

## STAGE Solti




# Sommaire

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.

Version 5.004 Copyright © Solti 2022

Journal

— □ ×

Produits

Fournisseurs

Clients

Affaires

Journal

Inventaire

Etiquettes

Outils

AIDE

Famille

Reference

Designation

Fournisseur

Affaire

Client

Commentaire

Type

Utilisateur

Emplacement

Tous critères

Total

☐ Période

15/01/2022

15/01/2022

Période

ID	Dates	Type	Comn	Famille	ID_prc	Refere	Design	Fourni	Quant	Mouv	Mini	Empla	PUA	PTA	PUV	PTV	Utilisat	Affaire	C
810	14/0...	Modif	SILIC...	210	1001...	DRO...	LDM...	2	0	0	0	D10 ...	0	0	0	0	PCS...		
809	14/0...	Modif	SILIC...	209	1001...	DRO...	LDM...	1	0	0	0	D10 ...	0	0	0	0	PCS...		
808	14/0...	Modif	SILIC...	208	1001...	DRO...	LDM...	1	0	0	0	D10 ...	0	0	0	0	PCS...		
807	14/0...	Cre...	SILIC...	0	1001...	DRO...	LDM...	1	1	0	0	D10 ...	0	0	0	0	PCS...		
806	14/0...	Cre...	SILIC...	0	1001...	DRO...	LDM...	1	1	0	0	D10 ...	0	0	0	0	PCS...		
805	14/0...	Cre...	SILIC...	0	1001...	DRO...	LDM...	1	1	0	0	D10 ...	0	0	0	0	PCS...		
804	14/0...	Modif	SILIC...	213	1001...	DRO...	LDM...	2	0	0	0	D10 ...	0	0	0	0	PCS...		
803	14/0...	Modif	Exh...	47	1001...	Join...	LDM...	2	0	0	0	5,83	0	9,04	0	0	PCS...		
802	14/0...	Entree	Exh...	47	1001...	Join...	LDM...	3	1	0	0	5,83	5,83	9,04	9,04	0	PCS...		
801	14/0...	Modif	Exh...	47	1001...	Join...	LDM...	2	0	0	0	5,83	0	9,04	0	0	PCS...		
800	14/0...	Cre...	SILIC...	0	1001...	DRO...	LDM...	1	1	0	0	D10 ...	0	0	0	0	PCS...		
799	14/0...	Cre...	SILIC...	0	1001...	DRO...	LDM...	1	1	0	0	D10 ...	0	0	0	0	PCS...		
798	14/0...	Cre...	SILIC...	0	1001...	DRO...	LDM...	1	1	0	0	D10 ...	0	0	0	0	PCS...		
797	14/0...	Modif	SILIC...	180	1001...	DRO...	LDM...	2	0	0	0	D10 ...	2,02	0	2,68	0	PCS...		
796	14/0...	Cre...	SILIC...	0	1001...	DRO...	LDM...	2	2	0	0	D10 ...	0	0	0	0	PCS...		
795	14/0...	Cre...	SILIC...	0	1001...	DRO...	LDM...	1	1	0	0	D10 ...	0	0	0	0	PCS...		
794	14/0...	Cre...	SILIC...	0	1001...	DRO...	LDM...	1	1	0	0	D10 ...	0	0	0	0	PCS...		
793	13/0...	Modif	CAR...	177	1001...	Plaq...		1	0	0	0	C3	43,3	0	57,66	0	PCS...		
792	13/0...	Modif	SILIC...	191	1001...	DRO...	LDM...	1	0	0	0	D10 ...	3,94	0	5,25	0	PCS...		

810

Rafraichir

Export

Imprimer

Facture

# 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 tâches qui me seront confiés.

Une partie du logiciel est utilisée 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 sélectionné en facture.

The screenshot displays the 'GSM' software interface. On the left is a sidebar with navigation icons for 'Produits', 'Journal', 'CRM', and 'Outils'. The main area is titled 'Modifier' and contains a 'Société' tab with details for 'Green Buzz', including address, phone, and website. Below this is a table of '1 Contacts' with columns for ID, nom, prenom, poste, mail, tel1, tel2, listes, and id\_societ. A context menu is open over the '13/01/2022' date in the 'listes' column of the first row, showing options 'Transformer Devis en Facture' and 'Cloner'. At the bottom, there are tabs for 'Actions', 'Echanges', and 'Ventes', with a table of invoices showing columns for ID, nom, creation, and description. On the right side, there is a vertical toolbar with various icons like 'Imprimer', 'Aide', 'A faire', 'Rafraichir', 'Nouveau', 'Supprimer', 'Exporter', 'Requêtes', and 'Outils'.

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.

```

2026 1 référence
2027 private void dataGridViewExchanges_MouseClick(object sender, MouseEventArgs e)
2028 {
2029     if (e.Button == MouseButtons.Right)
2030     {
2031         ContextMenu m = new ContextMenu();
2032
2033         int currentMouseOverCol = dataGridViewExchanges.HitTest(e.X, e.Y).ColumnIndex;
2034
2035         if (currentMouseOverCol >= 0)
2036         {
2037             m.MenuItems.Add(new MenuItem("Transformer Devis en Facture", MenuItemNew_Click));
2038             m.MenuItems.Add(new MenuItem("Cloner", MenuItemClone_Click));
2039         }
2040
2041         m.Show(dataGridViewExchanges, new Point(e.X, e.Y));
2042     }
2043 }
2044

```

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.

```

12 13 références
13 public class ExcelTools
14 {
15     string path = "";
16     _Application excel = new _Excel.Application();
17     Workbook wb;
18     Worksheet ws;
19
20     6 références
21     public ExcelTools(string path, int Sheet)
22     {
23         this.path = path;
24         wb = excel.Workbooks.Open(path);
25         ws = wb.Worksheets[Sheet];
26     }
27
28     //i = vertical ; j=horizontal
29     9 références
30     public string ReadCell(int row, int col)
31     {
32         row++;
33         col++;
34         if (ws.Cells[row, col].Value2 != null)
35         {
36             return ws.Cells[row, col].Value2.ToString();
37         }
38         else
39         {
40             return "";
41         }
42     }
43 }
44

```

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 :

	A	B	C	D	E	F	G
16		La solution que nous vous proposons se décompose ainsi :					
17							
18		<b>Abonnement annuel</b>					
19		<b>Référence</b>	<b>Désignation</b>	<b>Qté</b>	<b>PU</b>	<b>HT</b>	
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		<b>Tarif unitaire</b>					
26		<b>Référence</b>	<b>Désignation</b>	<b>Qté</b>	<b>PU</b>	<b>HT</b>	
27		Option	Interface pour la lecture code-barres	1	100	100	
28		Option	Impression étiquettes standard de code-barres	1	100	100	
29		Option	Champs supplémentaires, une à 10 colonnes texte	1	100	100	
30		Option	Alarmes par mail	1	100	100	
31		Option	Bordereaux: bons de réception, BL, proforma, factures	1	100	100	
32		Option	Bordereaux: génération de PDF et impression automatique	1	100	100	
33		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'indiquerai où commencer, j'ai donc choisi de récupérer les informations après la ligne où se trouvait le mot « Référence ».

```

43 public List<List<string>> ExtractSrcFile(string delimitateur)
44 {
45     int src_row = 1;
46     int src_col = 1;
47     List<string> reference = new List<string>();
48     List<string> designation = new List<string>();
49     List<string> quantite = new List<string>();
50     List<string> PU = new List<string>();
51
52     src_row = findStartCell(src_row, src_col, delimitateur);
53
54     while (ReadCell(src_row, src_col) != "")
55     {
56         reference.Add(ReadCell(src_row, src_col));
57         designation.Add(ReadCell(src_row, src_col + 1));
58         quantite.Add(ReadCell(src_row, src_col + 2));
59         PU.Add(ReadCell(src_row, src_col + 3));
60
61         src_row = src_row + 1;
62     }
63
64     List<List<string>> devisList = new List<List<string>>();
65
66     for (int i = 0; i < reference.Count(); i++)
67     {
68         List<string> ligne = new List<string>();
69         ligne.Add(reference[i]);
70         ligne.Add(designation[i]);
71         ligne.Add(quantite[i]);
72         ligne.Add(PU[i]);
73
74         devisList.Add(ligne);
75     }
76
77     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 données 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.

```
public void WriteDstFile(List<List<string>> devisList1, List<List<string>> devisList2, string FileName, CCompany company, List<string> contact)
{
    CExcelDoc MyExcel = new CExcelDoc();
    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("$DATE$", date);
    ReplaceWord("$MAIL$", contact[2]);
    ReplaceWord("$FICHIER$", 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 créées précédemment. ReplaceWord afin de remplir l'en-tête du document Excel qui comprend les informations de la société cliente.

 <b>SOLUTIONS INNOVATIONS TECHNOLOGIES</b>		 37 rue des muriers 38180 Seyssins	 06 86 27 33 51 09 62 32 16 07	 <a href="mailto:info@solti.fr">info@solti.fr</a>
Devis Logiciel	Gestion de Stock	\$SOCIETE\$		
Devis N°	\$FICHIER\$	\$ADRESSE\$		
Date du devis	\$DATE\$	\$CP\$ \$VILLE\$		

```
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]);
        }
    }
}
```

## 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.

```
1 référence
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 maîtrisais 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
