

Thomas Fortes,
 Thomas Goepp,
 Yassin Kioua,
 Wladimir Bozenko

- Déclaration des variables globales dans le formulaire de départ :

```
OleDbConnection cx;
List<string> listrecap = new List<string>();
List<string> RecapTypeDeRecette = new List<string>();
List<int> RecapIngredientsEnInt = new List<int>();
List<int> RecapTypeDeRecetteEnInt = new List<int>();
int budget = -1;
int temps = -1;
DataSet ds;

const int COEFFICIENT_TAILLE = 1;

Panel pnlAccueil = new Panel();

Panel pChargement2 = new Panel();
Panel pChargement = new Panel();

Panel pnlEtape1 = new Panel();
Panel pnlFamilles = new Panel();
Panel pnlIngredients = new Panel();
Panel pnlRecap = new Panel();

Panel pnlFiltres = new Panel();
Panel pnlTypeDePlat = new Panel();
Panel pnlPrix = new Panel();
Panel pnlTemps = new Panel();

Panel pnlRecette = new Panel();

string requete;
```

- Procédure d'affichage des ingrédients d'une famille sélectionnée

```
private void AffichageIngredient(object sender, EventArgs e)
{
    pnlIngredients.Controls.Clear();
    int nbIngredient = 0;
    int x = 30;
    int y = 60;

    RadioButton rbIngredients = (RadioButton)sender;

    string codeFamille = "";
    foreach (DataRow dr in ds.Tables["Famille"].Rows)
```

```

    {
        if (dr[1].ToString().CompareTo(rbIngredients.Text) == 0)
        {
            codeFamille = dr[0].ToString();
        }
    }
    foreach (DataRow dtr in ds.Tables["Ingrédients"].Rows)
    {
        if (dtr[2].ToString().CompareTo(codeFamille) == 0)
        {
            if (nbIngredient % 4 == 0)
            {
                x = 40;
                y += 40;
            }
            CheckBox ckb = new CheckBox();

            ckb.Size = new Size(270, 40);
            ckb.Location = new Point(x, y);
            ckb.Text = dtr[1].ToString();
            ckb.BackColor = Color.White;
            pnlIngredients.Controls.Add(ckb);
            pnlIngredients.AutoScroll = true;

            if (listrecap.Contains(ckb.Text) == true)
            {
                ckb.Checked = true;
            }
            x += 270;
            nbIngredient++;
            ckb.CheckedChanged += new
System.EventHandler(recapIngredient);
        }
    }
}

```

- Procédure événementielle qui recherche les recettes correspondant au critères utilisateurs.

```

private void TypeDePlat(object sender, EventArgs e)
{
    RecapTypeDeRecette.Clear();

    foreach (Control ckbTypedeplat in pnlTypeDePlat.Controls)
    {
        CheckBox ckb = (CheckBox)ckbTypedeplat;

        if (ckbTypedeplat is CheckBox)
        {
            if (ckb.Checked == true)
            {

```

```

false)                if (RecapTypeDeRecette.Contains(ckbTypedeplat.Text) ==
                        {
                            RecapTypeDeRecette.Add(ckbTypedeplat.Text);

                        }
                    }
else if (ckb.Checked == false)
{
    if (RecapTypeDeRecette.Contains(ckbTypedeplat.Text) ==
true)
    {
        RecapTypeDeRecette.Remove(ckbTypedeplat.Text);
        ckb.Checked = false;
    }
}

}

}

}

```

Cette Fonction vérifie et enregistre dans une liste les types de plats choisis par l'utilisateur.

```

private void TempsPlat(object sender, EventArgs e)
{
    TextBox txtTemp = (TextBox)sender;
    if(txtTemp.Text != "")
    {
        temps = int.Parse(txtTemp.Text);
    }
}

```

Cette Fonction Récupère le temps que l'utilisateur a indiqué et l'enregistre dans « int Temps »

```

private void PrixChere(object sender, EventArgs e)
{
    RadioButton rb = (RadioButton)sender;
    if (rb.Checked == true)
    {
        budget = 3;
    }
else
{

```

```

        budget = 0;
    }
}

private void PrixMoyen(object sender, EventArgs e)
{
    RadioButton rb = (RadioButton)sender;
    if (rb.Checked == true)
    {
        budget = 2;
    }
    else
    {
        budget = 0;
    }
}

private void PrixBas(object sender, EventArgs e)
{
    RadioButton rb = (RadioButton)sender;
    if (rb.Checked == true)
    {
        budget = 1;
    }
    else
    {
        budget = 0;
    }
}

```

Ces 3 fonctions renvoient 1 2 ou 3 selon le choix du prix de l'utilisateur

```

private void recapIngredient(object sender, EventArgs e)
{
    foreach (Control ancieningr in pnlRecap.Controls)
    {
        if (ancieningr is CheckBox)
        {
            CheckBox ckb = (CheckBox)ancieningr;
            if (listrecap.Contains(ancieningr.Text) == false)
            { ckb.Checked = false; }
        }
    }
    foreach (Control ckbIngreChoisi in pnlIngredients.Controls)
    {
        CheckBox ckb = (CheckBox)ckbIngreChoisi;

        if (ckbIngreChoisi is CheckBox)
        {
            if (ckb.Checked == true)
            {
                if (listrecap.Contains(ckbIngreChoisi.Text) == false)
                {
                    if (listrecap.Count < 3)
                    {
                        listrecap.Add(ckbIngreChoisi.Text);
                    }
                }
            }
        }
    }
}

```

```

        }
        else
        {
            ckb.Checked = false;
        }
    }

}
else if (ckb.Checked == false)
{
    if (listrecap.Contains(ckbIngreChoisi.Text) == true)
    {
        listrecap.Remove(ckbIngreChoisi.Text);
        ckb.Checked = false;
    }
}

}
//ckb.CheckedChanged += new
System.EventHandler(AfficheRecapIngredient);
AfficheRecapIngredient(sender, e);
}
}
private void enleveRecapIngredient(object sender, EventArgs e)
{
    CheckBox chkb = (CheckBox)sender;
    if (chkb.Checked == false)
    {
        pnlRecap.Controls.Remove(chkb);
        listrecap.Remove(chkb.Text);
    }
    foreach (Control ancieningr in pnlIngredients.Controls)
    {
        CheckBox ckb = (CheckBox)ancieningr;
        if (listrecap.Contains(ancieningr.Text) == false)
        { ckb.Checked = false; }
    }

    foreach (Control newIngre in pnlRecap.Controls)
    {
        CheckBox ckb = (CheckBox)newIngre;
        if (newIngre is CheckBox)
        {
            if (ckb.Checked == false)
            {
                if (listrecap.Contains(newIngre.Text) == true)
                {
                    listrecap.Remove(newIngre.Text);
                    pnlRecap.Controls.Remove(newIngre);
                    ckb.Checked = false;
                }
            }
        }
    }

}
AfficheRecapIngredient(sender, e);

```

```
}
```

```
}
```

Ces 2 fonctions permettent d'enregistrer les Ingrédients que l'utilisateur a renseigné

```
private void TransformeFiltreEnInt()
{
    RecapTypeDeRecetteEnInt.Clear();

    for (int i = 0; i < RecapTypeDeRecette.Count; i++)
    {
        string filtre1 = "libCategorie =" + RecapTypeDeRecette[i] + "";
        DataRow[] res1 = ds.Tables["Catégories"].Select(filtre1);
        DataRow ligne1 = res1[0];

        RecapTypeDeRecetteEnInt.Add(Convert.ToInt32(ligne1["codeCategorie"]));
    }
}

public void TransformIngredientEnInt()
{
    RecapIngredientsEnInt.Clear();

    if(RecapIngredientsEnInt.Count !=0)
    {
        for (int i = 0; i < listrecap.Count; i++)
        {
            string filtre = "libIngredient =" + listrecap[i] + "";
            DataRow[] res = ds.Tables["Ingrédients"].Select(filtre);
            DataRow ligne = res[0];

            RecapIngredientsEnInt.Add(Convert.ToInt32(ligne["codeIngredient"]));
        }
    }
}
```

Ces 2 fonctions transforme La liste filtre et la liste Ingrédients en Int (les int qui correspondent à leur code Ingrédients et filtre)

```
public string creeRequete(List<int> typeRecette_, int budget_, int duree_,
List<int> listIng)
{
    //Création d'un booléen pour savoir si on doit mettre un and ou pas
    dans la construction de la requete
    bool AND = false;

    requete = "";
    requete = @"SELECT * FROM Recettes";
```

```

requete //Si la catégorie a été choisie, on ajoute à la construction de la
{
    if (typeRecette_.Count > 0)
    {
        if (typeRecette_.Count >= 1)
        {
            if (AND)
            {
                requete += " AND codeRecette IN (SELECT codeRecette FROM
CatégoriesRecette WHERE codeCategorie =" + typeRecette_[0]+")";
            }
            else
            {
                requete += " WHERE codeRecette IN (SELECT codeRecette
FROM CatégoriesRecette WHERE codeCategorie =" + typeRecette_[0]+")";
            }
            for (int i = 1; i < typeRecette_.Count; i++)
            {
                requete += " OR codeRecette =" + typeRecette_[i];
            }
        }

        AND = true;
    }
}

requete //Si le budget a été choisi, on ajoute a la construction de la
{
    if (budget_ != -1)
    {
        if (AND)
        {
            requete += " AND categPrix <=" + budget_;
        }
        else
        {
            requete += " WHERE categPrix <=" + budget_;
        }
        AND = true;
    }
    if (duree_ != -1)
    {
        if (AND)
        {
            requete += " AND tempsCuisson <=" + duree_;
        }
        else
        {
            requete += " WHERE tempsCuisson <=" + duree_;
        }
    }
}

//On ajoute ensuite les ingrédients à la requete si la taille de la
liste est supérieure à 0
if (listIng.Count > 0)
{
    if (listIng.Count >= 1)
    {
        if (AND)
        {

```

```

        requete += " AND codeRecette IN (SELECT codeRecette FROM
IngrédientsRecette WHERE codeIngredient =" + listIng[0];
    }
    else
    {
        requete += " WHERE codeRecette IN (SELECT codeRecette
FROM IngrédientsRecette WHERE codeIngredient =" + listIng[0];
    }
    for (int i = 1; i < listIng.Count; i++)
    {
        requete += " OR codeIngredient =" + listIng[i];
    }
    requete += ")";
}
}

return requete;
}

```

Cette fonction nous permet de créer la requête sql.

```

private void PageRecette(object sender, EventArgs e)
{
    freePage();
    TransformIngrediantEnInt();
    TransformeFiltreEnInt();
    creeRequete(RecapTypeDeRecetteEnInt,
budget, temps, RecapIngredientsEnInt);

    pnlRecette.Controls.Clear();

    Button btnRetour2 = new Button();
    btnRetour2.Click += new System.EventHandler(pageFiltre);
    btnRetour2.BackgroundImage =
Image.FromFile(System.Environment.GetFolderPath(System.Environment.SpecialFolder.
Personal) +
@"\IUT\SEMESTRE_2\A21\SAEWIN\saewinform\SAE_FRIGO\SAE_FRIGO\bin\Debug\ImageBtnRet
our.png");
    btnRetour2.BackgroundImageLayout = ImageLayout.Stretch;
    btnRetour2.FlatAppearance.BorderSize = 0;
    btnRetour2.FlatStyle = FlatStyle.Flat;
    btnRetour2.Size = new Size(100, 80);
    btnRetour2.Location = new Point(20, 850);

    this.Controls.Add(btnRetour2);

    pnlRecette.Location = new Point(0 * COEFFICIENT_TAILLE, 0 *
COEFFICIENT_TAILLE);
    pnlRecette.Size = new Size(1920 * COEFFICIENT_TAILLE, 1080 *
COEFFICIENT_TAILLE);
    pnlRecette.Name = "Text";
    pnlRecette.BackColor = Color.White;
    pnlRecette.BackgroundImage =
Image.FromFile(System.Environment.GetFolderPath(System.Environment.SpecialFolder.
Personal) +
@"\IUT\SEMESTRE_2\A21\SAEWIN\saewinform\SAE_FRIGO\SAE_FRIGO\bin\Debug\BackgroundE
tape3.png");
    pnlRecette.BackgroundImageLayout = ImageLayout.Zoom;
}

```



```

Panel pnlAfficheRecette = new Panel();
pnlAfficheRecette.Controls.Clear();
pnlAfficheRecette.Location = new Point(200, 300);
pnlAfficheRecette.Size = new Size(1680, 720);
pnlAfficheRecette.BackColor = Color.FromArgb(239, 239, 239);
pnlAfficheRecette.BorderStyle = BorderStyle.FixedSingle;
pnlAfficheRecette.AutoScroll = true;

this.Controls.Add(pnlRecette);
pnlRecette.Controls.Add(pnlAfficheRecette);

OleDbCommand OdCommand = new OleDbCommand(requete, cx);
OleDbDataReader OdDataReader = OdCommand.ExecuteReader();
int X = 5;
int Y = 30;
int nbRecetteTrouvee = 0;
pnlAfficheRecette.Controls.Clear();
while (OdDataReader.Read())
{
    string nomRecette = OdDataReader[1].ToString();

    int prixEnInt = Convert.ToInt32(OdDataReader[5]);
    int duree = Convert.ToInt32(OdDataReader[3]);
    string Image = OdDataReader[4].ToString();

    int CodeRecette = Convert.ToInt32(OdDataReader[0]);
    string prix;
    if (prixEnInt == 1)
    {
        prix = "Bon Marché";
    }
    else if (prixEnInt == 2)
    {
        prix = "Coût Moyen";
    }
    else
    {
        prix = "Assez Chère";
    }

    SAEFrigoBibliothèqueUtilisateur.UserControlRecette ucrecette =
new SAEFrigoBibliothèqueUtilisateur.UserControlRecette(nomRecette, prix, duree,
Image, CodeRecette);
    ucrecette.Location = new Point(X, Y);
    ucrecette.Size = new Size(800, 150);
    pnlAfficheRecette.Controls.Add(ucrecette);

    Y += 170;
    nbRecetteTrouvee++;
}

```

Cette Fonction créer la page Recette et utilise la requête sql pour appeler les UserControls qui correspondent aux attentes des utilisateurs.

```

- Code du User Contrôle qui inclue aussi la génération du PDF
- using System;
- using System.Data.OleDb;
- using System.Data.SqlClient;
- using System.Drawing;
- using System.IO;
- using System.Reflection;
- using System.Windows.Forms;
- using iTextSharp.text;
- using iTextSharp.text.pdf;
- using SAEFrigoBibliothèqueUtilisateur;
- using static System.Net.WebRequestMethods;
-
-
- namespace SAEFrigoBibliothèqueUtilisateur
- {
-     public partial class UserControlRecette : UserControl
-     {
-         private string nomRecette;
-         private String Budget;
-         private int duree;
-         private string ImageRecette;
-         private int CodeRecette;
-         OleDbConnection cx;
-
-         public UserControlRecette(string nomRecette_, string budget_, int
duree_, string ImageRecette_, int CodeRecette_)
-         {
-             InitializeComponent();
-             this.Budget = budget_;
-             this.duree = duree_;
-             this.ImageRecette = ImageRecette_;
-             this.nomRecette = nomRecette_;
-             this.CodeRecette = CodeRecette_;
-             string chaine = @"Provider=Microsoft.jet.OLEDB.4.0; Data Source="
+ System.IO.Path.GetDirectoryPath(Assembly.GetEntryAssembly().Location) +
"\baseFrigo.mdb";
-             cx = new OleDbConnection(chaine);
-         }
-
-         private void UserControlRecette_Load(object sender, EventArgs e)
-         {
-             lblNomRecette.Text = nomRecette;
-             lblPrix.Text = Budget;
-             lblTemps.Text = duree.ToString();
-             pctImageRecette.SizeMode = PictureBoxSizeMode.StretchImage;
-
-             pnlAfficheRecette.BackgroundImage =
System.Drawing.Image.FromFile("BackgroundUserControlsRecette.png");
-             pnlAfficheRecette.BackgroundImageLayout = ImageLayout.Stretch;
-
-             btnRecettePdf.BackgroundImage =
System.Drawing.Image.FromFile("ImageLogoPDF.png");
-             btnRecettePdf.BackgroundImageLayout = ImageLayout.Stretch;
-             btnRecettePdf.Click += btnRecettePdf_Click;
-
-             btnAvis.BackgroundImage =
System.Drawing.Image.FromFile("ImageAvis.png");
-             btnAvis.BackgroundImageLayout = ImageLayout.Stretch;

```

```

-         btnAvis.Click += btnAvis_Click;
-
-
-         btnRecette.BackgroundImage =
System.Drawing.Image.FromFile("Imageplat.png");
-         btnRecette.BackgroundImageLayout = ImageLayout.Stretch;
-
-
-         lblNomRecette.BackColor = Color.White;
-         lblPrix.BackColor = Color.White;
-         lblTemps.BackColor = Color.White;
-         try
-         {
-             if (ImageRecette != "")
-             {
-                 pctImageRecette.Image =
System.Drawing.Image.FromFile(ImageRecette);
-             }
-         }
-         catch { }
-         try
-         {
-             cx.Open();
-
-         }
-         catch (Exception ex)
-         {
-             MessageBox.Show(ex.Message);
-         }
-
-     }
-
-     private void btnRecettePdf_Click(object sender, EventArgs e)
-     {
-         string req = "SELECT
nbPersonnes, categPrix, description, imageDesc, tempsCuisson FROM Recettes WHERE
codeRecette =" + CodeRecette.ToString();
-         OleDbCommand cmd = new OleDbCommand(req, cx);
-         OleDbDataReader read = cmd.ExecuteReader();
-         string requestEtape = "SELECT texteEtape, imageEtape FROM
EtapesRecette WHERE codeRecette = " + CodeRecette.ToString();
-         OleDbCommand cmdEtape = new OleDbCommand(requestEtape, cx);
-         OleDbDataReader readerEtape = cmdEtape.ExecuteReader();
-         string requestIngredients = "SELECT libIngredient FROM
Ingrédients WHERE codeIngredient IN (SELECT codeIngredient FROM
IngrédientsRecette WHERE codeRecette =" + CodeRecette.ToString() + ")";
-         OleDbCommand cmdIngredients = new
OleDbCommand(requestIngredients, cx);
-         OleDbDataReader readerIngredients =
cmdIngredients.ExecuteReader();
-         genererPDF(read, readerEtape, readerIngredients);
-
-     }
-
-     private void pctImageRecette_Click(object sender, EventArgs e)
-     {
-
-     }
-

```

```

private void btnRecette_Click(object sender, EventArgs e)
{
    FormRecetteComplete frmrecette = new
FormRecetteComplete(CodeRecette, ImageRecette, duree);
    frmrecette.ShowDialog();
}

private void btnAvis_Click(object sender, EventArgs e)
{
    FormAvis frmAvis = new FormAvis(ImageRecette, nomRecette);
    frmAvis.ShowDialog();
}

private void genererPDF(OleDbDataReader readerTitre, OleDbDataReader
readerEtape, OleDbDataReader readerIngrédient)
{
    Document document = new Document();
    SaveFileDialog saveFileDialog = new SaveFileDialog()
    {
        Filter = "Fichiers PDF (.pdf)|.pdf",
        FileName = "document.pdf"
    };
    saveFileDialog.ShowDialog();
    PdfWriter writer = PdfWriter.GetInstance(document, new
FileStream(saveFileDialog.FileName, FileMode.Create));
    document.Open();
    PdfPTable tableTitre = new PdfPTable(4);
    tableTitre.WidthPercentage = 100; // Définir la largeur totale de
la table (ici, 100% de la largeur disponible)
    Paragraph saut = new Paragraph(" ");
    while (readerTitre.Read())
    {
        string nom = readerTitre["description"].ToString();
        string cuisson = readerTitre["tempsCuisson"].ToString();
        string nbPersonne = readerTitre["nbPersonnes"].ToString();
        string image = readerTitre["imageDesc"].ToString();
        string cate_prix = readerTitre["categPrix"].ToString();

        iTextSharp.text.Font fontTitre = new
iTextSharp.text.Font(iTextSharp.text.Font.FontFamily.HELVETICA, 20f);
        Chunk chunkTitre = new Chunk("Recette : " + nom, fontTitre);
        chunkTitre.SetUnderline(1f, -2f);
        Paragraph titre = new Paragraph(chunkTitre);
        titre.Alignment = Element.ALIGN_MIDDLE;
        document.Add(chunkTitre);
        document.Add(saut);
    }
    PdfPTable table = new PdfPTable(2);
    table.WidthPercentage = 100; // Définir la largeur totale de la
table (ici, 100% de la largeur disponible)
    int etape = 1;
    while (readerEtape.Read())
    {
        string texteEtape = "Étape " + etape + " :\n \n" +
readerEtape["texteEtape"].ToString();
        string imagePath = readerEtape["imageEtape"].ToString();
        if (!string.IsNullOrEmpty(imagePath))
        {
            iTextSharp.text.Image image =
iTextSharp.text.Image.GetInstance(imagePath);
            float Width = image.Width / 3;
            float Height = image.Height / 3;

```

```

-         image.ScaleToFit(Width, Height);
-         PdfPCell imageCell = new PdfPCell(image);
-         imageCell.Border = PdfPCell.NO_BORDER;
-         imageCell.HorizontalAlignment = Element.ALIGN_CENTER;
-         imageCell.VerticalAlignment = Element.ALIGN_CENTER;
-         imageCell.FixedHeight = 100; // Définir la hauteur de la
cellule de l'image (ici, 100 unités)
-         table.AddCell(imageCell);
-     }
-     Paragraph paragraph = new Paragraph(texteEtape);
-     PdfPCell textCell = new PdfPCell(paragraph);
-     textCell.Border = PdfPCell.NO_BORDER;
-     textCell.FixedHeight = 100; // Définir la hauteur de la
cellule de l'image (ici, 100 unités)
-     textCell.HorizontalAlignment = Element.ALIGN_LEFT;
-     textCell.VerticalAlignment = Element.ALIGN_MIDDLE;
-     table.AddCell(textCell);
-     etape = etape + 1;
- }
- document.Add(table);
- string sep = "-----"
-----";
-     Paragraph separateur = new Paragraph(sep);
-     document.Add(saut);
-     iTextSharp.text.Font font = new
iTextSharp.text.Font(iTextSharp.text.Font.FontFamily.HELVETICA, 16f);
-     Chunk chunk = new Chunk("Ingrédients :", font);
-     chunk.SetUnderline(1f, -2f);
-     document.Add(chunk);
-     document.Add(saut);
-     while (readerIngrédient.Read())
-     {
-         string texteIngrédiant = "- " +
readerIngrédient["LibIngrédient"].ToString();
-         Paragraph ing = new Paragraph(texteIngrédiant);
-         document.Add(ing);
-     }
-     MessageBox.Show("Pdf exporter avec succès");
-     document.Close();
- }
- }
- }

```

La Fonction qui Permet de naviguer en mode liaison de donnée entre les étapes de la recette

```

private void EtapeParEtape(object sender, EventArgs e)
{
    try
    {
        cx.Open();
        int nb;
        pnlRecetteEtapeParEtape.Controls.Clear();
        pnlEtapee.Controls.Clear();
        pnlEtapeRecette.Visible = false;
        btnEtapeParEtape.Visible = false;

        pnlRecetteEtapeParEtape.Visible = true;
    }
}

```

```

btnRecetteComplete.Visible = true;

btnRecetteComplete.Text = "Recette Complete";
btnRecetteComplete.Location = new Point(1700, 280);
btnRecetteComplete.Size = new Size(150, 100);
btnRecetteComplete.BackColor = Color.FromArgb(239, 239, 239);

pnlRecetteEtapeParEtape.Size = new Size(1300, 650);
pnlRecetteEtapeParEtape.Location = new Point(620, 400);
pnlRecetteEtapeParEtape.BackColor = Color.FromArgb(239, 239,
239);

pnlRecetteEtapeParEtape.BorderStyle = BorderStyle.FixedSingle;
pnlRecetteEtapeParEtape.AutoScroll = true;

pnlEtapee.Size = new Size(1200, 400);
pnlEtapee.Location = new Point(67, 100);
pnlEtapee.BackgroundImage =
Image.FromFile("ImageEtapeParEtape.png");
pnlEtapee.BackgroundImageLayout = ImageLayout.Stretch;

pnlRecetteComplete.Controls.Add(btnRecetteComplete);
pnlRecetteComplete.Controls.Add(pnlRecetteEtapeParEtape);

PictureBox pctImageEtape = new PictureBox();
pctImageEtape.Size = new Size(300, 300);
pctImageEtape.Location = new Point(800, 50);

Button btnSuivant = new Button();
btnSuivant.Size = new Size(200, 80);
btnSuivant.Location = new Point(712, 500);

btnSuivant.Tag = 1;
btnSuivant.BackgroundImage = Image.FromFile("ImageSuivant.png");
btnSuivant.BackgroundImageLayout = ImageLayout.Stretch;

Button btnPrecedent = new Button();
btnPrecedent.Size = new Size(200, 80);
btnPrecedent.Location = new Point(387, 500);

btnPrecedent.Tag = 0;
btnPrecedent.BackgroundImage =
Image.FromFile("ImagebtnRetourEtpae.png");
btnPrecedent.BackgroundImageLayout = ImageLayout.Stretch;

Button btnPremiereEtape = new Button();
btnPremiereEtape.Size = new Size(200, 80);
btnPremiereEtape.Location = new Point(62, 500);

btnPremiereEtape.Tag = 2;
btnPremiereEtape.BackgroundImage =
Image.FromFile("ImageEtape1.png");
btnPremiereEtape.BackgroundImageLayout = ImageLayout.Stretch;

```

```

Button btnDernier = new Button();
btnDernier.Size = new Size(200, 80);
btnDernier.Location = new Point(1027, 500);

btnDernier.Tag = 3;
btnDernier.BackgroundImage = Image.FromFile("ImageFinal.png");
btnDernier.BackgroundImageLayout = ImageLayout.Stretch;

btnPrecedent.Click += new EventHandler(btnSuivant_Click);
btnSuivant.Click += new EventHandler(btnSuivant_Click);
btnPremiereEtape.Click += new EventHandler(btnSuivant_Click);
btnDernier.Click += new EventHandler(btnSuivant_Click);

Label lblEtapas = new Label();
lblEtapas.Size = new Size(100, 40);
lblEtapas.Location = new Point(500, 70);
lblEtapas.BackColor = Color.White;

Label lbldescEtape = new Label();
lbldescEtape.Size = new Size(600, 150);
lbldescEtape.Location = new Point(150, 200);
lbldescEtape.BackColor = Color.White;

string requestEtape = "SELECT texteEtape,imageEtape FROM
EtapesRecette WHERE codeRecette = " + CodeRecette.ToString();
OleDbCommand cmdEtape = new OleDbCommand(requestEtape, cx);
OleDbDataReader readerEtape = cmdEtape.ExecuteReader();

bs.DataSource = ds.Tables["EtapesRecette"];

if (numCat == 0) // GAUCHE
{
    bs.MovePrevious();
}
else if (numCat == 1) // DROITE
{
    bs.MoveNext();
}
else if (numCat == 2) // DEBUT
{
    bs.MoveFirst();
}
else // FIN
{
    bs.MoveLast();
}
lblEtapas.Text =
((DataRowView)bs.Current)["NumEtape"].ToString();
lbldescEtape.Text =
((DataRowView)bs.Current)["TexteEtape"].ToString();
pctImageEtape.Image =
Image.FromFile(((DataRowView)bs.Current)["imageEtape"].ToString());

pnlRecetteEtapeParEtape.Controls.Add(pnlEtapee);
pnlRecetteEtapeParEtape.Controls.Add(btnPrecedent);
pnlEtapee.Controls.Add(lblEtapas);
pnlEtapee.Controls.Add(pctImageEtape);
pnlRecetteEtapeParEtape.Controls.Add(btnDernier);

```

```
        pnlRecetteEtapeParEtape.Controls.Add(btnPremiereEtape);  
        pnlRecetteEtapeParEtape.Controls.Add(btnSuivant);  
        pnlEtapee.Controls.Add(lbldescEtape);  
    }  
    catch { MessageBox.Show("erreur"); }  
    finally { cx.Close(); }  
  
}
```


