

Document de validation de compétence

A4.2.1 Analyse et correction d'un dysfonctionnement, d'un problème de qualité de service ou de sécurité

C4.2.1.2 Repérer les composants à l'origine du dysfonctionnement

Certaines données ne respectaient pas la nomenclature imposée par GSB, par la suite d'un certain changement la classe Visite ne fonctionnait plus en raison de la non-acceptation de certaines données. Il a donc fallu modifier le code de la classe et partout où la classe était appelée afin que tout fonctionne correctement. Ainsi que modifier graphiquement l'application pour répondre à la nomenclature.

C4.2.1.3 Concevoir les mises à jour à effectuer

Pour ces mises à jour le code va être modifié ainsi que certains formulaires qui vont devoir être graphiquement modifiés.

C4.2.1.4 Réaliser les mises à jour

```
int numVisite;
DateTime dateVisite;
int nbProduitsPresentes;
int nbProduitsRetenus;
visiteur leVisiteurs;
Professionnel leProfessionnels;
int tempsVisite;
int noteAccueil;

//constructeur permettant d'initialiser un objet visite
1 référence
public Visite(DateTime uneDateVisite, int unNbProduitsRetenus, int unNbProduitsPresentes, visiteur levisiteur, Professionnel leprofessionnel, int leTempsVisite, int laNoteAccueil)
{
    this.numVisite = Globale.lesVisites.Count;
    this.dateVisite = uneDateVisite;
    this.nbProduitsPresentes = unNbProduitsPresentes;
    this.nbProduitsRetenus = unNbProduitsRetenus;
    this.leVisiteurs = levisiteur;
    this.leProfessionnels = leprofessionnel;
    this.tempsVisite = leTempsVisite;
    this.noteAccueil = laNoteAccueil;
}
```

Exemple d'endroit ou la modification a été effectuée :

```
// --> lesVisites
Globale.lesVisites = new List<Visite>();

maCommande = connection.CreateCommand();
maCommande.CommandText = "select * from tb_visite";
connection.Open();
mLigne = maCommande.ExecuteReader();
while (mLigne.Read())
{
    string[] valeurColonnes = new string[9];
    for (int i = 0; i < mLigne.FieldCount; i++) valeurColonnes[i] = mLigne.GetValue(i).ToString();

    int numero = int.Parse(valeurColonnes[0]);
    DateTime dateVisite = DateTime.Parse(valeurColonnes[1]);
    int nbProdPres = int.Parse(valeurColonnes[2]);
    int nbProdRet = int.Parse(valeurColonnes[3]);
    string matricule = valeurColonnes[4];
    string nom = valeurColonnes[5];
    string prenom = valeurColonnes[6];
    int leTempsVisite = int.Parse(valeurColonnes[7]);
    int laNoteAcceuil = int.Parse(valeurColonnes[8]);

    visiteur leVisiteur = rechercherVisiteurVisite(matricule);
    Professionnel leProfessionnel = rechercherProfessionnelVisite(nom, prenom);

    Visite uneVisite = new Visite(dateVisite, nbProdRet, nbProdPres, leVisiteur, leProfessionnel, leTempsVisite, laNoteAcceuil);
    Globale.lesVisites.Add(uneVisite);
}
connection.Close();
```

Il y a également eu des modifications graphiques à effectuer :

Saisie d'une nouvelle visite

Date de la visite :

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Aujourd'hui : 10/06/2020

Visiteur :

Professionnel :

Nombre de produits présentés :

Nombre de produits retenus :

Temps de la visite (en minute) :

Note accueil :

- excellent accueil
- échange très satisfaisant
- échange satisfaisant
- peu intéressé
- pas du tout intéressé