APPLICTATION DE GESTION DES ETUDIANTS

Rapport

Remerciements

+ Nous remercions nos professeurs **M. Amine Benamrane** et **M. Saif Bouderba** pour leur aide, leurs conseils et leur encadrement tout au long de ce projet.
+ Ce travail a été réalisé par **Aymane El Melouly** et **Radouan Kartite** dans le cadre du module de Programmation Orientée Objet et Base de Données, à la Faculté Polydisciplinaire de Safi (FPS), Université Cadi Ayyad.
+ Nous exprimons également notre gratitude à l'ensemble du corps enseignant
pour les connaissances et les outils que nous avons pu mettre en œuvre dans cette application.

Introduction

Dans le cadre de notre formation à la Faculté Polydisciplinaire de Safi, nous avons réalisé un projet informatique visant à développer une application de gestion des étudiants. Ce projet nous a permis d'appliquer des notions vues en programmation orientée objet, en gestion de base de données et en conception d'interfaces graphiques.
L'objectif principal est de proposer une solution simple, fonctionnelle et utile à l'administration universitaire pour faciliter le suivi des étudiants et de leurs notes.

<< Partie Conception >>

Problématique !!

À la Faculté Polydisciplinaire de Safi, la gestion manuelle des informations
des étudiants, des matières et des notes devient de plus en plus difficile à mesure que leur nombre augmente. Ce mode de fonctionnement peut entraîner des erreurs, des lenteurs et une surcharge de travail pour l'administration.
Il est donc nécessaire de mettre en place une application permettant de centraliser ces données et d'automatiser leur traitement, afin d'améliorer l'efficacité et la fiabilité du système.

Quel est le besoin traité par le projet ?

Le projet vise à répondre au besoin de digitalisation et de simplification de la gestion académique. Plus précisément, il permet de :

- Enregistrer et gérer les informations des étudiants
- Associer les étudiants aux matières
- Enregistrer et modifier les notes
- Consulter, trier et filtrer les données selon différents critères

Cette application contribuera à alléger le travail de l'administration et à garantir un meilleur suivi pédagogique.

Le Modèle Fonctionnel :

Le modèle fonctionnel définit la structure de base de notre application et les principales règles de gestion des données.
Notre projet vise à gérer les informations liées aux étudiants, aux matières et aux notes, à travers une interface simple.

Les principales parties de l'application sont :

- Gestion des étudiants : ajout, modification, suppression, consultation
- Gestion des matières : association matière ↔ étudiant
- Gestion des notes : saisie, modification, consultation
- Accès administrateur : seul l'administrateur peut gérer les données.

Ce modèle assure une organisation claire et facilite le développement des différentes fonctionnalités.

Les Règles de Gestion

Les données qu'on aura à stocker dans la base de données seront définies selon les règles suivantes :

Un Étudiant est défini par :

- Nom_étud
- Prénom_étud
- CNE_étud
- Massar_étud
- Date_naissance_étud

Une Note est définie par :

- Nom_étud
- Prénom_étud
- Massar_étud
- Module
- Note

Une Consultation est définie par :

- Massar_étud
- Date_consultation

Une Matière est définie par :

- Nom_professeur
- Nombre_heures

Modèle Logique de Données (MLD)

- **Étudiant** (CNE_étud, Nom_étud, Prénom_étud, Massar_étud, Date_naissance_étud)
- **Note** (CNE_étud, Nom_étud, Prénom_étud, Massar_étud, Module, Note)
- **Consultation** (Massar_étud, Date_consultation)
- **Matière** (Nom_prof, Nombre_heures)

Les différentes fonctionnalités du projet

L'application que nous souhaitons réaliser aura plusieurs méthodes pour gérer les données stockées dans la base de données. Nous définirons donc les méthodes principales que nous développerons pour chaque entité.

Pour toutes les entités, nous aurons quatre méthodes principales :

- Ajouter un élément
- Modifier un élément
- Supprimer un élément
- Afficher la liste de tous les éléments

<< Partie Technique>>

Les Outils Utilisés

Pour ce projet, nous avons choisi d'utiliser la programmation orientée objet avec le langage C++.

Pour créer une interface graphique conviviale, nous utilisons le framework Qt.

Qt est une bibliothèque en C++ qui facilite la création d'interfaces graphiques, la gestion des données, et la communication réseau.

Elle est compatible avec plusieurs systèmes d'exploitation comme Linux, Windows, et Mac OS, ce qui permet de porter facilement l'application sur différentes plateformes.

Qt est aussi utilisé par des environnements connus comme KDE sous Linux, ce qui montre sa fiabilité et sa popularité.

Pour le système de gestion de base de données (SGBD) on a utilisé SQLite pour la plus simple raison
de son adaptation et sa simplicité utilitaire.

- La relation entre C++ et le SGBD

Tout d'abord on introduit la bibliothèque concernée

```
#include <QSqlDatabase>
#include <QSqlDriver>
#include <QSqlError>
#include <QSqlQuery>
#include <QDebug>
#include <QSqlTableModel>
#include <QFile>
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

database.h

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

QSqlDatabase database = QSqlDatabase::addDatabase("QSQLITE");
    database.setDatabaseName("C:/Users/hp/OneDrive/Desktop/Documents/gestionEdatabase/mygestion.db");

    if(QFile::exists("C:/Users/hp/OneDrive/Desktop/Documents/gestionEdatabase/mygestion.db"))
    {
        qDebug()<<"Database file exists";
    }
    else
    {
        qDebug()<<"Database file doesnt't exists";
        return;
    }

    if(!database.open())
    {
        qDebug()<<"error: unable to open Database";
    }
    else
    {
        qDebug()<<"Database open successfully..";
```

Gestion de la base de données

Pour notre projet, on utilise **SQLite** car c'est simple et pratique.

Connexion entre C++ et la base de données

On utilise Qt pour connecter notre programme C++ à SQLite. On inclut les bibliothèques nécessaires dans le code.

Création de l'interface avec Qt-Creator

Pour créer les fenêtres, on choisit **Qt-Creator** car il permet de tout coder et avoir plus de contrôle.On crée la structure de la fenêtre, puis on ajoute les boutons et les zones de texte.

Envoi des commandes à la base de données

On envoie des commandes SQL grâce à Qt avec ces fonctions :

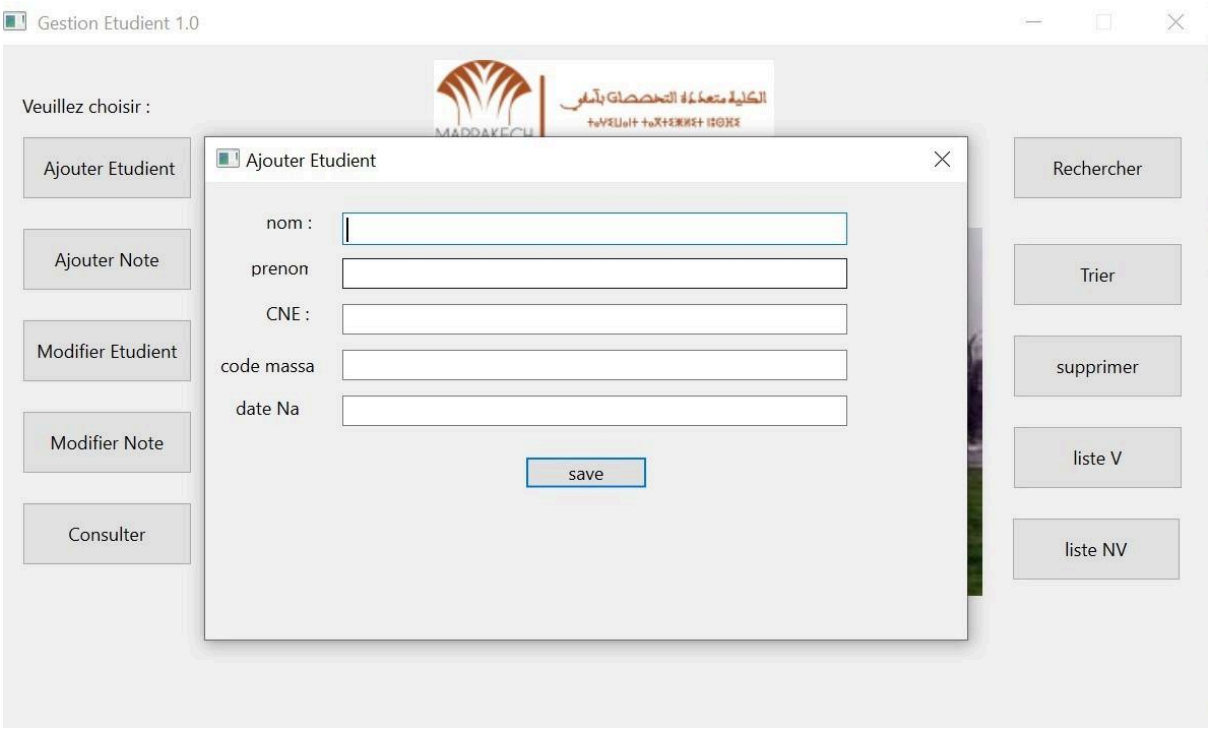
- `prepare()` pour préparer la commande
- `exec()` pour l'exécuter
- `next()` pour lire les résultats

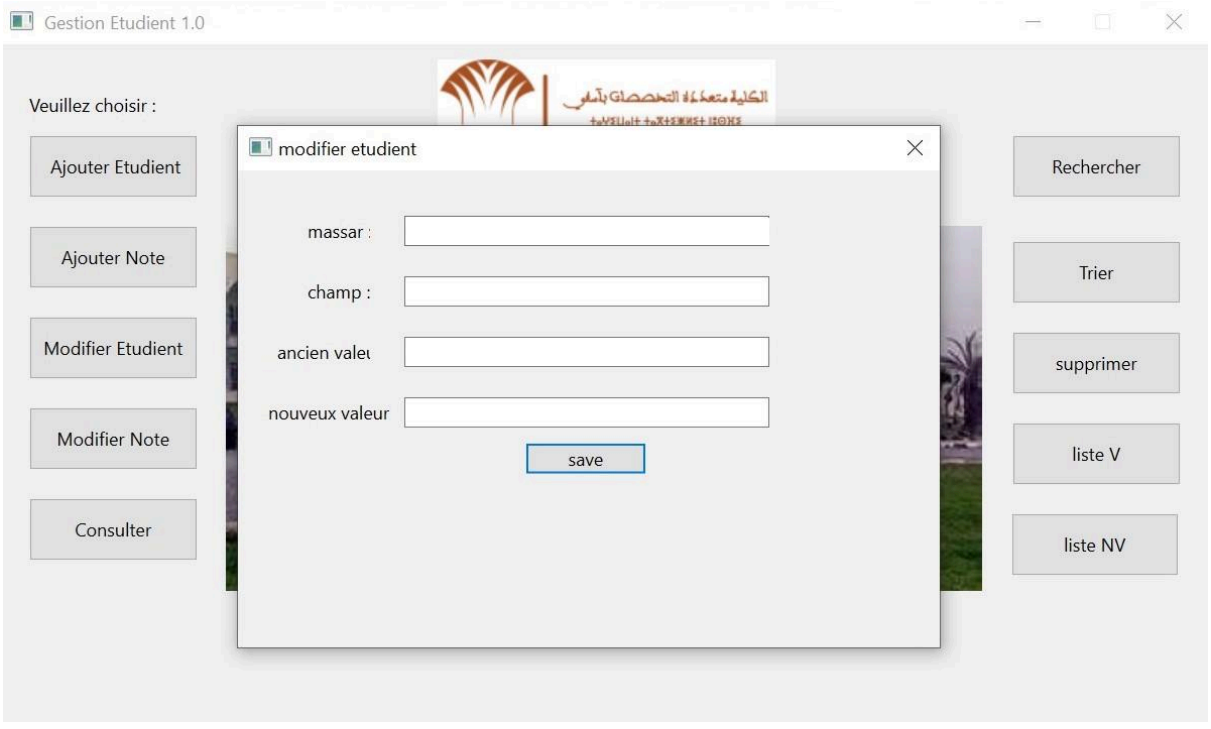
Quatre opérations principales :

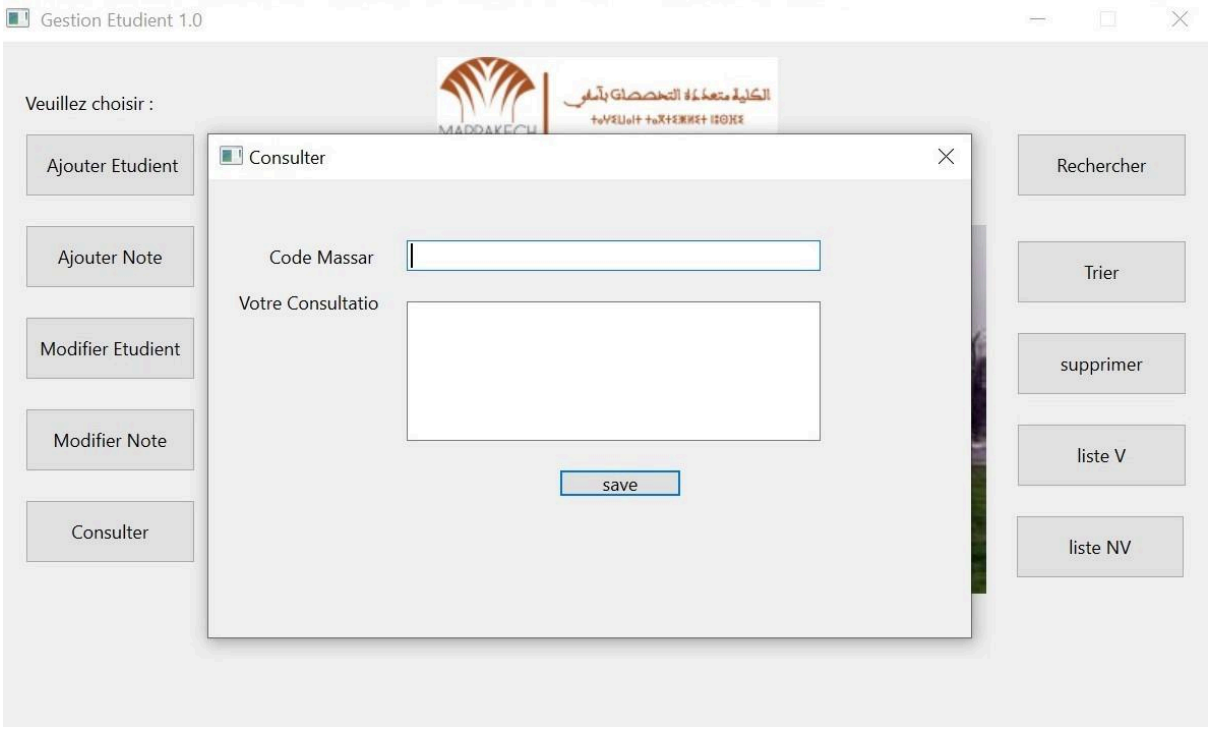
1. **Ajouter un élément** : on utilise `INSERT INTO` pour ajouter des données.
2. **Modifier un élément** : on utilise `UPDATE` pour changer des données.
3. **Supprimer un élément** : on utilise `DELETE` pour supprimer.
4. **Afficher les éléments** : on utilise `SELECT` pour voir les données, puis on les affiche dans un tableau.

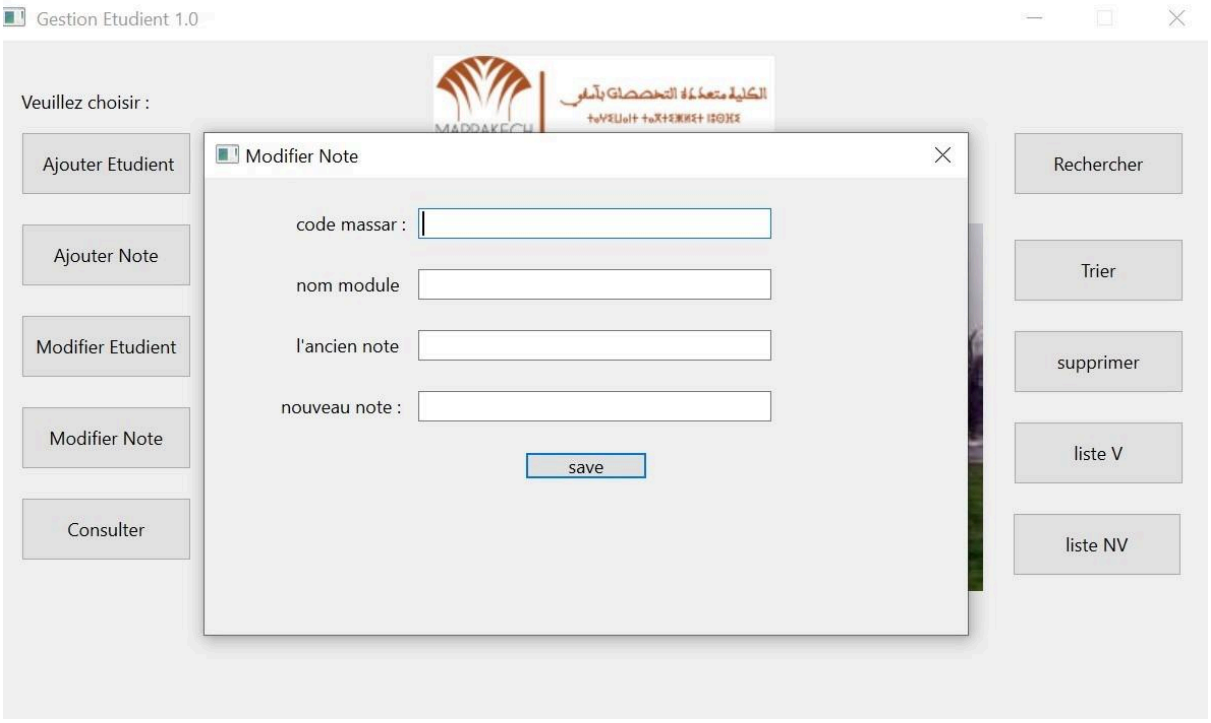
<<Le Résultat Final >>

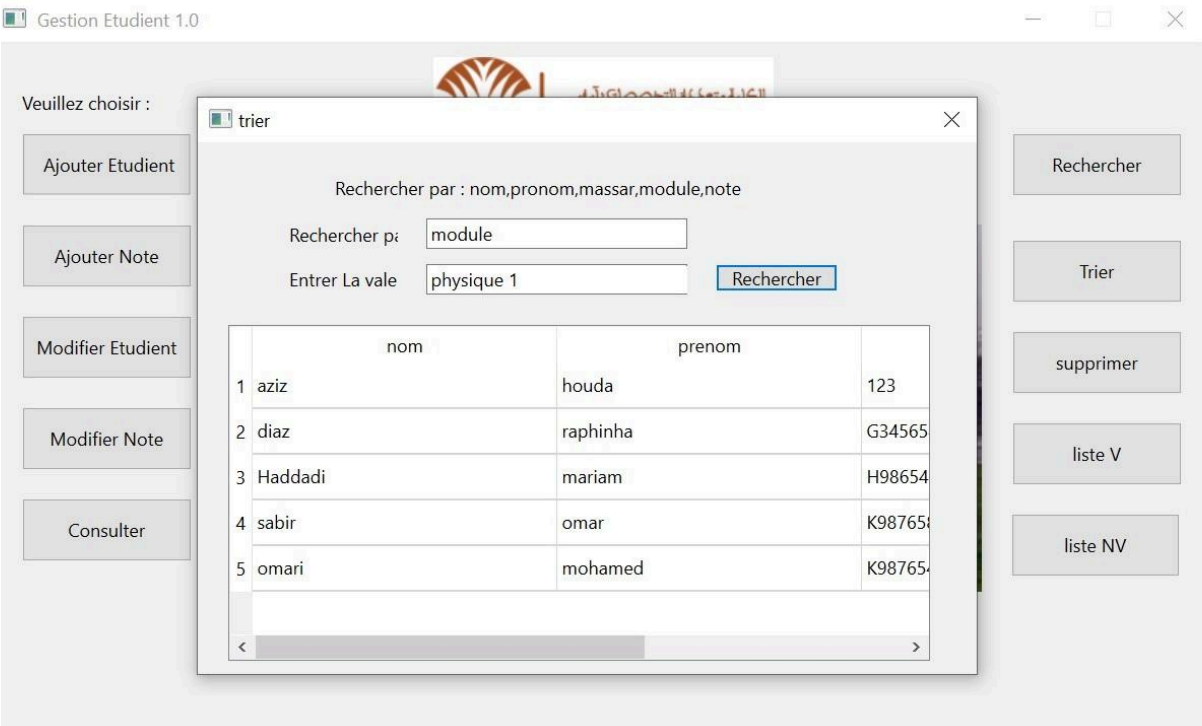
Presque la totalité du trajet est déjà effectué, il ne reste donc qu'à mettre pour chaque entité de la base de donnée les quatre fonctions qu'on a définie. Puis on obtient le
projet Final suivant :

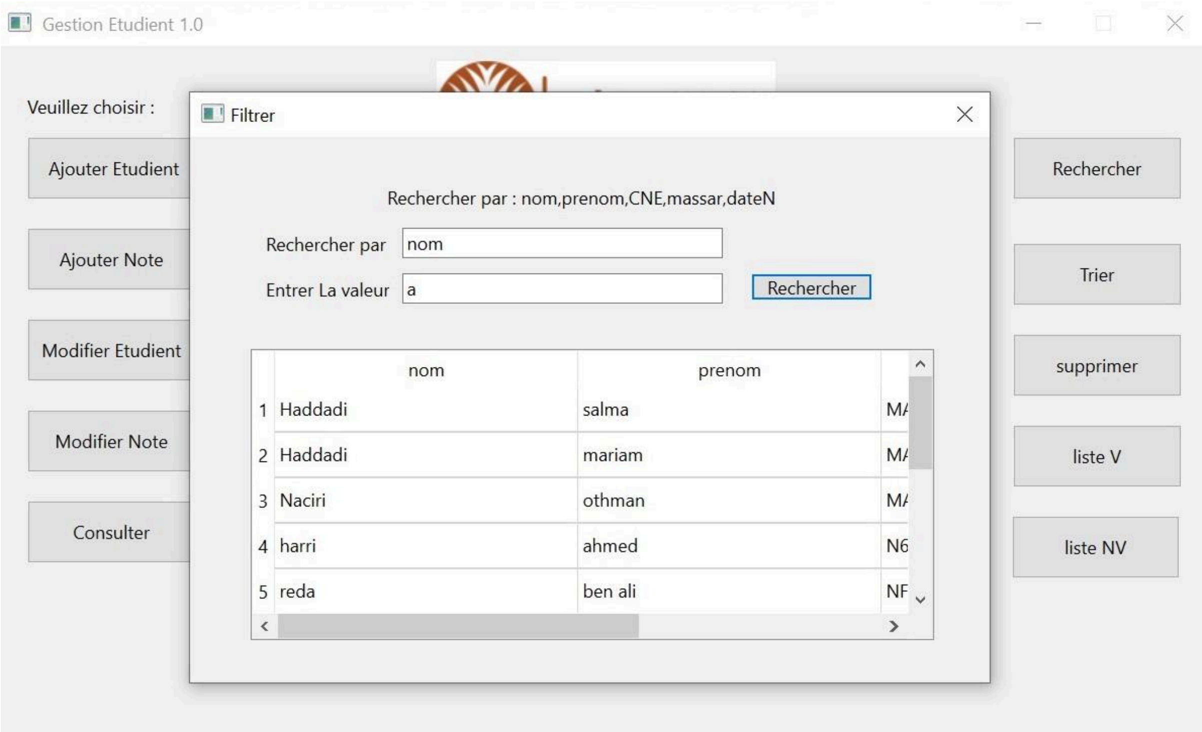

Gestion Etudient 1.0
Veuillez choisir :
Ajouter Etudient
Ajouter Note
Modifier Etudient
Modifier Note
Consulter
Rechercher
Trier
supprimer
liste V
liste NV
Ajouter Etudient
nom :
prenon
CNE :
code massa
date Na
save

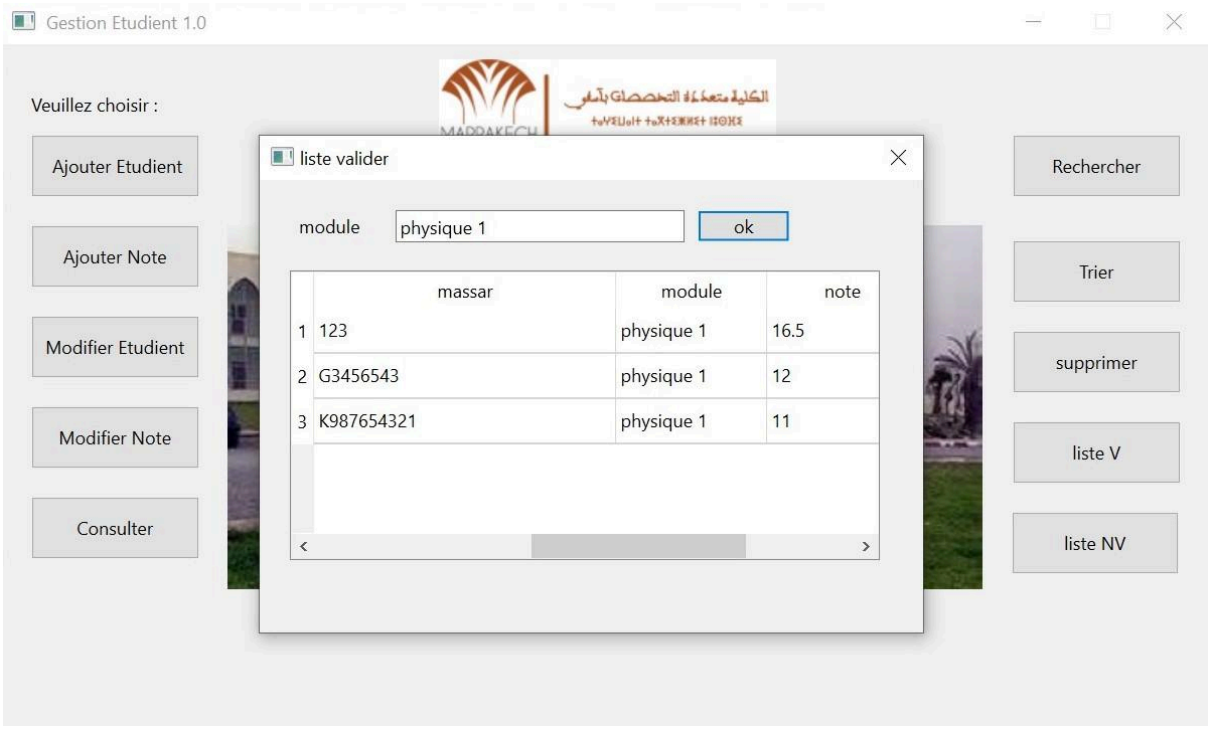

Gestion Etudient 1.0
Veuillez choisir :
Ajouter Etudient
Ajouter Note
Modifier Etudient
Modifier Note
Consulter
Rechercher
Trier
supprimer
liste V
liste NV
modifier etudient
massar :
champ :
ancien valeı
nouveux valeur
save

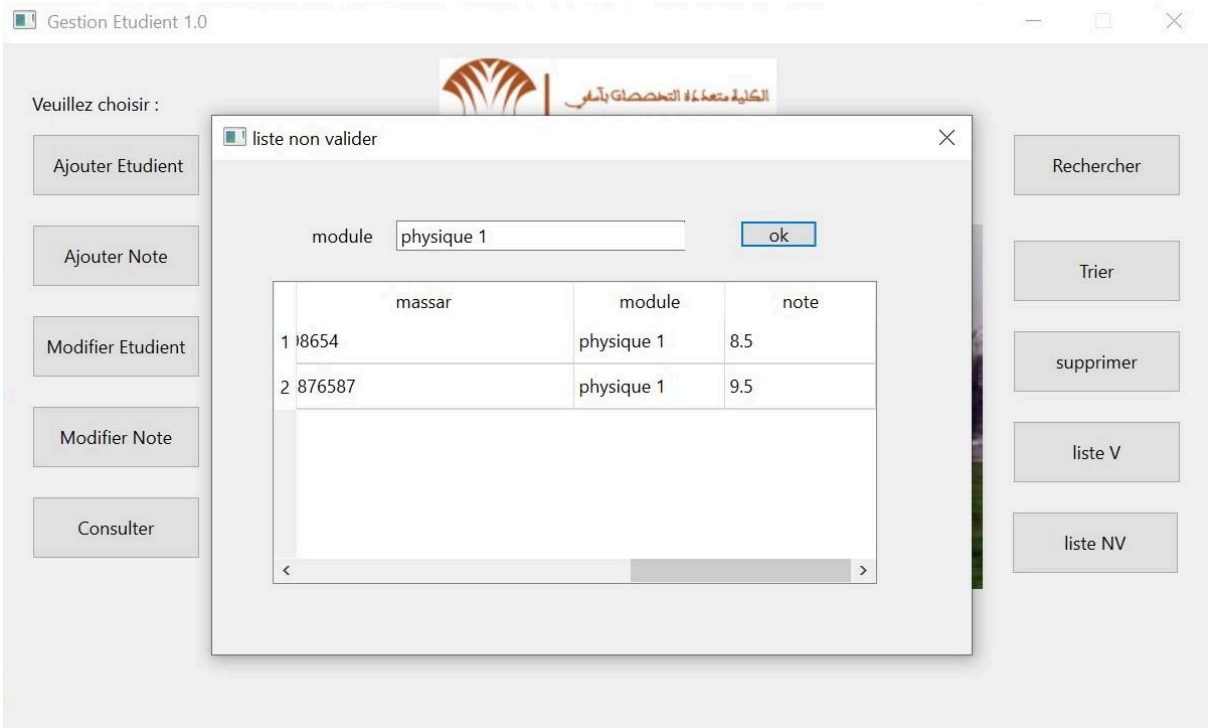

Gestion Etudient 1.0
Veuillez choisir :
Ajouter Etudient
Ajouter Note
Modifier Etudient
Modifier Note
Consulter
Rechercher
Trier
supprimer
liste V
liste NV
Consulter
Code Massar
Votre Consultatio
save

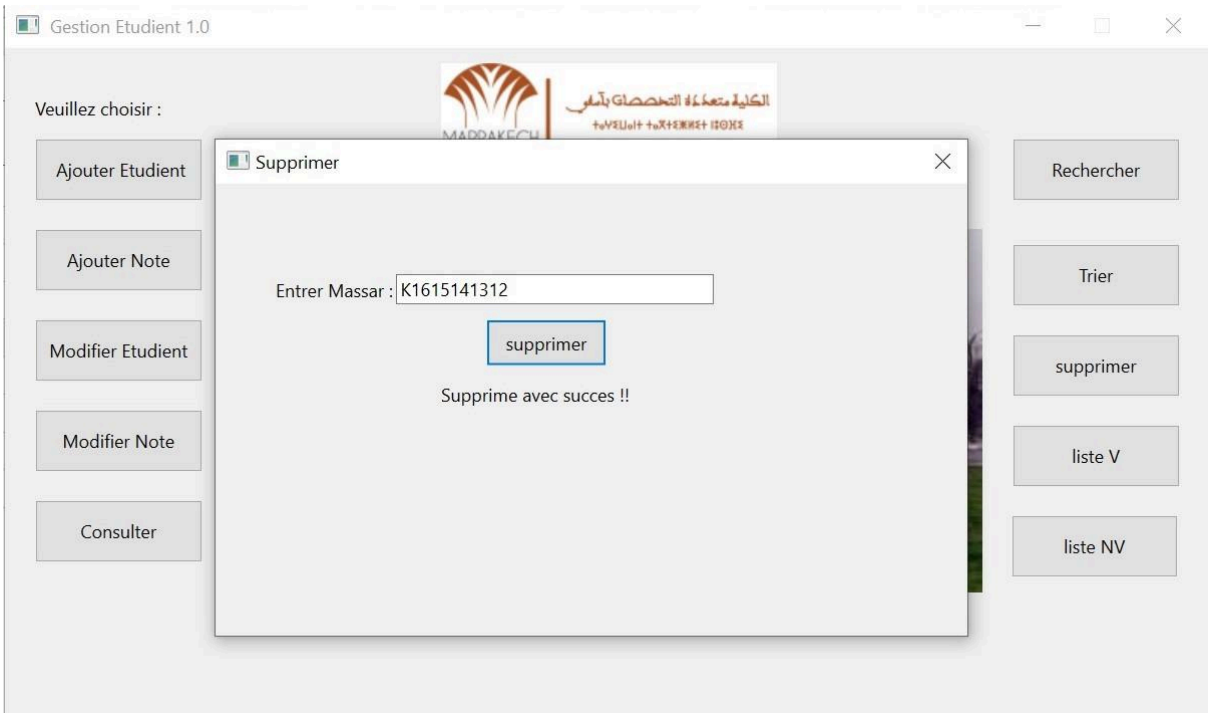

Gestion Etudient 1.0
Veuillez choisir :
Ajouter Etudient
Ajouter Note
Modifier Etudient
Modifier Note
Consulter
Rechercher
Trier
supprimer
liste V
liste NV
Modifier Note
code massar :
nom module
l'ancien note
nouveau note :
save


Gestion Etudient 1.0
Veuillez choisir :
Ajouter Etudient
Ajouter Note
Modifier Etudient
Modifier Note
Consulter
trier
Rechercher par : nom,pronom,massar,module,note
Rechercher pa
module
Entrer La vale
physique 1
Rechercher
nom
prenom
1 aziz houda 123
2 diaz raphinha G34565
3 Haddadi mariam H98654
4 sabir omar
5 omari mohamed
Rechercher
Trier
supprimer
liste V
liste NV


Gestion Etudient 1.0
Veuillez choisir :
Ajouter Etudient
Ajouter Note
Modifier Etudient
Modifier Note
Consulter
Filtrer
Rechercher par : nom,prenom,CNE,massar,dateN
Rechercher par
nom
Entrer La valeur
a
Rechercher
nom
prenom
1 Haddadi salma
2 Haddadi mariam
3 Naciri othman
4 harri ahmed
5 reda ben ali
Rechercher
Trier
supprimer
liste V
liste NV


Gestion Etudient 1.0
Veuillez choisir :
Ajouter Etudient
Ajouter Note
Modifier Etudient
Modifier Note
Consulter
liste valider
module
physique 1
ok
massar
module
note
1 123 physique 1 16.5
2 G3456543 physique 1 12
3 K987654321 physique 1 11
Rechercher
Trier
supprimer
liste V
liste NV


Gestion Etudient 1.0
Veuillez choisir :
Ajouter Etudient
Ajouter Note
Modifier Etudient
Modifier Note
Consulter
liste non valider
module
physique 1
ok
massar
module
note
1 8654 physique 1 8.5
2 876587 physique 1 9.5
Rechercher
Trier
supprimer
liste V
liste NV



<<Conclusion>>

Le projet d'application de gestion des étudiants avait pour but de créer un programme permettant de gérer les informations des étudiants, leurs notes, les consultations et les matières.

Pour cela, nous avons utilisé le langage **C++** avec la programmation orientée objet, ainsi que la base de données **SQLite** pour stocker et gérer les données.

Ce projet nous a permis de mieux comprendre la liaison entre C++ et une base de données, ainsi que la création d'une interface graphique avec **Qt**.

Pendant le développement, nous avons rencontré des difficultés, notamment pour connecter la base de données avec l'interface, mais cela nous a permis d'apprendre beaucoup.Malgré le travail accompli, le projet peut être amélioré et étendu.

Quelques idées pour l'avenir :

- Ajouter une fonctionnalité pour générer automatiquement des bulletins de notes.
- Étendre l'application pour gérer aussi les enseignants et les emplois du temps.

sommaire :