

Restitution du stage de deuxième année  
Effectué du 02 novembre au 18 décembre

Dans la Société : Amiciel



A Saint Quentin Fallavier

Pauline Arnaud

Etudiante en

Brevet de Technicien Supérieur  
Services Informatiques Aux Organisations  
2019-2021

Au Lycée Ella Fitzgerald à Saint Romain en Gal



## Plan

### Sommaire

Introduction .....	2
I. L'entreprise.....	3
A. Contexte de l'entreprise .....	3
B. Les outils et technologies de l'entreprise .....	3
II. La mission en entreprise .....	4
A. Objectif de la mission.....	4
B. La réalisation de la mission .....	5
III. Bilan du stage.....	9
A. Bilan de la mission .....	9
B. Bilan personnel .....	9
Conclusion.....	10

## Introduction

Du 2 novembre au 18 décembre 2020, j'ai effectué un stage au sein de l'entreprise Amiciel (située à Saint Quentin Fallavier). Au cours de ce stage en développement, j'ai pu m'intéresser à une nouvelle technologie, le Blazor ou apprendre de nouvelles techniques comme le MVC.

Ce stage m'a permis d'appréhender le travail de développeur en entreprise, entre la charge de travail, l'organisation, la communication entre toutes les personnes de l'entreprise. Il a consisté à créer un module de gestion de suivis de contrats en Blazor qui doit permettre d'ajouter, supprimer, modifier un contrat mais aussi d'éditer tous les contrats voulu en une seule fois.

Je vais dans un premier temps présenter l'entreprise puis dans un deuxième temps la mission de mon stage et enfin je ferai le bilan du stage.

# I. L'entreprise

## A. Contexte de l'entreprise

J'ai fait mon stage dans l'entreprise Amiciel qui se trouve à Saint Quentin Fallavier en Isère.

L'entreprise Amiciel est une SARL (société à responsabilité limitée), elle propose des prestations comme :

- L'installation d'équipement informatique
- Conception et vente d'application métier (Malice, Sacha)
- L'infogérance
- L'hébergement

Amiciel se trouve dans un parc d'activité, possède un effectif de 5 personnes pour environ 400 clients qui se trouve la plupart du temps être des structures qui accueillent des enfants (crèches, accueils périscolaires, mais aussi entrepôt logistique).

L'équipe se compose de :

- 1 responsable de suivi de clientèle
- 1 technicien réseau
- 3 développeurs, Issam a fait un BTS SIO il a aussi suivi une formation d'ingénieur qu'il a arrêté pour travailler chez Amiciel, Florian a fait une reconversion professionnelle en suivant une formation dans le web.

Amiciel est une entreprise qui sait se moderniser et utiliser les nouvelles technologies qui se développent.

## B. Les outils et technologies de l'entreprise

L'entreprise Amiciel met en place des progiciels mais pas seulement, elle gère aussi l'installation de matériel comme des badgeuses, des distributeurs de plateaux, la configuration du matériel, des tablettes, la configuration des progiciels.

Amiciel s'occupe de plusieurs progiciels de gestion :

- Sacha, un progiciel que les gérant ont racheté il y a plusieurs années mais ils ont de moins en moins de clients qui l'utilise
- Malice, un progiciel utilisé en : petite-enfance, enfance, cantine, jeunesse, activités annuelles, planification avancée, gestion d'accès, aires d'accueil des

gens du voyages. L'application est sous forme de modules qui peuvent être ajouté en fonction des besoins de client.

L'équipe d'Amiciel développe depuis plusieurs mois en Blazor WebAssembly qui est une technologie assez récente. Elle permet d'utiliser du .NET afin de réaliser l'intégralité d'un site web. Le code se trouve être du C# qui compilé dans une DLL pour être exécuté dans le client sur la machine. Donc l'application Malice est développé en Blazor et tous les nouveaux modules qu'ils ajoutent sont aussi en Blazor.

L'un des clients d'Amiciel a demandé à l'entreprise de développer un module de gestion de contrats de maintenance. C'est un module de gestion de la relation client (Customer Relationship Management en anglais). Il devra permettre d'éditer rapidement tous les contrats voulus. Un contrat de maintenance est globalement identique pour tous les clients, il régit le contenu des prestations proposés et les obligations de chaque partie. Les éléments qui diffèrent d'un contrat à l'autre sont le nom du client, les informations liées aux clients telles que l'adresse, l'email, le numéro de téléphone mais aussi l'objet du contrat, le tarif qui en découle ou encore les dates du contrat.

L'entreprise a donc décidé de me mettre sur ce projet pour toute la durée de mon stage.

## II. La mission en entreprise

### A. Objectif de la mission

L'objectif de ma mission est de développer le module de gestion de contrats qui sera intégrer à leur progiciel Malice. Dans un premier temps Amiciel utilisera pour eux ce module afin de l'améliorer au mieux. Par la suite, leurs clients pourront utiliser ce module si cela leur est nécessaire.

Quand le client sera sur l'application Malice, qu'il aura créé son nouvel abonnement en insérant des informations comme la date de début et de fin. Il pourra cliquer sur un bouton modifier qui l'amènera sur une page internet où se trouve un formulaire afin de compléter les informations liées au contrat de maintenance (l'objet du contrat, le mode de renouvellement du contrat, les tarifs, les structures concernées, le signataire...).

Ensuite le client pourra visualiser tous les contrats et en sélectionner certains selon des filtres mit en place. Ces contrats pourront alors être éditer pour être imprimer ou envoyer.

La mission s'est faites étape par étape avec à chaque fois un élément à développer.

## B. La réalisation de la mission

Avant de débiter sur le projet qui utilise la technologie Blazor WebAssembly que je ne connais pas, l'un des développeur, Issam m'a fait débiter sur un petit projet qui n'était pas en lien avec le projet. J'ai fait des petites missions telles que la création d'un petit formulaire afin de voir comment les données étaient utilisées, déplacées, enregistrées ; ou changer la couleur d'un texte en fonction du bouton sur lequel on click. Ce petit projet m'a permis de comprendre ce qu'était le MVC (Model View Controller) et comment fonctionnait Blazor.

J'ai par la suite commencé à travailler sur le vrai projet. Dans un premier temps mon maître de stage, Bruno m'a expliqué dans les grandes lignes comment fonctionnait le projet, ce qu'il faisait, les sous projet sur lesquelles je travaillerai. J'ai ensuite commencé à créer la page internet qui contiendra un formulaire où l'utilisateur remplira les informations nécessaires au contrat de maintenance. Au début j'ai recherché des composants sur Bootstrap et des idées de positionnement des éléments sur la page.

Ci-dessous se trouve la partie vue du formulaire ainsi que le résultat.

```
28 <editform Model="Contract" @ValidSubmit="Submit">
29 <div class="form-group" id="form" style="width: 100%;>
30 <div class="form-group row">
31 <div class="col-lg-3">
32 <DropdownFor TValue="Malicelite.MaintenanceContract.Contract_Object" Text="Objet du contrat" Bind-Value="@Contract.ContractObject" />
33 </div>
34 <div class="col-lg-3">
35 <DropdownFor TValue="Malicelite.MaintenanceContract.Renewal_Mode" Text="Mode de renouvellement" Bind-Value="@Contract.RenewalMode" />
36 </div>
37 <div class="col-lg-3" id="input-group" style="top: 0.55rem;">
38 <label for="tariff" class="col-form-label search-label-up">Tariff/Label</label>
39 <InputNumber class="form-control" Bind-Value="@Contract.PriceOff" /></InputNumber>
40 </div>
41 <div class="col-lg-3" id="input-group" style="top: 0.55rem;">
42 <label for="tariff" class="col-form-label search-label-up">Indice Syntec/Label</label>
43 <InputNumber class="form-control" Bind-Value="@Contract.IndicesSyntec" /></InputNumber>
44 </div>
45 </div>
46
47 <div class="form-group row">
48 <div class="col-lg-6"><creator Label="Liste des structures" Bind-Value="@Contract.ListStructures" Rows="4" /></div>
49 <div class="col-lg-6"><creator Label="Detail du tarif" Bind-Value="@Contract.PriceDetail" Rows="4" /></div>
50 </div>
51
52 <div class="form-group row">
53 <div class="col-lg-4">
54 <DropdownFor TValue="Malicelite.MaintenanceContract.Status_Contract" Text="Etat du contrat" Bind-Value="@Contract.StatusContract" />
55 </div>
56 <div class="col-lg-4" style="top: 0.67rem;"><textfor Label="Emplacement du serveur" Bind-Value="@Contract.ServerPlace" /></div>
57 <div class="col-lg-4" style="top: 0.67rem;"><textfor Label="Signature" Bind-Value="@Contract.Signatory" /></div>
58 </div>
59
60 <div class="form-group float-right">
61 <button type="button" class="Link" href="/managerclient/subscriptions/allsub13details" @Color="WebColor.Secondary">Visualiser les contracts</button>
62 <button type="button" class="Submit" @Color="WebColor.Primary">Enregistrer</button>
63 </div>
64 </div>
65 </editform>
```

Afin d’avoir un meilleur aperçu du rendu et des fonctionnalités qui seront sur la page, Issam m’a conseillée de faire un schéma et de poser des questions à Bruno pour mieux comprendre la finalité du projet. Par exemple comment organiser les données, si on fait un bouton valider, supprimer ou autre. J’ai dû enregistrer les données renseignées pour les enregistrer dans la base de données, pour cela j’utilisais le Linq. Je devais faire appel au serveur afin d’accéder à la base de données.

Voici un exemple d’appel du serveur.

```

35 references | 0 changes | 0 authors, 0 changes
protected override async Task OnInitializedAsync()
{
    id = Convert.ToInt32(HttpUtility.ParseQueryString(Nav.Uri.Substring(Nav.Uri.IndexOf("?")))[ "id"]);
    ApiResponse<MaintenanceContract> response = await Api.Get<MaintenanceContract>(Area.manager, $"subscription130get?id={id}", "subscription130");
    if (Api.ShowPopupError(response)) return;
    Contract = response.ObjectResult;
}

```

Pour créer le formulaire de la manière la plus efficace possible, l’entreprise a créé des composants (bouton, check box, text box, dropdown, modal, alert), j’ai donc créé un sous projet commun qui permettra à tous les développeurs d’y avoir accès facilement. Cela a rendu la création et la modification du formulaire plus simple.

Après avoir fini le formulaire et la validation de ce dernier, j’ai commencé une deuxième page qui a pour but de répertorier tous les contrats, de pouvoir les filtrer selon des critères et une fois les contrats choisis de pouvoir les exporter, les imprimer.

Pour cela j'ai dans un premier temps récupérer les données déjà présentes dans la base afin de les afficher dans un tableau. Par la suite j'ai mis en place de manière visuelle les filtres et j'ai commencé à trier les contrats selon les filtres.

Ceci est la vue des filtres.

```
<div class="col">
    <div class="form-check">
        <input class="form-check-input" type="checkbox" id="RenewalModeAll" value="@Checkbox_RM_All"
            onchange="@((e) => { Checkbox_RM_All = (bool)e.Value; All03(); Refresh_Tableau_Datas(); })" checked="@Checkbox_RM_All">
        <label class="form-check-label" for="RenewalModeAll">Tout</label>
    </div>

    <div class="form-check">
        <input class="form-check-input" type="checkbox" id="RenewalMode0"
            onchange="@((e) => { FilterRenewalMode((bool)e.Value, (int)MaintenanceContract.Renewal_Mode.None); Refresh_Tableau_Datas(); })"
            disabled="@Checkbox_RM_All" checked="@Checkbox_RM_All || Filtre_RM.Contains((int) MaintenanceContract.Renewal_Mode.None)">
        <label class="form-check-label" for="RenewalMode0">None</label>
    </div>

    <div class="form-check">
        <input class="form-check-input" type="checkbox" id="RenewalMode1"
            onchange="@((e) => { FilterRenewalMode((bool)e.Value, (int)MaintenanceContract.Renewal_Mode.Tacite); Refresh_Tableau_Datas(); })"
            disabled="@Checkbox_RM_All" checked="@Checkbox_RM_All || Filtre_RM.Contains((int) MaintenanceContract.Renewal_Mode.Tacite)">
        <label class="form-check-label" for="RenewalMode1">Tacite</label>
    </div>

    <div class="form-check">
        <input class="form-check-input" type="checkbox" id="RenewalMode2"
            onchange="@((e) => { FilterRenewalMode((bool)e.Value, (int)MaintenanceContract.Renewal_Mode.Expresse); Refresh_Tableau_Datas(); })"
            disabled="@Checkbox_RM_All" checked="@Checkbox_RM_All || Filtre_RM.Contains((int) MaintenanceContract.Renewal_Mode.Expresse)">
        <label class="form-check-label" for="RenewalMode2">Expresse</label>
    </div>
</div>
```

Et le tri se faisant sur des checkbox.

```
35 references | 0 changes | 0 authors, 0 changes
protected override async Task OnInitializedAsync()
{
    ApiResponse<List<AllSub1300Detail>> response = await Api.Get<List<AllSub1300Detail>>(Area.manager, $"allsubscription130get", "subscription130");
    if (Api.ShowPopupError(response)) return;
    Tableau_db = response.ObjectResult;
    Filtre_SC = new List<int>();
    Filtre_CO = new List<int>();
    Filtre_RM = new List<int>();
    All01();
    All02();
    All03();
    Refresh_Tableau_Datas();
    PageLoaded = true;
}

3 references | 0 changes | 0 authors, 0 changes
public void All01()
{
    if (Checkbox_SC_All)
        Filtre_SC = (from x in Enum.GetValues(typeof(MaintenanceContract.Status_Contract)).Cast<int>() select x).ToList();
    else
        Filtre_SC = new List<int>();
}

4 references | 0 changes | 0 authors, 0 changes
private void FilterStatusContract(bool Ischecked, int value)
{
    if (Ischecked && !Filtre_SC.Contains(value))
    {
        Filtre_SC.Add(value);
        if (Filtre_SC.Contains((int)MaintenanceContract.Status_Contract.Attente)
            && Filtre_SC.Contains((int)MaintenanceContract.Status_Contract.Transmit)
            && Filtre_SC.Contains((int)MaintenanceContract.Status_Contract.Refuse)
            && Filtre_SC.Contains((int)MaintenanceContract.Status_Contract.Accepte))
        {
            Checkbox_SC_All = true;
        }
    }
    else
    {
        if (Filtre_SC.Contains(value))
            Filtre_SC = (from x in Filtre_SC where x != value select x).ToList();
    }
}
```

Comme la mise en place des filtres étaient compliqué et que je ne savais pas comment m'y prendre, j'ai fait des schémas des différentes actions qui se pouvaient se passer sur la page et comment toutes ces actions se faisaient et fonctionnaient. Le but était que je comprenne la méthode et que la solution mise en place ne soit pas trop lourde.

J'ai donc réussi à filtrer les sur les valeurs voulues et obtenir un tableau complet.

Malice

Liste des contrats

Etat du contrat

☒ Tout  
☐ Attentie  
☐ Transmis  
☐ Refuse  
☐ Accepte

Objet du contrat

☒ Tout  
☐ Progieciel  
☐ Materiel  
☐ Progieciel et Materiel

Mode de renouvellement

☒ Tout  
☐ None  
☐ Tacite  
☐ Expresse

Exporter tableau

Reinitialiser

Nom du client	Liste des structures	Adresse1	Ville	Code Postal	Mail	Tarif HT	Détail du tarif	Date de debut	Date de fin	Statut du contrat	Objet du contrat	Mode de renouvellement
XXXX	1 2 3 4	XXXX	XXXX	XXXX	XXXX	1	1	10/11/2020	09/11/2021	Attentie	Progieciel	Tacite
XXXX	abc def ghi	XXXX	XXXX	XXXX	XXXX	2	2	01/01/2021	31/12/2023	Transmis	Progieciel_Materiel	Expresse
XXXX	toto tata titi	XXXX	XXXX	XXXX	XXXX	3	3	01/01/2021	01/03/2023	Accepte	Materiel	None
XXXX	abcde	XXXX	XXXX	XXXX	XXXX	4	4	30/01/2021	31/12/2021	Refuse	Progieciel_Materiel	Tacite
XXXX	azerty	XXXX	XXXX	XXXX	XXXX	150	15	30/11/2020	31/12/2020	Accepte	Progieciel_Materiel	Expresse
XXXX	abc def ghi	XXXX	XXXX	XXXX	XXXX	17	7+10	01/01/2021	31/12/2021	Transmis	Materiel	Expresse
XXXX	aaa bbb ccc	XXXX	XXXX	XXXX	XXXX	85	4*20+5	01/01/2021	31/12/2021	Attentie	Progieciel	Tacite

button1

https://localhost:44304/managerclient/login/loginfrommalice?malicekeys=d371efc393aff333645e7c09d10

Go

Ctrl-C

Enfin j'ai beaucoup cherché et réfléchi à la création et à l'exportation du fichier csv qui contient les données des différents contrats. J'ai dû créer un fichier csv qui contenait toutes les informations qui serait utiles dans la partie publipostage des contrats.

Dans cette méthode se trouve la création du fichier csv ainsi que la partie qui consiste à le remplir.

```
[HttpGet("allsub130createdownloadCSV")]
[AllowAnonymous]
public IActionResult allsub130createdownloadCSV(string stringlistid) // , string namescsvfile
{
    try
    {
        List<int> listid = (from x in stringlistid.Split(';') select Convert.ToInt32(x)).ToList();

        IAbonnement[] AbosAll = Sess.db1.L.AbonnementGet();
        IEntite[] EntityAll = Sess.db1.L.EntiteGet();
        IEntiteAdr[] EntityAdrAll = Sess.db1.L.EntiteAdrGet();
        IVilles[] VillesAll = Sess.db1.L.VillesGet();
        List<AllSub130Detail> listsubs = (from A in AbosAll
            join E in EntityAll on A.InvidE equals E.Id
            join EAdr in EntityAdrAll on E.Id equals EAdr.IDE
            join V in VillesAll on EAdr.IdVile equals V.Id
            where listid.Contains(A.Id)
            select new AllSub130Detail(A.Id, E.Id, A.DateD, A.DateF, E.EntName, EAdr.Mail,
                A.DetailsContract._StatusContract, A.DetailsContract._ContractObject,
                A.DetailsContract._RenewalMode, A.DetailsContract._PriceHT, A.DetailsContract._IndiceSyntec,
                A.DetailsContract._PriceDetail, A.DetailsContract._ListStructures,
                A.DetailsContract._Signatory, A.DetailsContract._ServerPlace, EAdr.Adr1, V.Ville, V.CP.Value)).ToList();

        StringBuilder UserFile = new StringBuilder();
        UserFile.AppendLine("Id;" + "NumeroContrat;" + "Nom du client;" + "Mail;" + "ObjetContrat;" + "ModeRenouvellement;" +
            "ListeStructures;" + "DateDebut;" + "DateFin;" + "DateDujour;" + "AnneContrat;" + "AnneIndiceSyntec;" + "TarifHT;" + "TarifTTC;" + "IndiceSyntec;" +
            "DetailTarif;" + "Signataire;" + "EmplacementServeur;" + "Adresse;" + "Ville;" + "CodePostal");

        foreach (AllSub130Detail Ligne in listsubs)
        {
            int YearIndiceSyntec = Ligne.StartDate.Year - 1;
            decimal PriceTTC = Ligne.PriceHT + (Ligne.PriceHT * 20 / 100);
            int NumeroContrat = Ligne.StartDate.Year * 10000 + Ligne.NumClient;

            UserFile.AppendLine(Ligne.Id + ";" + NumeroContrat + ";" + Ligne.EntityName + ";" + Ligne.EntityMail + ";" +
                Ligne.ContractObject + ";" + Ligne.RenewalMode + ";" + Ligne.ListStructures + ";" +
                Ligne.StartDate.ToLongDateString() + ";" + Ligne.EndDate.ToLongDateString() + ";" + DateTime.Today.ToShortDateString() + ";" + Ligne.StartDate.Year + ";" +
                YearIndiceSyntec + ";" + Montant(Ligne.PriceHT) + ";" + Montant(PriceTTC) + ";" + Ligne.IndiceSyntec + ";" + Ligne.PriceDetail + ";" + Ligne.
                Ligne.Signatory + ";" + Ligne.ServerPlace + ";" + Ligne.Address1 + ";" + Ligne.Ville + ";" + Ligne.CodePostal);
        }
        return File(Encoding.UTF8.GetBytes(UserFile.ToString()), "text/csv", "ContratMaintenance" + ".csv"); // namescsvfile
    }
    catch
    {
        return NoContent();
    }
}
```

Lors de la dernière semaine de stage, j'ai pu finir mon projet en faisant le publipostage des contrats grâce à Word. Cela consistait à récupérer le fichier csv, l'intégrer à un fichier Word (modèle du contrat).



Afin de bien finaliser les contrats j'ai dû apporter certaines données que je n'avais pas prévu au début. Puis j'ai expliqué à Bruno la partie publipostage pour qu'il puisse refaire les contrats.

Tout le long de mon stage Issam m'a aidé, lorsque je ne comprenais pas certaines parties ou que je ne savais pas résoudre un problème, c'est lui qui me disait ce que je devais faire étape par étape. Bruno venait environ une fois par semaine vérifier mon avancement, je pouvais lui poser des questions sur les détails du projet.

### III. Bilan du stage

#### A. Bilan de la mission

Suite à mes sept semaines de stage, j'ai pu finir mon projet et montrer à mon maître de stage comment utiliser ce que j'ai produit.

Dans un premier temps le module de gestion des contrats de maintenance ne sera utilisé que par Amiciel. Le module devra être amélioré de manière à être plus ergonomique pour l'utilisateur et plus performant afin d'être déployé pour les clients.

Lors de ce stage, j'ai pu apprendre de nouvelles technologies, progresser dans ma manière de développer, de réfléchir, d'aborder les problèmes qui surviennent.

#### B. Bilan personnel

Le fait d'être en entreprise m'a beaucoup apporté car j'ai pu voir dans quel environnement un développeur pouvait travailler, comment au sein de l'équipe les développeurs communiquent. De plus, les développeurs ont des petites astuces ou des habitudes qui ne peuvent être vues et apprises qu'en entreprise et c'est ce savoir qui est très intéressant et que malheureusement on ne peut pas l'apprendre dans une salle de classe. Le fait d'arriver dans un très gros projet qui a déjà 15-20 ans qui est complexe et comporte énormément de sous-projets m'a un peu déstabilisé au départ, mais assez rapidement je me suis rendu compte que je travaillais sur certaines parties et se repérer est devenu plus facile. J'ai de nombreuses fois eu des difficultés pour résoudre des erreurs de codes, certaines liées au fait que la technologie utilisée est récente et d'autres à ma difficulté à comprendre l'erreur ou le code lui-même.

## **Conclusion**

Pour conclure, j'ai terminé la réalisation qui m'était demandé durant mon stage, cela m'a permis d'acquérir de nouvelles connaissances, de voir comment est le monde dans une entreprise informatique. J'ai aussi pu voir de quoi j'étais capable et d'en être fière. Ce que j'ai appris me permettra de mettre en valeur de nouvelles connaissances pour la suite.