

Validation compétences

Compétences :

A4.1.2 Conception ou adaptation de l'interface utilisateur d'une solution applicative

A4.1.6 Gestion d'environnements de développement et de test

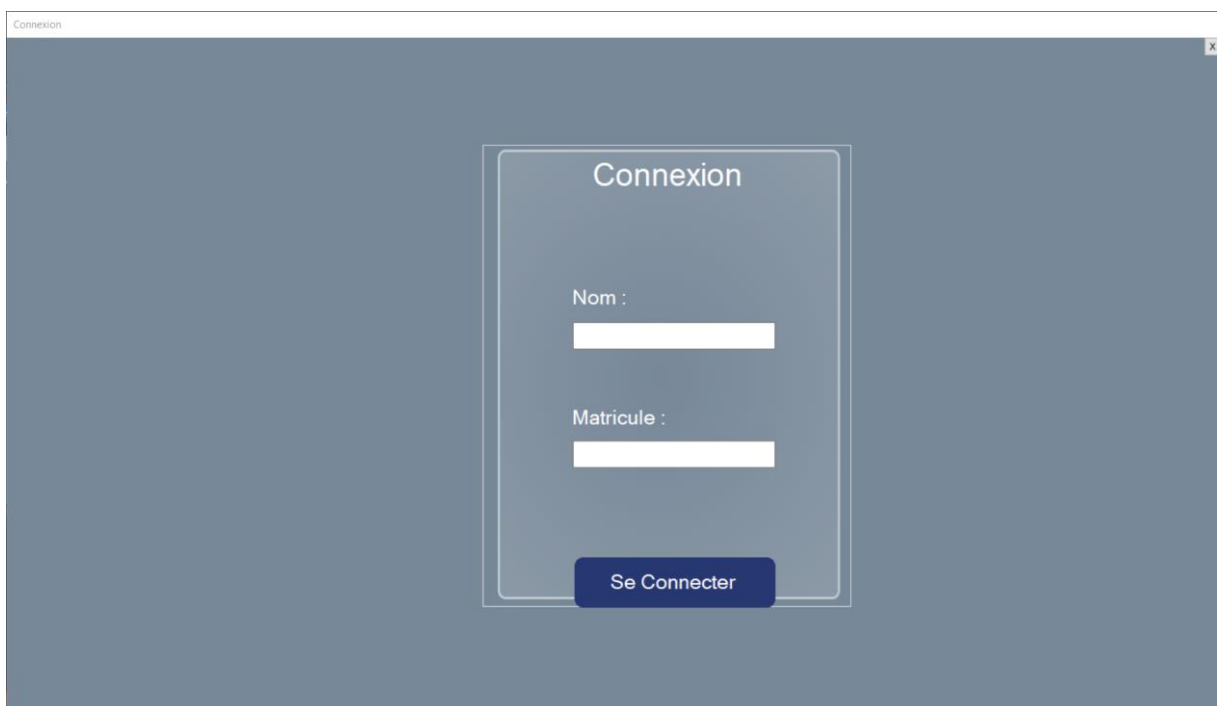
A4.1.7 Développement, utilisation ou adaptation de composants logiciels

A4.1.8 Réalisation des tests nécessaires à la validation d'éléments adaptés ou développés

A5.2.1 Exploitation des référentiels, normes et standards adoptés par le prestataire informatique

A5.2.4 Étude d'une technologie, d'un composant, d'un outil ou d'une méthode

A4.1.2 Conception ou adaptation de l'interface utilisateur d'une solution applicative



The image shows a screenshot of a web application's login interface. The background is a solid dark blue-grey color. In the top-left corner, the word "Connexion" is written in a small, light font. Centered on the screen is a white rectangular box with a thin grey border. Inside this box, the title "Connexion" is displayed in a bold, dark grey font. Below the title, there are two input fields: the first is labeled "Nom :" and the second is labeled "Matricule :". Both labels are in a dark grey font. Each label is followed by a white rectangular input field. At the bottom of the white box, there is a dark blue button with the text "Se Connecter" in white. The overall design is clean and modern.

A4.1.6 Gestion d'environnements de développement et de test

En utilisant Visual Studio 2017 pour créer une application WindowsForm C#, j'ai inséré et laissé en commentaires les potentiels bouts de code de test, ainsi pouvant être réutilisé à certains endroits en cas de bug afin de tester. De plus la mise en place d'un système de test (dans le code) pour l'entrée de caractères incorrectes en fonction de la situation a également été mis en place :

```
private void lvConsultation_Click(object sender, EventArgs e)
{
    try
    {
        btAnnuler.Enabled = true;
        btModifier.Enabled = true;
        btSupprimer.Enabled = true;
        btAnnuler.Visible = true;
        btModifier.Visible = true;
        btSupprimer.Visible = true;
        //tbSaisieNom.Text = lvConsultation.SelectedIndices[0].ToString();
        ligneSelect = lvConsultation.SelectedIndices[0];
    }
}
```

```
int[] tabInt = new int[10] {0,1,2,3,4,5,6,7,8,9};
6 références
private bool ChaineTest (string chaine,string choice)
{
    bool numeric = false;
    bool alpha = false;
    bool alphanumeric = false;
    int o = 0;
    while (o < chaine.Length)
    {
        for (int i = 0; i < tabInt.Length; i++)
        {
            if (chaine.Substring(o, 1) == tabInt[i].ToString())
            {
                numeric = true;
                alphanumeric = true;
            }
        }
        if (alphanumeric == false)
        {
            alpha = true;
        }
        alphanumeric = false;
        o++;
    }
    if (choice == "Num")
    {
        return numeric;
    }
    else return alpha;
}
```

A4.1.7 Développement, utilisation ou adaptation de composants logiciels

Pour notre solution, la compréhension du fonctionnement de l'utilisation de fichier.csv sous C# devait se faire. Ensuite il ne restait plus qu'à utiliser les listes de notre programme pour afficher les données contenues dans notre csv, et les modifier.

Connexion

Consultation liste

Matricule	Nom	Prénom	Autorisation
8541	FORTET	Patrick	I
8541	FORTET	Patrick	E
9632	DREAU	Valerie	E
1789	QUENTIN	Antoine	E
1789	QUENTIN	Antoine	I
5479	GUILLET	Maud	E
3254	GARNIER	Theo	I
3254	GARNIER	Theo	E
9632	DAUMY	Nicolas	E
4521	PIROT	Lea	E
1236	MONNET	Michel	E
1236	MONNET	Michel	I
7413	VANNIER	Louis	E
5874	CHAMBON	Patrick	T
3254	GARNIER	Theo	E
1306	LECOUBLET	Theo	T

Sélectionnez le type d'autorisation d'accès:

Bâtiment

Salle informatique

Nom :

Prénom :

Matricule :

Accès : E I T

	A	B	C	D
1	8541	FORTET	Patrick	I
2	8541	FORTET	Patrick	E
3	9632	DREAU	Valerie	E
4	1789	QUENTIN	Antoine	E
5	1789	QUENTIN	Antoine	I
6	5479	GUILLET	Maud	E
7	3254	GARNIER	Theo	I
8	3254	GARNIER	Theo	E
9	9632	DAUMY	Nicolas	E
10	4521	PIROT	Lea	E
11	1236	MONNET	Michel	E
12	1236	MONNET	Michel	I
13	7413	VANNIER	Louis	E
14	5874	CHAMBON	Patrick	T
15	3254	GARNIER	Theo	E
16	1306	LECOUBLET	Theo	T
17				

A4.1.8 Réalisation des tests nécessaires à la validation d'éléments adaptés ou développés

Afin de tester et en cas de problème, tout les tests unitaires, le code associer à tout problème, et le code de test (comme l'affichage de l'acquisition des données) sont laissé en commentaire dans le code comme ci-dessous afin de facilité la réutilisation de test précédents.

```
14888888
private void btModifier_Click(object sender, EventArgs e)
{
    //StreamWriter FichierDigicodPerso = new StreamWriter("digicod_perso_csv.csv", true, Encoding.Default);
    FichierDigicodPerso.WriteLine(tbSaisieMatricule.Text + ";" + tbSaisieNom.Text.ToUpper() + ";" + tbSaisiePrenom.Text.Substring(0, 1).ToUpper() + tbSaisiePrenom.Text.Substring(1).ToLower() + ";" + AjoutAcces + ";");
    FichierDigicodPerso.Close();
    Initialisation_load();
    File.CreateText("digicod_perso_csv.csv");
    if (tbSaisieNom.Text != "" && tbSaisieMatricule.Text != "" && tbSaisiePrenom.Text != "" && ((cbAccesE.Checked || cbAccesI.Checked || cbAccesT.Checked) == true) && ChainTest(tbSaisieNom.Text, "Num") == false && ChainTest(tbSaisieMatricule.Text, "Alpha") == false)
    {
        int x = 0;
        StreamWriter FichierReecritureDigicode = File.CreateText("digicod_perso_csv.csv");
        do
        {
            if (x != ligneSelect)
            {
                FichierReecritureDigicode.WriteLine(tabEmploye[x].matricule + ";" + tabEmploye[x].nom + ";" + tabEmploye[x].prenom + ";" + tabEmploye[x].autorisation + ";");
            }
            else FichierReecritureDigicode.WriteLine(tbSaisieMatricule.Text + ";" + tbSaisieNom.Text.ToUpper() + ";" + tbSaisiePrenom.Text.Substring(0, 1).ToUpper() + tbSaisiePrenom.Text.Substring(1).ToLower() + ";" + AjoutAcces + ";");
            x++;
        } while (x < nbrLigne);
        FichierReecritureDigicode.Close();
        Initialisation_load();
        creationliste(Batiment, Informatique);
    }
    else MessageBox.Show("Valeurs Invalides");
}
```

A5.2.4 Étude d'une technologie, d'un composant, d'un outil ou d'une méthode

Ainsi pour utiliser les fichiers externes de type csv, l'utilisation d'une nouvelle méthode nommée StreamWriter a été faite (Voir docs.microsoft.com) :

```
if (tbSaisieNom.Text != "" && tbSaisieMatricule.Text != "" && tbSaisiePrenom.Text != "" && ((cbAccesE.Checked || cbAccesI.Checked || cbAccesT.Checked) == true) && ChainTest(tbSaisieNom.Text, "Num") == false && ChainTest(tbSaisiePrenom.Text, "Num") == false && ChainTest(tbSaisieMatricule.Text, "Alpha") == false)
{
    int x = 0;
    StreamWriter FichierReecritureDigicode = File.CreateText("digicod_perso_csv.csv");
    do
    {
        if (x != ligneSelect)
        {
            FichierReecritureDigicode.WriteLine(tabEmploye[x].matricule + ";" + tabEmploye[x].nom + ";" + tabEmploye[x].prenom + ";" + tabEmploye[x].autorisation + ";");
        }
        else FichierReecritureDigicode.WriteLine(tbSaisieMatricule.Text + ";" + tbSaisieNom.Text.ToUpper() + ";" + tbSaisiePrenom.Text.Substring(0, 1).ToUpper() + tbSaisiePrenom.Text.Substring(1).ToLower() + ";" + AjoutAcces + ";");
        x++;
    } while (x < nbrLigne);
    FichierReecritureDigicode.Close();
    Initialisation_load();
    creationliste(Batiment, Informatique);
}
else MessageBox.Show("Valeurs Invalides");
```