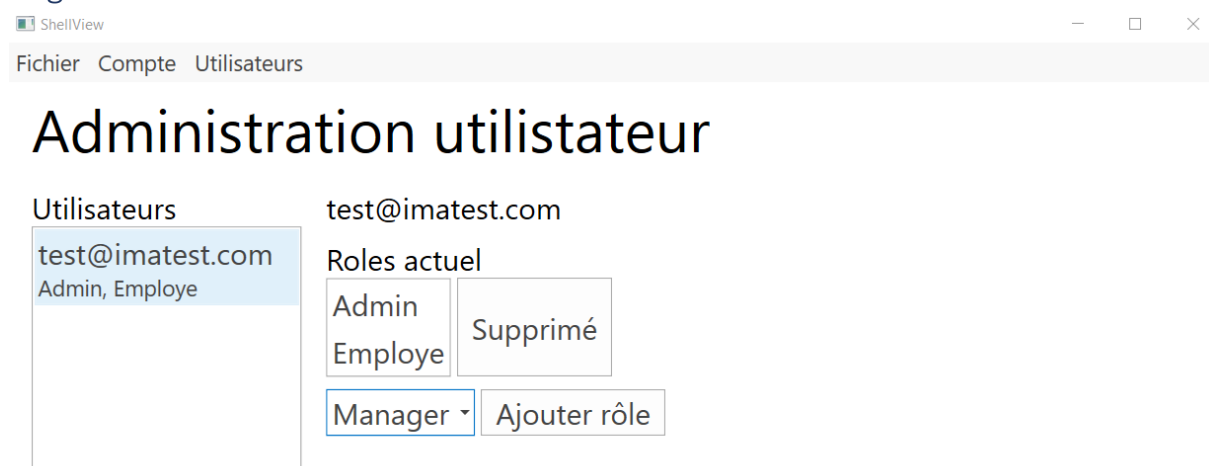
Temps passé :

Pointer

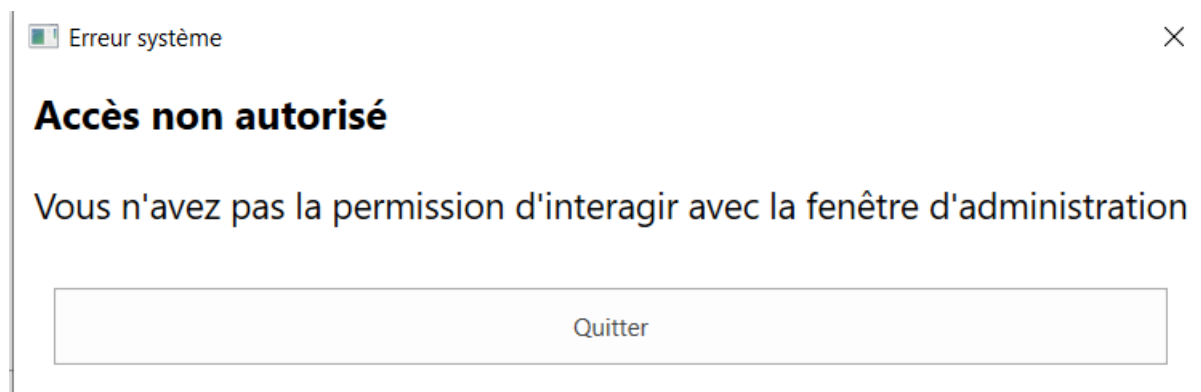
Si toutes les informations obligatoire pour le pointage sont remplies et que l'utilisateur clique sur « Pointer » :



Page d'administration des utilisateurs



Si l'utilisateur n'a pas le rôle « Admin » il ne pourra pas accéder à cette page



Ce que fait mon programme

Mon programme permet de cocher la mise en œuvre et la vérification de la conformité de la solution applicative et de son développement à un référentiel, une norme ou un standard de sécurité.

Pour la page de login, je récupère le login et le mot de passe entré dans les emplacements prévu, et grâce à plusieurs méthode j'identifie si l'utilisateur est apte à se connecter ou non et s'il est autorisé à interagir avec la fenêtre de pointage qui se lance juste après. J'ai pris en compte la sécurité dans le projet de développement de la solution applicative en hachant les mots de passe dans la base de données locale.

La page de pointage qu'en à elle va récupérer la date préremplie, l'activité choisie grâce à un ComboBox, le commentaire ainsi que le temps passer avec le composant de DevExpress, SpinEdit. Le bouton pour le pointage n'est accessible qu'après avoir choisis une activité et avoir ajouter un nombre positif dans le SpinEdit. Si le pointage est possible, les informations remplies seront ajouté dans la base de données

Pour finir la page d'administration utilisateurs va lister tous les utilisateurs du système et les Admins pourrons modifier les rôles de chacun d'entre eux, les changements s'opèrent directement.

Pour pouvoir relier toutes ces informations, j'ai exploité les ressources du cadre applicatif avec des frameworks.

V-Conclusion

Pour conclure, grâce à ce stage, j'ai pu avoir un aperçus de ce que pouvait être un Web Service ainsi que les différentes méthodes pour parvenir à en faire un. J'ai été confronté à plusieurs dysfonctionnements dans mon programme que j'ai su analyser et corriger.

J'ai pu commencer à développer un projet qui sera repris par les développeurs après mon stage et j'ai réussi à atteindre différents objectifs fixés dans la réalisation de celui-ci.