STAGE Solti



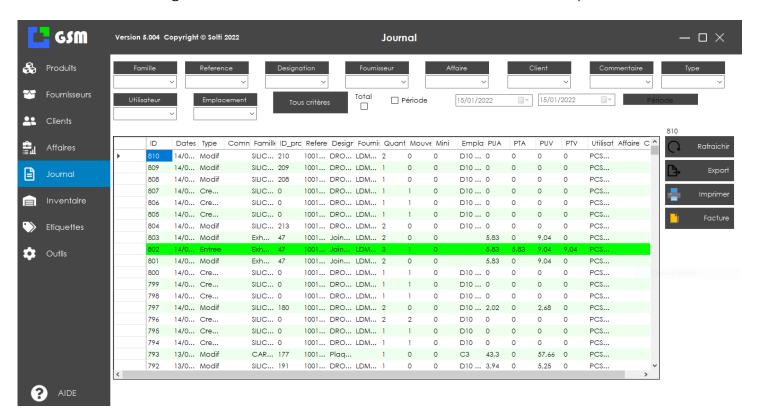
<u>Sommaire</u>

Introduction:	3
Mes missions à Solti :	4
Devis en facture :	4
Cloner une facture :	8
Conclusion	9

Introduction:

Solti est une entreprise basée à Seyssinet-Pariset qui commercialise différents logiciels concernant la gestion de stock. Le logiciel nommé GSM (Gestion de stock modulaire) est le plus utilisé, il permet aux entreprises de gérer leur stock. Elle possède actuellement plus de 1550 client à travers 75 pays différents.

Durant mes semaines de stages mes mission seront l'ajout de nouvelles fonctionnalités sur leurs différents logiciels en fonction des besoins des clients et de l'entreprise.



Mes missions à Solti :

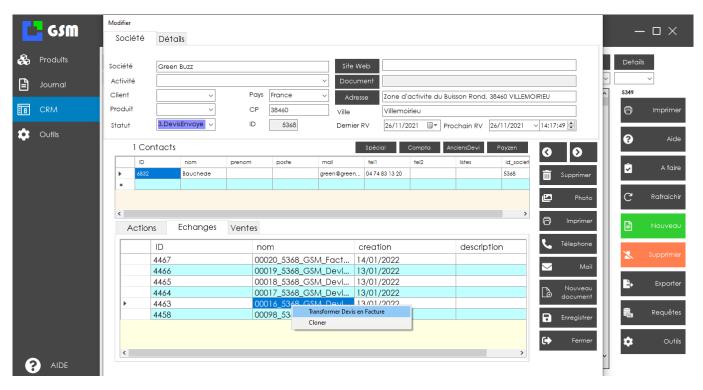
Semaine 2:

Au début de ma première semaine de stage mon tuteur m'a donné une copie des fichiers du logiciel. J'ai dû d'abord apprendre à me familiariser avec le logiciel afin de comprendre le fonctionnement de celui-ci pour ne pas être perdu pour effectuer les taches qui me seront confiés.

Une partie du logiciel est utiliser par l'entreprise afin de gérer ses clients, ma première mission a été de créer une fonctionnalité qui permettra de transformer un devis pour une entreprise en facture.

Devis en facture :

Pour commencer je devais donc faire en sorte que lors d'un clic droit sur une ligne du tableau, je fasse apparaître un menu qui propose de transformer le devis sélectionner en facture.



Grâce à quelques recherches sur internet et l'aide de mon tuteur, je réussis à mettre en place un ContextMenu qui sera afficher lors d'un clic droit.

Pour la suite il fallait que j'apprenne comment manipuler des fichier Excel en C#. Je me suis renseigné sur internet et ai demandé de l'aide par certains moments. J'ai finalement créé une class ExcelTools qui me servira pour mettre toutes mes fonctions dont j'aurais besoin.

```
public class ExcelTools
               string path = "";
               _Application excel = new _Excel.Application();
               Workbook wb;
               Worksheet ws;
               6 références
               public ExcelTools(string path, int Sheet)
                   this.path = path;
                   wb = excel.Workbooks.Open(path);
                   ws = wb.Worksheets[Sheet];
25
               //i = vertical ; j=horizontal
               public string ReadCell(int row, int col)
                   row++;
                   col++;
                   if (ws.Cells[row, col].Value2 != null)
                       return ws.Cells[row, col].Value2.ToString();
                   }
                   else
                       return "";
```

Pour récupérer les informations dont j'avais besoin dans un fichier Excel j'allait devoir me servir de fonction qui lise une cellule par rapport à sa ligne et sa colonne.

Exemple de devis :

1	АВ	C	D	E	F			
16	La solution que nous vous proposons se décompose ainsi :							
17								
18	Abonnement annuel							
19	Référence	Designation	Qté	PU	HT			
20	Licence logiciel GSM	Licence par poste	1	100	100			
21	Maintenance	Support, Mise à jour, Maintenance	1	500	500			
22	Web	Base MySQL, Interface Web, Hébergement	1	1000	1000			
23	Option	Installation et licence MsOffice installée	1	100	100			
24								
25	Tarif unitaire							
26	Référence	Designation	Qté	PU	HT			
.7	Option	Interface pour la lecture code-barres	1	100	100			
8	Option	Impression étiquettes standard de code-barres	1	100	100			
9	Option	Champs supplémentaires, une à 10 colonnes texte	1	100	100			
0	Option	Alarmes par mail	1	100	100			
1	Option	Bordereaux: bons de réception, BL, proforma, factures	1	100	100			
2	Option	Bordereaux: génération de PDF et impression automatique	1	100	100			
3	Option	Impression du journal à votre format	1	100	100			
34	Option	Interface point de vente CASH	1	100	100			
35	Option	Sous-familles de produits	1	100	100			

Il me fallait donc coder une fonction qui extrairais toutes les données d'un tableau. Pour savoir où aller chercher les informations il me fallait un caractère qui m'indiquerais où commencer, j'ai donc choisi de récupérer les informations après la ligne où se trouvait le mot « Référence ».

```
public List<List<string>> ExtractSrcFile(string delimiteur)
   int src_row = 1;
   int src_col= 1;
   List<string> reference = new List<string>();
   List<string> designation = new List<string>();
   List<string> quantite = new List<string>();
   List<string> PU = new List<string>();
   src_row = findStartCell(src_row, src_col, delimiteur);
   while (ReadCell(src_row, src_col) != "")
       reference.Add(ReadCell(src_row, src_col));
       designation.Add(ReadCell(src_row, src_col + 1));
       quantite.Add(ReadCell(src_row, src_col + 2));
       PU.Add(ReadCell(src_row, src_col + 3));
       src_row = src_row + 1;
   List<List<string>> devisList = new List<List<string>>();
   for (int i = 0; i < reference.Count(); i++)
       List<string> ligne = new List<string>();
       ligne.Add(reference[i]);
       ligne.Add(designation[i]);
       ligne.Add(quantite[i]);
       ligne.Add(PU[i]);
       devisList.Add(ligne);
   return devisList;
```

Cette fonction me retourne une liste de liste que je vais pouvoir prendre en paramètre dans ma prochaine fonction qui servira à écrire les donné extraites dans un nouveau fichier Excel. La condition if sert à gérer les anciens modèles de facture et les nouveaux, les nouveaux ayant 2 tableaux à remplir et un seul pour les anciens.

```
oid WriteDstFile(List<List<string>> devisList1, List<List<string>> devisList2, string FileName, CCompany company,List<string> contact)
int dest_row = 1;
int dest_col = 1;
string date = DateTime.Now.ToString("d");
string nomFichier = FileName.Remove(FileName.Length - 5, 5);
ReplaceWord("§SOCIETE§", company.Nom);
ReplaceWord("§CP$ §VILLE§", company.cp + " " + company.ville);
ReplaceWord("§PRENOM$ $NOM$", contact[1] + " " + contact[0]);
ReplaceWord("§ADRESSE§", company.adresse);
ReplaceWord("SDATES", date);
ReplaceWord("SMAILS", contact[2]);
ReplaceWord("SFICHIERS",nomFichier);
if (devisList2.Count == 0)
     dest_row = findStartCell(dest_row, dest_col, "Référence");
     FillTable(dest_row, dest_col, devisList1);
else
    string item = devisList1[devisList1.Count - 1][0];
if (item != "option" )
    dest_row = findStartCell(dest_row, dest_col, "Abonnement annuel");
    FillTable(dest_row, dest_col, devisList1);
    dest_row = 1;
     dest_row = findStartCell(dest_row, dest_col, "Tarif unitaire");
     FillTable(dest_row, dest_col, devisList2);
```

Ici j'ai dû utiliser deux autres fonctions créer précédemment. ReplaceWord afin de remplir l'en tête du document Excel qui comprend les informations de la société cliente.



```
public void ReplaceWord(string delimiter, string word)
{
   List<int> CellFichier = findInAllSheet(delimiter);
   int x = CellFichier[0];
   int y = CellFichier[1];
   WriteCell(x, y, word);
}
```

```
public List<int> findInAllSheet(string word)
{
    List<int> RowCol = new List<int>();
    int row = 1;
    int col = 1;

    while (ReadCell(row, col) != word)
    {
        row = row + 1;
        if (row == 90)
        {
            col = col + 1;
            row = 1;
        }
        if (col > 8)
        {
            return RowCol;
        }
    }
}
RowCol.Add(row);
RowCol.Add(col);
return RowCol;
}
```

Et FillTable pour remplir les tableaux de la facture :

```
public void FillTable(int dest_row,int dest_col, List<List<string>> devisList)
{
    for (int i = 0; i < devisList.Count; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            WriteCell(dest_row + i, dest_col + j, devisList[i][j]);
        }
    }
}</pre>
```

Cloner une facture:

Cette deuxième mission était assez simple et rapide car grâce à ma mission précédente je savais déjà comment procéder. Il me fallait juste ouvrir la facture souhaiter puis la sauvegarder sous un nouveau nom. J'ai donc rajouté dans mon menu au niveau du clic droit une option cloner pour pouvoir accéder à cette nouvelle fonctionnalité.

```
1référence
private void CloneInvoice(string filePath, string FileName)
{
    ExcelTools ET = new ExcelTools(filePath, 1);
    string factureName = ET.SetFactureCloneName( FileName);
    string dst = Environment.CurrentDirectory + @"\DOC\" + factureName;
    ET.SaveAs(dst);
    ET.Close();
    SoltiProcess.Start(dst);
}
```

Conclusion

Durant cette semaine j'ai beaucoup appris sur la programmation en C# et je me sens bien plus à l'aise avec ce langage que je ne maitrisais pas vraiment. Pouvoir mettre du concret dans la programmation en créant des fonctionnalités pour un véritable logiciel est quelque chose qui m'a beaucoup plus. Le vendredi après-midi nous avons fait une sortie « teambuilding» au ski de fond afin de créer un esprit de groupe. J'ai trouvé cette démarche très intéressante car elle démontre qu'en entreprise il est aussi très important d'avoir une bonne cohésion au sein de l'entreprise.

Niko Niko

Lundi	Mardi	Mercredi	Jeudi	Vendredi