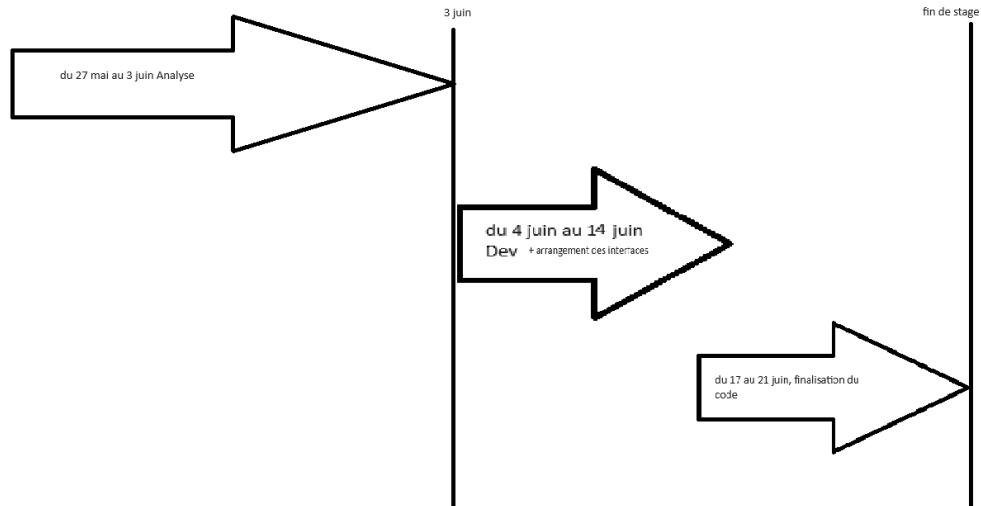


## Qu'ai-je fait pendant le stage ?

Pendant mon stage, je devais mettre en place un moyen d'enregistrer les produits dans la base d'une armoire Kardex.

### Timeline du travail



### Représentation de ce que à quoi doit ressembler la base de données

modelId	Alias	Designation	Typologie	QteMax	PropGestion
01	GEN CODE	APPAT	5	150	P1
02	GENCODE FUGAM	CANNE DE L'HE	3	200	P1
03	GENCODE FUGAM	LEURRE	1	180	P1
04	GEN CODE	APPAT GAP BIS	6	160	P1
05	GEN CODE	CANNE	2	170	P1
06	GENCODE FUGAM	LEURRE	7	190	P1
07	GENCODE FUGAM	CANNE	4	150	P1

#### Première partie |

OBJECTIF : générer un fichier contenant des articles à initialiser dans l'armoire Kardex

Le fichier doit contenir les colonnes suivantes :

- ModelId : numéro unique pour identifier l'article
- Alias : gencode et gencodefugam (code barre)
- Designation : dénomination, appellation
- Typologie : identifiant du bac de l'armoire
- QteMax : nombre maximale d'articles qui peut être rentré
- PropGestion : valeur fixe [P1]

Le fichier sera envoyé sur un serveur pour que l'armoire puisse le traiter.

#### Deuxième partie |

OBJECTIF : Générer un fichier d'entrées des stocks dans l'armoire Kardex

On entre les articles dans les stocks après vérification.

#### Troisième partie |

OBJECTIF : générer un fichier contenant les sorties d'articles présent dans l'armoire Kardex

Le fichier sera envoyé sur un serveur et l'armoire traitera ces changements.

```

namespace inter_patch.Data
{
    8 références
    public class ArticlesDBContext : DbContext
    {
        3 références
        public ArticlesDBContext() : base("name=ArticlesDBContext")
        {
            Database.SetInitializer(new CreateDatabaseIfNotExists<ArticlesDBContext>());
        }

        2 références
        public DbSet<Articles> Article { get; set; }
    }

    0 références
    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        modelBuilder.Entity<Articles>()
            .Property(a => a.ModelId)
            .IsRequired();

        modelBuilder.Entity<Articles>()
            .Property(a => a.QteMax)
            .IsRequired();

        base.OnModelCreating(modelBuilder);
    }
}

```

Initialisation de la base

Initialisation des noms de colonnes de la base de données.

```

4 références
public class Articles
{
    // Il faut changer ModelId et QteMax en int.
    [Key]
    2 références
    public int ModelId { get; set; }
    1 référence
    public string Alias { get; set; }
    1 référence
    public string Designation { get; set; }
    1 référence
    public string Typologie { get; set; }
    2 références
    public int? QteMax { get; set; }
    1 référence
    public string PropGestion { get; set; }
}

```

```

4 références
public partial class Initialisation : Form
{
    private ArticlesDBContext DBContext;

    1 référence
    public Initialisation()
    {
        InitializeComponent();
        DBContext = new ArticlesDBContext();
    }

    // Permet de retourner au menu
    1 référence
    private void button3_Click(object sender, EventArgs e)
    {
        Menu form1 = new Menu();
        form1.Show();
        this.Hide();
    }

    // Permet d'enregistrer les données dans un fichier
    1 référence
    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            var article = new Articles
            {
                /*
                 * Conversion de ModelId et QteMax en entier (int)
                 * (voir Articles.cs)
                 */
                ModelId = int.Parse(ModelId.Text),
                Alias = Alias.Text,
                Designation = Designation.Text,
                Typologie = Typologie.Text,
                QteMax = int.Parse(QteMax.Text),
                PropGestion = PropGestion.Text
            };

            using (var context = new ArticlesDBContext())
            {
                context.Article.Add(article);
                context.SaveChanges();
            }

            MessageBox.Show("Données enregistrées avec succès !");
        }
        catch (Exception ex)
        {
            MessageBox.Show("Erreur d'enregistrement des données : " + ex.Message);
        }
    }

    // Permet d'effacer les données saisies dans les textbox
    1 référence
    private void button4_Click(object sender, EventArgs e)
    {
        ModelId.Clear();
        Alias.Clear();
        Designation.Clear();
        Typologie.Clear();
        QteMax.Clear();
        PropGestion.Clear();
    }
}

```

Formulaire d'Initialisation

ModelId :	<input type="text"/>	Alias :	<input type="text"/>
Désignation :	<input type="text"/>	Typologie :	<input type="text"/>
QteMax :	<input type="text"/>	PropGestion :	<input type="text"/>
Enregistrer		Effacer	
Retour			

```

2 références
public partial class InitialCreate : DbMigration
{
    0 références
    public override void Up()
    {
        CreateTable(
            "dbo.Articles",
            c => new
            {
                ModelId = c.Int(nullable: false),
                Alias = c.String(),
                Designation = c.String(),
                Typologie = c.String(),
                QteMax = c.Int(nullable: true),
                PropGestion = c.String(),
            })
            .PrimaryKey(t => t.ModelId);

        AddColumn("dbo.entree_sorties", "ModelId", c => c.Int(nullable: false));
        CreateIndex("dbo.entree_sorties", "ModelId");
        AddForeignKey("dbo.entree_sorties", "ModelId", "dbo.Articles", "Id", cascadeDelete: true);
    }

    // entree_sorties est une autre table
    0 références
    public override void Down()
    {
        DropForeignKey("entree_sorties", "ModelId", "dbo.Articles");
        DropIndex("dbo.entree_sorties", new[] { "ModelId" });
        DropColumn("dbo.entree_sorties", "ModelId");
        DropTable("dbo.Articles");
    }
}

```

```

4 références
public partial class EntréeSortie : Form
{
    1 référence
    public EntréeSortie()
    {
        InitializeComponent();
        InitializedDataGridView();
    }

    1 référence
    private void InitializedDataGridView()
    {
        if (dataGridView1.Columns.Count == 0)
        {
            dataGridView1.Columns.Add("model", "modelId");
            dataGridView1.Columns.Add("alias", "Alias");
            dataGridView1.Columns.Add("designation", "Désignation");
            dataGridView1.Columns.Add("typologie", "Typologie");
            dataGridView1.Columns.Add("qtemax", "QteMax");
            dataGridView1.Columns.Add("propgestion", "PropGestion");
        }
    }

    // Permet de retourner au menu
    1 référence
    private void retournerAuMenuToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Menu form1 = new Menu();
        form1.Show();
        this.Hide();
    }

    // Quand on appuie sur ce bouton, cela importe un fichier dans le DataGridView
    1 référence
    private void fichierToolStripMenuItem_Click(object sender, EventArgs e)
    {
        using (var context = new ArticlesDBContext())
        {
            var article = context.Article.ToList();

            dataGridView1.DataSource = article;
        }
    }
}

```

