

Compte rendu Développement en C# de la partie application lourde

Table des matières :

1. Logiciels utilisés

2. Arborescence des fichiers

3. Schéma de la base de données

4. Modification code source

5. Gestion de version

7. Compétences

1. Logiciels utilisés :

- Visual Studio Community 2022
- MySQL
- C# (Windows Forms)
- Windows 64 bit
- Git et Github

Compte rendu Développement en C# de la partie application lourde

2. Arborescence des fichiers

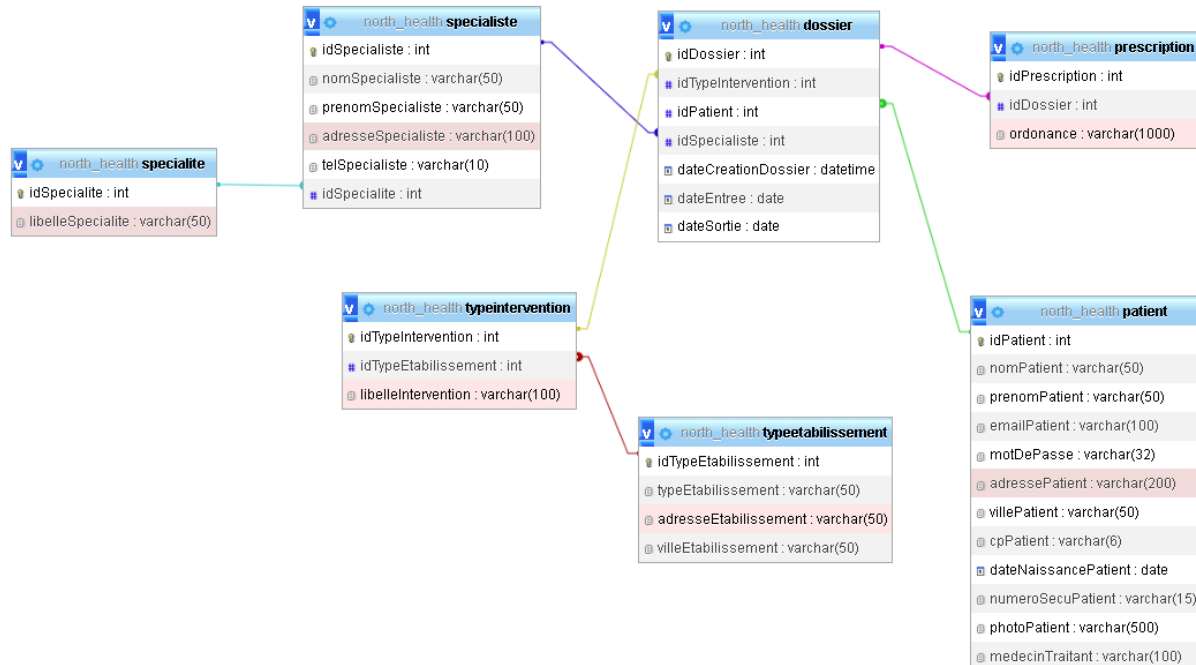
Ce dossier contient les entités
Ce dossier contient le
gestionnaire de la session
Ce dossier contient les
requêtes SQL vers la base de
données



Nom	Statut	Modifié le	Type	Taille
bin	✓	15/04/2023 15:12	Dossier de fichiers	
data	✓	15/04/2023 15:12	Dossier de fichiers	
Entities	✓	15/04/2023 15:12	Dossier de fichiers	
Helper	✓	15/04/2023 15:12	Dossier de fichiers	
obj	✓	15/04/2023 15:12	Dossier de fichiers	
QueryDesigner	✓	15/04/2023 15:12	Dossier de fichiers	
src	✓	15/04/2023 15:01	Dossier de fichiers	
AccueilForm	✓	15/04/2023 15:12	Fichier source C#	2 Ko
AccueilForm.Designer	✓	15/04/2023 15:12	Fichier source C#	6 Ko
AccueilForm.resx	✓	15/04/2023 15:12	Fichier RESX	3 Ko
AccueilFormForm	✓	15/04/2023 15:12	Fichier source C#	3 Ko
AccueilFormForm.Designer	✓	15/04/2023 15:12	Fichier source C#	4 Ko
AccueilFormForm.resx	✓	15/04/2023 15:12	Fichier RESX	3 Ko
CompteRenduForm	✓	15/04/2023 15:12	Fichier source C#	3 Ko
CompteRenduForm.Designer	✓	15/04/2023 15:12	Fichier source C#	4 Ko
CompteRenduForm.resx	✓	15/04/2023 15:12	Fichier RESX	3 Ko
FormLogin	✓	15/04/2023 15:12	Fichier source C#	2 Ko
FormLogin.Designer	✓	15/04/2023 15:12	Fichier source C#	5 Ko
FormLogin.resx	✓	15/04/2023 15:12	Fichier RESX	3 Ko
NorthHealthFormApp	✓	15/04/2023 15:12	Fichier source C#	1 Ko
NorthHealthFormApp.Designer	✓	15/04/2023 15:12	Fichier USER	1 Ko
NorthHealthFormApp.resx	✓	15/04/2023 15:12	Fichier SLN	2 Ko
PlanningForm	✓	15/04/2023 15:12	Fichier source C#	1 Ko
PlanningForm.Designer	✓	15/04/2023 15:12	Fichier source C#	3 Ko
PlanningForm.resx	✓	15/04/2023 15:12	Fichier RESX	3 Ko
Program	✓	15/04/2023 15:12	Fichier source C#	1 Ko
README	✓	15/04/2023 15:30	Fichier source Mar...	2 Ko

Compte rendu Développement en C# de la partie application lourde

3. Schéma de la base de données :



Dossier : Cette table contient les données relatives aux consultations des patients.

Patient : Cette table contient les informations relatives aux patients.

Spécialiste : Cette table contient les informations relatives aux spécialistes.

4. Modification code source

Compte rendu Développement en C# de la partie application lourde

Mission 1 : Un administratif est de se connecter et d'accéder au planning des interventions :

```
15 private void seConnecterButton_Click(object sender, EventArgs e)
16 {
17     try
18     {
19         // Je récupère les informations de connexion...
20         string emailLogin = loginBox.Text.Trim().ToString();
21         string pwd = pwdBox.Text.Trim().ToString();
22
23         // Il faut que je requête ma base de données pour savoir si l'utilisateur existe bien...
24         QDProvider qDProvider = new QDProvider();
25         Admin admin = qDProvider.LoginAdmin(emailLogin, pwd);
26
27         if (admin.IdAdmin != 0)
28         {
29             // TODO: Une fois l'admin est connecté il faut lui donner la possibilité de pouvoir consulter les dossiers médicaux...
30             // J'ouvre ma session...
31             SessionManager.openSession(admin);
32             AccueilForm accueilForm = new AccueilForm();
33             this.Hide();
34             accueilForm.ShowDialog();
35         }
36         else
37         {
38             MessageBox.Show("Erreur de connexion ! Veuillez vérifier vos infos.", "Echec de connexion");
39         }
40     }
41     catch (Exception ex) {
42         MessageBox.Show(ex.Message, "error");
43     }
44 }
```

```
public Admin LoginAdmin(string login, string pwd)
{
    string query = "SELECT * FROM admin WHERE login = @login AND pwd=@pwd";

    Admin loggedAdmin = new Admin();

    MySqlCommand mySqlCommand = new MySqlCommand(query, OpenConnection());
    mySqlCommand.Parameters.AddWithValue("@login", login);
    mySqlCommand.Parameters.AddWithValue("@pwd", pwd);

    using (MySqlDataReader reader = mySqlCommand.ExecuteReader())
    {
        while (reader.Read())
        {
            loggedAdmin.IdAdmin = (int)reader["idAdmin"];
            loggedAdmin.NomAdmin = (string)reader["nomAdmin"];
            loggedAdmin.Login = (string)reader["login"];
            loggedAdmin.Password = (string)reader["pwd"];
        }
    }

    return loggedAdmin;
}
```

Compte rendu Développement en C# de la partie application lourde

```
public List<Dossier> getPlanningPatient()
{
    string query = "SELECT d.idDossier, CONCAT(te.adresseEtablissement , " +
        "\"\", \"\", c.nomCommune ,\" \"\", te.codePostalEtablissement) AS \"Adresse\", \" +
        \" d.libelleIntervention, te.typeEtablissement, CONCAT(s.nomSpecialiste,\" \"\", \" +
        \" s.prenomSpecialiste) AS \"Medecin\", d.creneau, CONCAT(p.nomPatient, \" \"\", p.prenomPatient) AS \"Patient\"\"\"\" +
        "FROM dossier d\"\"\"INNER JOIN typeetablisement te\"\"\"ON d.idTypeEtablissement = te.idTypeEtablissement\"\"\" +
        "INNER JOIN specialiste s\"\"\"ON d.idSpecialiste = s.idSpecialiste\"\"\"INNER JOIN patient p\"\"\"ON p.idPatient = d.idPatient\"\"\" +
        "INNER JOIN commune c\"\"\"ON c.codePostal = d.codeCommune";

    MySqlCommand mySqlCommand = new MySqlCommand(query, OpenConnection());

    List<Dossier> dossiers = new List<Dossier>();
    using (MySqlDataReader reader = mySqlCommand.ExecuteReader())
    {
        while (reader.Read())
        {
            Dossier dossier = new Dossier();
            dossier.IdDossier = (int) reader["idDossier"];
            dossier.Adresse = (string) reader["Adresse"];
            dossier.Etablissement = (string) reader["typeEtablissement"];
            dossier.Patient = (string) reader["Patient"];
            dossier.Medecin = (string) reader["Medecin"];
            dossier.LibelleIntervention = (string) reader["libelleIntervention"];
            dossier.Horraire = (DateTime) reader["creneau"];

            dossiers.Add(dossier);
        }
    }

    return dossiers;
}
```

Mission 2 : Un personnel médical est capable d'ajouter des éléments (radio, scanner, rapport, ...) sur le patient.

```
public void addDocumentPath(int idDossier, string path)
{
    string query = "UPDATE dossier SET documents = @path WHERE idDossier = @idDossier";
    MySqlCommand mySqlCommand = new MySqlCommand(query, OpenConnection());
    mySqlCommand.Parameters.AddWithValue("@path", path);
    mySqlCommand.Parameters.AddWithValue("@idDossier", idDossier);

    mySqlCommand.ExecuteNonQuery();
}
```

Compte rendu Développement en C# de la partie application lourde

- **Mission 3** : Un administratif est capable de générer une facture .

```
namespace NorthHealthFormApp
{
    public partial class CompteRenduForm : Form
    {
        private QDProvider qdProvider;
        public CompteRenduForm()
        {
            InitializeComponent();
            qdProvider = new QDProvider();
        }

        private void CompteRenduForm_Load(object sender, EventArgs e)
        {
            // TODO: Il faut récupérer tous les users...
            List<Patient> patients = qdProvider.GetPatients();
            foreach (Patient p in patients)
            {
                userComboBox.Items.Add(String.Format("{0} {1} {2}", p.IdPatient, p.NomPatient, p.PrenomPatient));
            }
        }

        private void userSleected(object sender, EventArgs e)
        {
            // Récupérer la valeur sélectionnée...
            dossierComboBox.Items.Clear();

            int idPatient = int.Parse(userComboBox.SelectedItem.ToString().Split(' ')[0]);
            List<Dossier> dossiers = qdProvider.getDossierParId(idPatient);
            if (dossiers.Count > 0)
            {
                foreach (Dossier d in dossiers)
                {
                    dossierComboBox.Items.Add(String.Format("{0} {1} {2}", d.IdDossier, d.Horrire, d.LibelleIntervention));
                }
                dossierComboBox.Visible = true;
            }
            else
            {
                dossierComboBox.Visible = false;
                genererCompteRenduButton.Visible = false;

                string userName = userComboBox.SelectedItem.ToString().Split(' ')[1] + " " + userComboBox.SelectedItem.ToString().Split(' ')[2];
                MessageBox.Show("Aucun dossier n'a été trouvé pour le patient " + userName);
            }
        }
    }
}
```

```
private void GenerateInvoiceButton(object sender, EventArgs e)
{
    int idPatient = int.Parse(userComboBox.SelectedItem.ToString().Split(' ')[0]);
    int idDossier = int.Parse(dossierComboBox.SelectedItem.ToString().Split(' ')[0]);

    // J'appelle la fonction pour qu'elle me renvoie les données relatives au patient et au dossier sélectionnées...
    InvoicePDF invoicePDF = qdProvider.getInfoInvoice(idPatient, idDossier);
    Document pdf = PDFGenerator.GenerateInvoicePDF(invoicePDF);

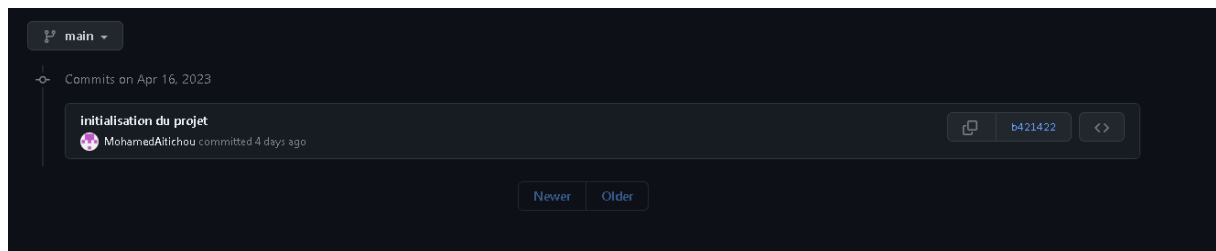
    MessageBox.Show("Le compte rendu a bien été généré ! Il se trouve ici: C:\north_health\compte_rendu\compte_rendu"+invoicePDF.IdDossier+".pdf", "Info");
}

private void ShowButton(object sender, EventArgs e)
{
    genererCompteRenduButton.Visible = true;
}
}
```

Compte rendu Développement en C# de la partie application lourde

5. Gestion de version

Git :



Github :

<https://github.com/MohamedAitichou/North-HEALTH-Windows-forms->

6. Compétences

A4.1.1 , Proposition d'une solution applicative.

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

A4.1.3 , Conception ou adaptation d'une base de données.

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.

A4.1.9 , Rédaction d'une documentation technique.

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

A4.2.2 , Adaptation d'une solution applicative aux évolutions de ses composants.

A4.2.4 , Mise à jour d'une documentation technique.