STAGE Solti



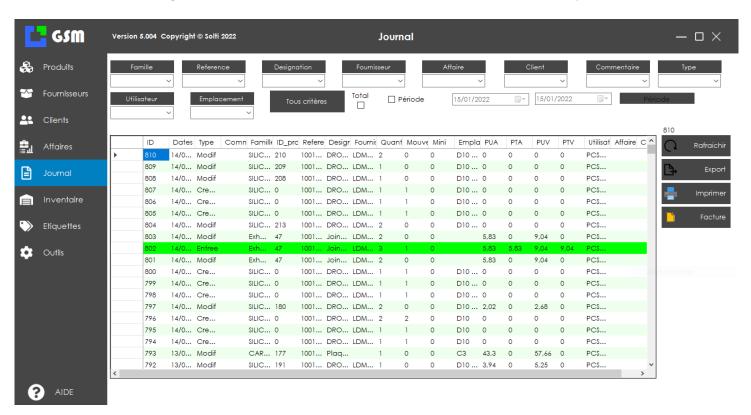
Sommaire

Introduction:	3
Mes missions à Solti :	4
Graphiques :	4
Conclusion	8
Niko Niko	8

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:

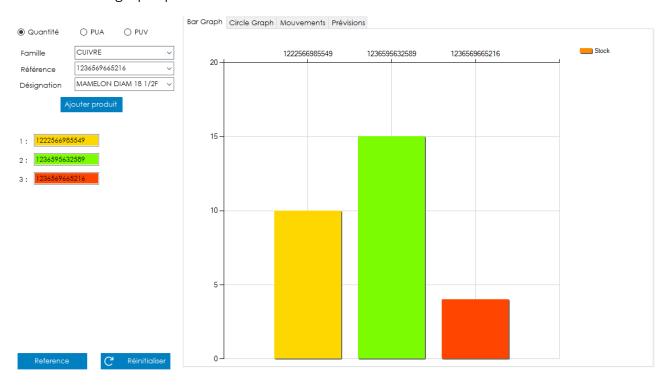
Semaine 5:

Graphiques:

Le patron de SOLTI s'était lancé dans l'idée il y à quelque temps de rajouter une option qui afficherait des graphiques par rapport au stock et aux différents mouvements de stock. Il m'a donc demandé de me lancer dans ce petit projet.

Les données que je devais essayer de traiter était le stock en fonction des années ou des mois et aussi le nombre d'entrée et de sortie.

Pour commencer j'ai utilisé un tabControl afin d'avoir différents onglets où je pourrais mettre mes différents graphiques.



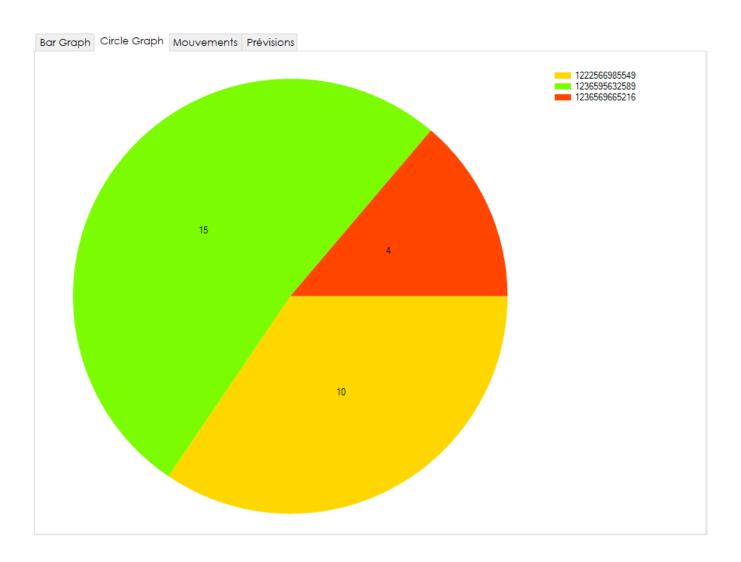
Dans le premier onglet j'ai décider de mettre un graphique en bar qui affiche le stock des différents produits sélectionner précédemment.

```
private void SetBarGraph(Control tBox, Color Color, int value)
{
    string toolTip = value.ToString();
    if (radioButtonPUA.Checked || radioButtonPUV.Checked) toolTip += " €";
    int y = chartStock.Series["Stock"].Points.Count();
    chartStock.Series["Stock"].Points.Add(value);
    chartStock.Series["Stock"].Points[y].AxisLabel = tBox.Text;
    chartStock.Series["Stock"].Points[y].Color = Color;
    chartStock.Series["Stock"].Points[y].ToolTip = toolTip;
}
```

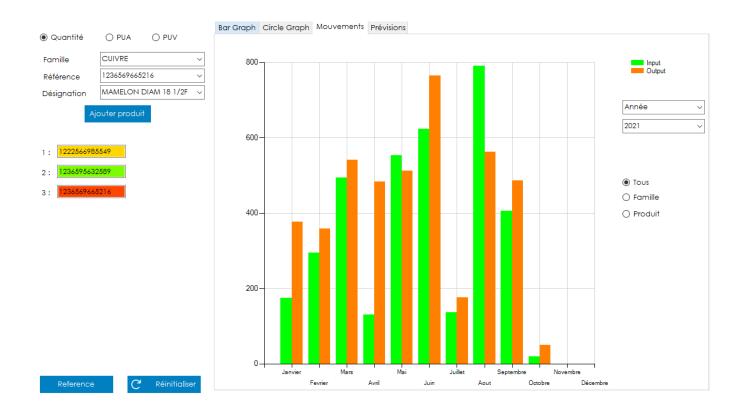
Les valeurs de chaque bar sont récupérées grâce à une requête SQL utiliser plus tôt dans le code.

```
public int GetQty(string designation)
{
    string valeur = "";
    if (radioButtonQuantity.Checked) valeur = "Quantite";
    if (radioButtonPUA.Checked) valeur = "PUA*Quantite";
    if (radioButtonPUV.Checked) valeur = "PUV*Quantite";
    object obj = m_GSM.GetValue("SELECT SUM(" + valeur + ") FROM t_produits WHERE " + column + " = \"" + designation + "\"");
    return Tools.ConvToInt(obj);
}
```

Le deuxième onglet contient le même graphique mais en circulaire avec un code relativement semblable.



Le troisième onglet devient plus intéressant car il va afficher le nombre d'entrée et de sortie de stock. On pourra avoir le choix entre afficher les donné de tous les articles, de seulement une famille ou bien des produits en particulier sélectionner.



Face à toutes ces différentes options je décide de créer une fonction pour construire mes requête SQL en fonction des radioButton qui sont sélectionnés.

```
vate string ReqSql(string format, string type, string date)
                                                                                                                                                                                                                         Aa 🕾 🏞 Document ac
int days = 0;
 string SUM = "Mouvement";
 string tradType;
 if (radioButtonPUA.Checked) SUM = "PUA*" + SUM; if (radioButtonPUV.Checked) SUM = "PUV*" + SUM;
 if (radioButtonFov.tnetkeu) som = Fov + som;
if (type == "Input") { tradType = "Entree"; }
else { tradType = "Sortie"; }
string year = comboBoxYear.SelectedItem.ToString();
string month = (1 + m_ComboMonth.SelectedIndex).ToString();
 if (comboBoxchoice.SelectedIndex == 1) days = DateTime.DaysInMonth(2021, Tools.ConvToInt(month));
string req = "SELECT Format([Dates], '";
req += format + "') as MyDate,SUM(" + SUM + ") AS Quantite FROM t_journal WHERE (Type = '" + tradType + "' OR Type = '" + type + "') AND ";
 if (radioButtonProduct.Checked)
        req += "( ";
int i = 0;
         foreach (Control tBox in m_list)
               if (i > 0) req += " OR ";
req += "" + column + " = \"" + tBox.Text + "\" ";
        req += ") AND ";
 if (radioButtonFamilly.Checked)
        req += "Famille = '" + m_familly + "' AND ";
 ;
if (date == "month") req += "Dates BETWEEN #" + year + "/01/01# AND #" + year + "/12/31# ";
if (date == "day") req += "Dates BETWEEN #" + year + "/" + month + "/01# AND #" + year + "/" + month + "/" + days + "# ";
req += "GROUP BY Format([Dates],'" + format + "')";
```

```
private void FillGraph(string value, int option)
    string[] xval = { "Janvier", "Fevrier", "Mars", "Avril", "Mai", "Juin", "Juillet", "Aout", "Septembre", "Octobre", "Novembre
chartInputOutput.ChartAreas[0].AxisX.CustomLabels.Clear();
    List<string> values = Tools.ConvObjListToStringList(GetInputOutput(value, option));
if (values == null) { MessageBox.Show("La requête échouer"); return; }
    List<string> mouvements = new List<string>();
    List<string> date = new List<string>();
    int index;
    int day;
    int days;
    string month = (1 + m_ComboMonth.SelectedIndex).ToString();
    for (int i = 0; i < values.Count(); i++)
         if (i % 2 == 1)
             string val = Math.Round(Tools.ConvToDouble(values[i]), 0).ToString();
             if (value == "Input") mouvements.Add(val);
             if (value == "Output") mouvements.Add((Tools.ConvToInt(val) * -1).ToString());
        else date.Add(values[i].Substring(0, 2));
    if (option == 1) days = 12;
    else days = DateTime.DaysInMonth(2021, Tools.ConvToInt(month));
```

```
chartInputOutput.ChartAreas[0].AxisX.Maximum = days;
for (int i = 0; i < days; i++)
{
    day = i + 1;

    if (day.ToString().Length == 1) index = date.IndexOf("0" + day);
    else index = date.IndexOf(day.ToString());
    if (index >= 0)
    {
        string toolTip = mouvements[index];
        if (radioButtonPUA.Checked || radioButtonPUV.Checked) toolTip += " €";
        chartInputOutput.Series[value].Points.Add(Tools.ConvToInt(mouvements[index]));
        chartInputOutput.Series[value].Points[i].ToolTip = toolTip;
    }
    else
    {
        chartInputOutput.Series[value].Points.Add(0);
        chartInputOutput.Series[value].Points[i].ToolTip = "0";
    }
    if (option == 0) chartInputOutput.ChartAreas[0].AxisX.CustomLabels.Add(i + 0.5, i + 1.5, day.ToString());
    else chartInputOutput.ChartAreas[0].AxisX.CustomLabels.Add(i + 0.5, i + 1.5, xval[i]);
}
```

Conclusion

Une super cinquième semaine où j'ai pu découvrir de nouvelles chose grâce aux graphiques. J'appréhendais beaucoup de me lancer dedans car cela me semblait très compliqué mais finalement avec de bonnes recherches sur internet j'ai fini par bien comprendre comment ça fonctionnait.

Niko Niko

Lundi	Mardi	Mercredi	Jeudi	Vendredi