

# Validation des compétences

Projet Personnel Encadré Digicod



# Sommaire des compétences

## **A1.4.2 Évaluation des indicateurs de suivi d'un projet et justification des écarts**

- C1.4.2.1 Suivre l'exécution du projet
- C1.4.2.2 Analyser les écarts entre temps prévu et temps consommé

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

- C4.1.2.1 Définir les spécifications de l'interface utilisateur de la solution applicative
- C4.1.2.2 Maquetter un élément de la solution applicative

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

- C4.1.6.1 Mettre en place et exploiter un environnement de développement
- C4.1.6.2 Mettre en place et exploiter un environnement de test

## **A4.1.7 Développement, utilisation ou adaptation de composants logiciels**

- C4.1.7.1 Développer les éléments d'une solution

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

- C4.1.8.1 Élaborer et réaliser des tests unitaires

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

- C5.2.1.1 Évaluer le degré de conformité des pratiques à un référentiel, à une norme ou à un standard adopté par le prestataire informatique
- C5.2.1.2 Identifier et partager les bonnes pratiques à intégrer

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

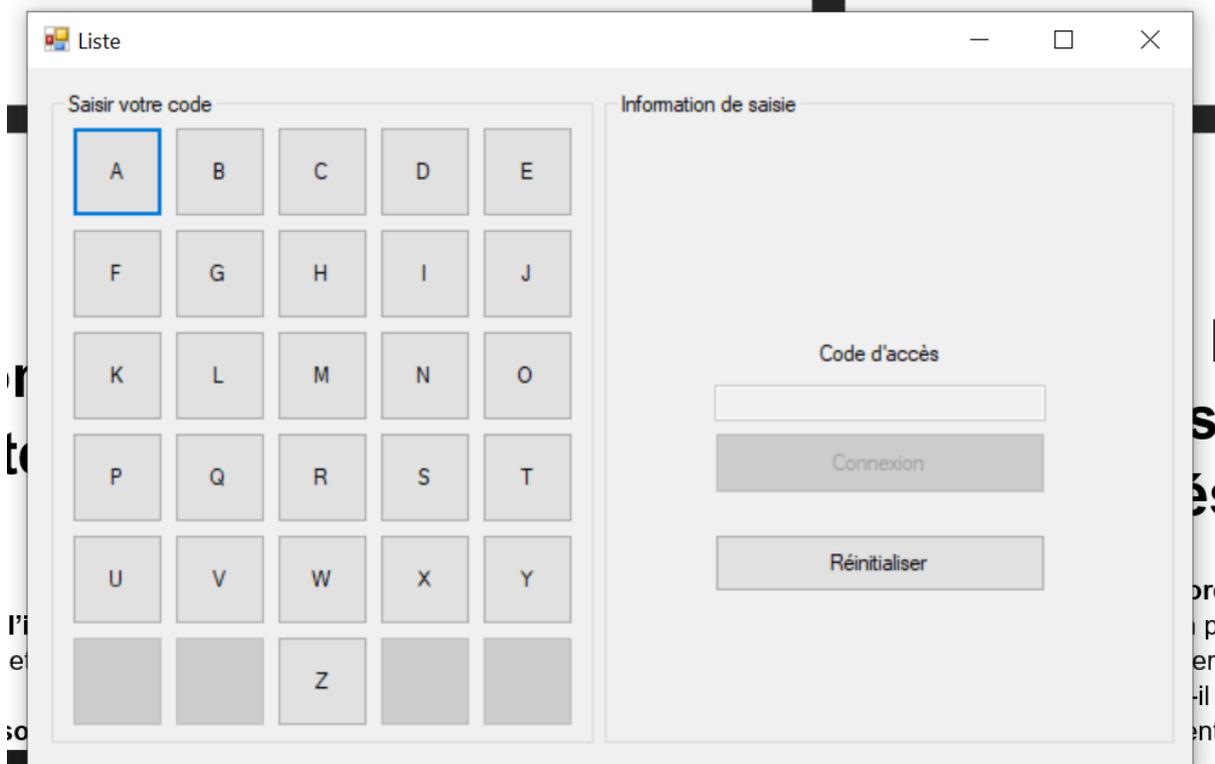
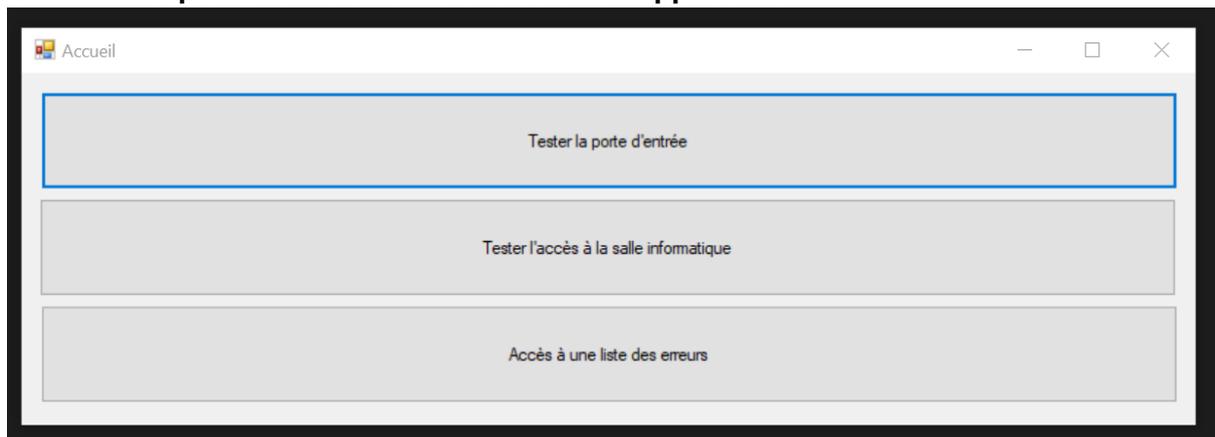
- C5.2.4.1 Se documenter à propos d'une technologie, d'un composant, d'un outil ou d'une méthode

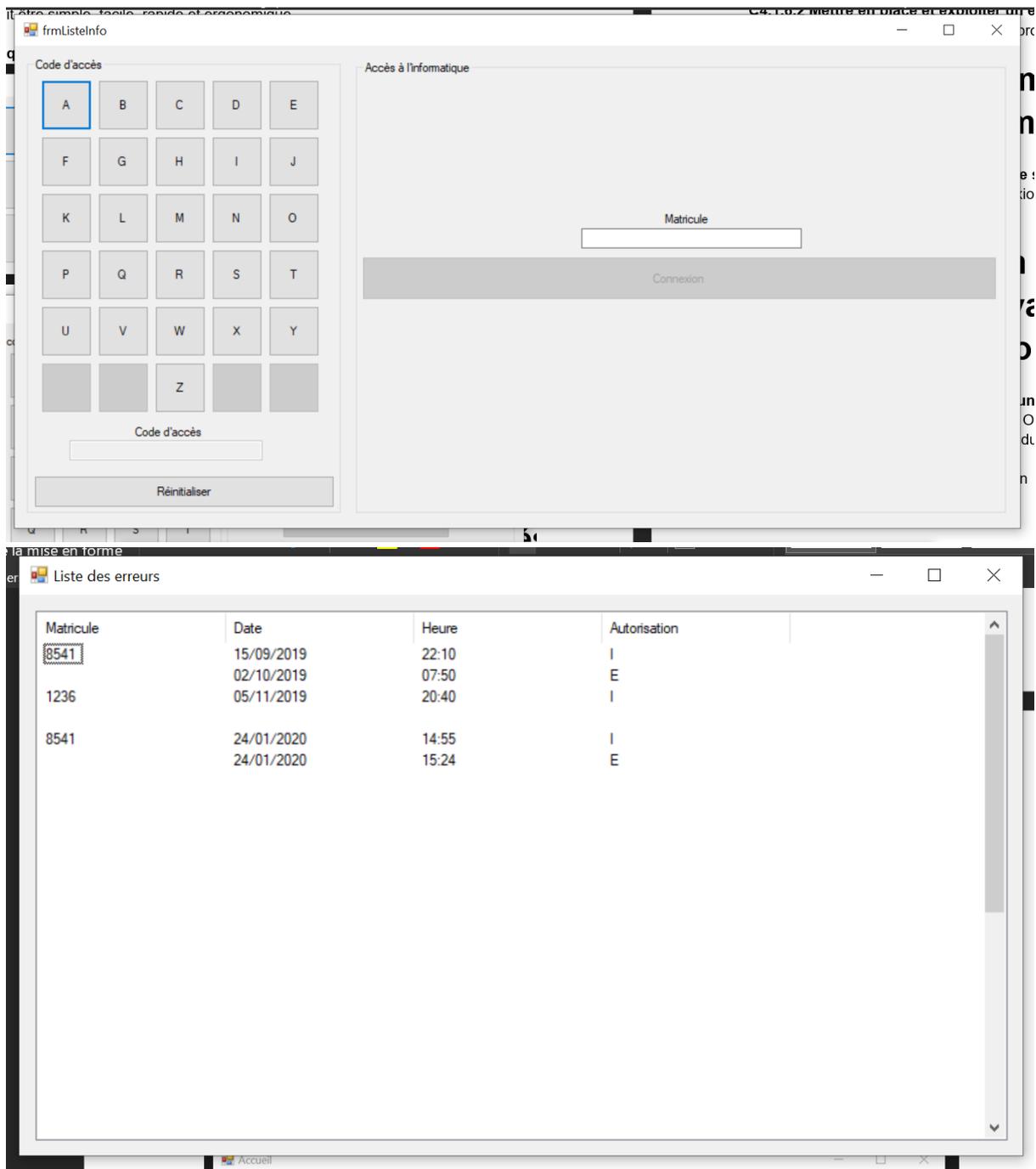
# A4.1.2 Conception ou adaptation de l'interface utilisateur d'une solution applicative

## C4.1.2.1 Définir les spécifications de l'interface utilisateur de la solution applicative

Le test devait être simple, facile, rapide et ergonomique.

## C4.1.2.2 Maquetter un élément de la solution applicative





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

### C4.1.6.1 Mettre en place et exploiter un environnement de développement

Avant de lancer un test de mon application, j'ai regardé si le logiciel m'informe si des erreurs sont présentes pour éviter d'avoir un plantage ou d'une erreur. Pratique pour connaître ses erreurs. Faire attention aussi à nos règles de nommages et des commentaires dans le code.

#### **C4.1.6.2 Mettre en place et exploiter un environnement de test**

Pour tester le code et continuer de développer, j'ai utilisé les outils de débogage de Visual Studio Community pour savoir si le programme fonctionne correctement et si dans le cas contraire je tente de comprendre la raison, le comment du pourquoi.

## **A4.1.7 Développement, utilisation ou adaptation de composants logiciels**

#### **C4.1.7.1 Développer les éléments d'une solution**

J'ai 3 Windows Forms sur mon projet, l'une d'elle permet l'accès à toutes les autres plateformes d'essai de test, la seconde permet de simuler l'accès par code d'accès uniquement, la troisième permet l'accès avec un code d'accès par matricule et en quatrième il y a la liste des erreurs.

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

#### **C4.1.8.1 Élaborer et réaliser des tests unitaires**

- Avoir une interface qui permet l'accès à nos essais
- Si le code d'accès n'est pas bon, on enregistre la date, le bâtiment, etc
- Idem pour l'accès avec un code et un matricule pas correct
- Faire attention aux systèmes de dates

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

C5.2.1.1 Évaluer le degré de conformité des pratiques à un référentiel, à une norme ou à un standard adopté par le prestataire informatique

C5.2.1.2 Identifier et partager les bonnes pratiques à intégrer

```
public partial class frmIndex : Form
{
    public frmIndex()
    {
        InitializeComponent();
    }

    private void btCode_Click(object sender, EventArgs e)
    {
        frmCode code = new frmCode();
        code.Show();
    }

    private void btListe_Click(object sender, EventArgs e)
    {
        frmListe liste = new frmListe();
        liste.Show();
    }

    private void frmIndex_Load(object sender, EventArgs e)
    {
    }

    private void btErreur_Click(object sender, EventArgs e)
    {
        frmErreur erreur = new frmErreur();
        erreur.Show();
    }
}
```

```

private void frmListeInfo_Load(object sender, EventArgs e)
{
    btInformatique.Enabled = false;
    tbNomInformatique.Enabled = true;
    tbEntree.Enabled = false;

    btEntreeVide1.Enabled = false;
    btEntreeVide2.Enabled = false;
    btEntreeVide3.Enabled = false;
    btEntreeVide4.Enabled = false;

    StreamReader monCode = new StreamReader("digicod_secure.csv");

    int i = 0;
    string codeLigne;

    while (!monCode.EndOfStream)
    {
        codeLigne = monCode.ReadLine();
        string[] colonnes = new string[4];
        colonnes = codeLigne.Split(';');
        tabDigicod_Code[i].idPorte = colonnes[0];
        tabDigicod_Code[i].dateDebut = colonnes[1];
        tabDigicod_Code[i].dateFin = colonnes[2];
        tabDigicod_Code[i].codeEncrypt = colonnes[3];
        i++;
    }

    monCode.Close();

    StreamReader monCompte = new StreamReader("digicod_perso.csv");

    int j = 0;
    string compteligne;

    while (!monCompte.EndOfStream)
    {
        compteligne = monCompte.ReadLine();
        string[] colonnes = new string[4];
        colonnes = compteligne.Split(';');
        tabDigicod_Compte[j].matricule = colonnes[0];
        tabDigicod_Compte[j].nom = colonnes[1];
        tabDigicod_Compte[j].prenom = colonnes[2];
        tabDigicod_Compte[j].autorisation = colonnes[3];
        j++;
    }

    monCompte.Close();
}

```

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

## C5.2.4.1 Se documenter à propos d'une technologie, d'un composant, d'un outil ou d'une méthode

GitHub c'est la façon dont les gens construisent les logiciels. Ils soutiennent une communauté où plus de 31 millions de personnes apprennent, partagent et travaillent ensemble pour créer des logiciels.

**GitHub** is how people build software

We're supporting a community where more than 31 million\* people learn, share, and work together to build software.

October 2007 First commit	San Francisco Headquarters	100 million* Repositories hosted
------------------------------	-------------------------------	-------------------------------------

\* As of November 2018

We're working hard to build a supportive, welcoming place for users and GitHubbers alike.  
[Learn more about our commitment to diversity and inclusion.](#)

On peut y mettre du contenu non-dynamique, un portfolio, une vitrine sur GitHub. Par exemple, voici un exemple de projet (ce projet est intégré à une appli Android) :

The screenshot shows a GitHub repository interface. At the top, it displays repository statistics: 42 commits, 1 branch, 0 releases, 1 environment, 1 contributor, and GPL-3.0 license. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The main content area shows a list of commits by user 'bryx95'. The latest commit is 'Update news.js' from 8 hours ago. Below it, a list of files and their update times is shown:

File	Update	Time
admin	Update index.html	a day ago
annuaire	Update index.html	a day ago
css	Update style.css	a day ago
js	Update news.js	8 hours ago
LICENSE	Initial commit	3 days ago
README.md	Update README.md	2 days ago
index.html	Update index.html	a day ago