Modifications Apportées à VOTRE Travail

📋 Résumé

IMPORTANT : Je n'ai **AUCUNE modification** apportée au travail de votre ami. Ses fichiers (`client.cpp` et `client.h`) ont été **copiés tels quels** sans aucune modification.

Toutes les modifications ci-dessous ont été faites dans **VOTRE projet** pour intégrer le module CLIENT.

☑ Modifications dans VOS Fichiers

1. **moemen.pro** - Fichier de projet Qt

Modifications :

```
# AVANT :
SOURCES += \
    connection.cpp \
    login.cpp \
    main.cpp \
    mainwindow.cpp \
    seance.cpp \
    vehicule.cpp \
    circuit.cpp

HEADERS += \
    connection.h \
    login.h \
    mainwindow.h \
    seance.h \
    vehicule.h \
    circuit.h

# APRÈS :
SOURCES += \
    connection.cpp \
    login.cpp \
    main.cpp \
    mainwindow.cpp \
    seance.cpp \
    vehicule.cpp \
    circuit.cpp \
    client.cpp          # ← AJOUTÉ

HEADERS += \
    connection.h \
    login.h \
    mainwindow.h \
```

```
    seance.h \
    vehicule.h \
    circuit.h \
    client.h            # ← AJOUTÉ
```

Raison : Pour que Qt compile les nouveaux fichiers `client.cpp` et `client.h`.

---

2. **mainwindow.h** - Header de la fenêtre principale

Modifications :

a) Includes ajoutés :

```
// AVANT :
#include <QMainWindow>
#include "seance.h"
#include "vehicule.h"
#include "circuit.h"

// APRÈS :
#include <QMainWindow>
#include <QTableWidgetItem>    // ← AJOUTÉ (pour les widgets CLIENT)
#include "seance.h"
#include "vehicule.h"
#include "circuit.h"
#include "client.h"            // ← AJOUTÉ
```

b) Slots ajoutés dans la section `private slots:` :

```
// AJOUTÉS :
// Slots pour les opérations CRUD des clients
void on_btnAjouter_clicked();
void on_btnModifier_clicked();
void on_btnSupprimer_clicked();
void on_btnChercher_clicked();
void on_tableWidget_6_itemClicked(QTableWidgetItem *item);
void on_lineEditRecherche_textChanged(const QString &arg1);
void on_comboBoxTrier_currentIndexChanged(int index);
```

c) Instance de classe ajoutée :

```
// AVANT :
private:
    Ui::MainWindow *ui;
    Seance currentSeance;
```

```
    Vehicule V; // Instance de Vehicule
    Circuit C; // Instance de Circuit

// APRÈS :
private:
    Ui::MainWindow *ui;
    Seance currentSeance;
    Vehicule V; // Instance de Vehicule
    Circuit C; // Instance de Circuit
    client C_Client; // Instance de Client  // ← AJOUTÉ
```

Note importante : J'ai utilisé `C_Client` au lieu de `C` pour éviter le conflit avec `Circuit C`.

d) Méthodes utilitaires ajoutées :

```
// AJOUTÉS :
void refreshTableClient();
void clearFieldsClient();
```

Raison : Pour intégrer le module CLIENT dans votre interface existante.

---

3. **mainwindow.cpp** - Implémentation de la fenêtre principale

Modifications :

a) Includes ajoutés :

```
// AJOUTÉS :
#include <QTableWidgetItem>
#include <QRegularExpression>
#include <QComboBox>
#include <QVariant>
#include "client.h"
```

b) Dans le constructeur `MainWindow::MainWindow()` :

```
// AVANT :
        refreshTableSeance();
        refreshTableVehicule();

        QMessageBox::information(this, "Succès", "Base de données initialisée
avec succès!");

// APRÈS :
        refreshTableSeance();
```

```
        refreshTableVehicule();
        refreshTableClient();  // ← AJOUTÉ

        QMessageBox::information(this, "Succès", "Base de données initialisée
avec succès!");
```

c) Fonction `clientclick()` modifiée :

```
// AVANT :
void MainWindow::clientclick() {
    ui->stack->setCurrentIndex(2);
}

// APRÈS :
void MainWindow::clientclick() {
    ui->stack->setCurrentIndex(2);
    refreshTableClient();  // ← AJOUTÉ
    // Initialiser le comboBox de tri à ID par défaut
    if (ui->comboBoxTrier) {  // ← AJOUTÉ
        ui->comboBoxTrier->setCurrentIndex(0);  // ← AJOUTÉ
    }
}
```

d) Nouvelles fonctions ajoutées (à la fin du fichier) :

Toutes ces fonctions ont été **ajoutées** (elles n'existaient pas avant) :

1. `void MainWindow::refreshTableClient()`
    - Affiche les clients dans le tableau `tableWidget_6`
    - Convertit les données de `QSqlQueryModel` vers `QTableWidget`
    - Formate les dates
2. `void MainWindow::clearFieldsClient()`
    - Vide tous les champs du formulaire CLIENT
    - Vérifie l'existence des widgets avant de les utiliser
3. `void MainWindow::on_btnAjouter_clicked()`
    - Ajoute un nouveau client
    - Valide les données (CIN, email, téléphone, etc.)
    - Vérifie l'unicité du CIN
4. `void MainWindow::on_btnModifier_clicked()`
    - Modifie un client existant
    - Valide les données
    - Vérifie l'unicité du CIN (en excluant le client actuel)

5. `void MainWindow::on_btnSupprimer_clicked()`
    - Supprime un client
    - Demande confirmation avant suppression
    - Vérifie l'existence du client
6. `void MainWindow::on_btnChercher_clicked()`
    - Recherche des clients par nom
    - Affiche les résultats dans le tableau
7. `void MainWindow::on_tableWidget_6_itemClicked(QTableWidgetItem *item)`
    - Remplit le formulaire quand on clique sur une ligne du tableau
    - Parse les dates dans différents formats Oracle
8. `void MainWindow::on_lineEditRecherche_textChanged(const QString &arg1)`
    - Recherche en temps réel quand le champ de recherche change
9. `void MainWindow::on_comboBoxTrier_currentIndexChanged(int index)`
    - Trie les clients selon le critère sélectionné (ID ou Nom)

Raison : Pour rendre le module CLIENT fonctionnel dans votre interface.

📝 Fichiers AJOUTÉS (pas modifiés)

Ces fichiers ont été **copiés** depuis le travail de votre ami, **sans aucune modification** :

1. `client.cpp` - Implémentation de la classe client
2. `client.h` - Déclaration de la classe client

🔍 Fichiers NON Modifiés

Ces fichiers de votre projet ont été **conservés intacts** :

- ☑ `circuit.cpp` - Votre version avec les corrections (TO_CHAR)
- ☑ `circuit.h` - Intact
- ☑ `seance.cpp` - Votre version conservée
- ☑ `seance.h` - Intact
- ☑ `vehicule.cpp` - Intact
- ☑ `vehicule.h` - Intact
- ☑ `connection.cpp` - Intact
- ☑ `connection.h` - Intact
- ☑ `mainwindow.ui` - Intact (mais vous devrez peut-être ajouter les widgets CLIENT)

⚠ Points Importants

1. Conflit de noms résolu

- Votre projet avait : `Circuit C`
- Le projet de votre ami avait : `client C`
- **Solution** : J'ai renommé l'instance en `C_Client` pour éviter le conflit

2. Vérifications de sécurité ajoutées

Toutes les fonctions CLIENT vérifient l'existence des widgets UI avant de les utiliser :

```
if (!ui->tableWidget_6) {
    qDebug() << "⚠ tableWidget_6 n'existe pas dans l'UI";
    return;
}
```

Cela évite les crashes si les widgets n'existent pas encore dans `mainwindow.ui`.

3. Compatibilité préservée

- Toutes vos fonctionnalités CIRCUIT sont **intactes**
- Toutes vos fonctionnalités SEANCE sont **intactes**
- Toutes vos fonctionnalités VEHICULE sont **intactes**
- Le module CLIENT a été **ajouté** sans casser l'existant

📊 Résumé des Modifications

Fichier	Type de Modification	Lignes Ajoutées	Lignes Modifiées
`moemen.pro`	Ajout de fichiers	2 lignes	0
`mainwindow.h`	Ajout includes/slots	~15 lignes	0
`mainwindow.cpp`	Ajout fonctions	~400 lignes	3 lignes
`client.cpp`	Fichier copié	302 lignes	0
`client.h`	Fichier copié	61 lignes	0

Total : ~780 lignes ajoutées, 3 lignes modifiées

☑ Ce qui reste à faire

1. **Vérifier** `mainwindow.ui` :
    - S'assurer que tous les widgets CLIENT existent :
        - `tableWidget_6`
        - `lineEditID`, `lineEditNom`, `lineEditPrenom`, `lineEditCIN`
        - `jjj` (QDateEdit pour la date)

- `lineEditMotDePasse` (pour l'adresse)
- `lineEditTelephone`
- `lineEditPoste` (pour l'email)
- `lineEditRecherche`
- `comboBoxTrier`
- `btnAjouter`, `btnModifier`, `btnSupprimer`, `btnChercher`
- `lineEditSupprimer`

2. **Compiler et tester** :

```
qmake moemen.pro
make  # ou mingw32-make
```

3. **Si des erreurs de compilation** :
   - Vérifier que tous les widgets existent dans `mainwindow.ui`
   - Vérifier les noms des widgets (ils doivent correspondre exactement)

Date : 2025-01-07
Statut : Intégration terminée, prêt pour compilation