|  |
| --- |
| Université du Québec à Rimouski |
| Travail Pratique I |
| 8INF957 - Programmation objet avancée  Professeur – Yacine Yaddaden, Ph. D.  **Application pour Médecin et diagnostique par IA** |

|  |
| --- |
| **⭢ Mamadou Mouslim Diallo**  **⭢ Benjamin Lapointe-Pinel**  **⭢ FateMeh Bashardoustjoubjarkouli** |

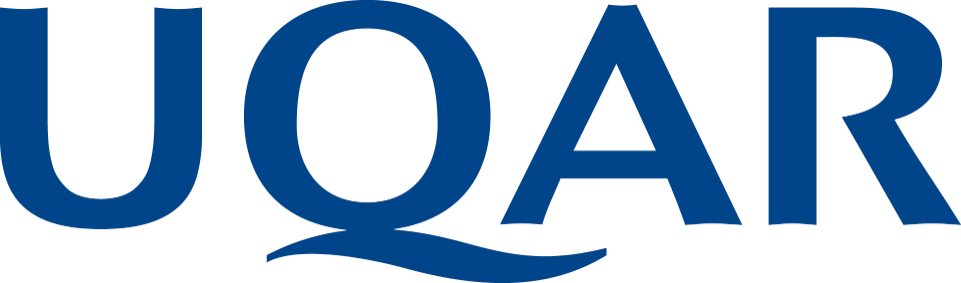


Table des matières

[Introduction 2](#_Toc96175531)

[Modélisation UML 2](#_Toc96175532)

[Aperçu de l’application (GUI) 2](#_Toc96175533)

[Algorithme du -NN et Résultats 2](#_Toc96175534)

[Problèmes et Difficultés rencontrées 2](#_Toc96175535)

[Conclusion 2](#_Toc96175536)

[Références 2](#_Toc96175537)

# Introduction

Pour la réalisation de notre application de bureau destinée aux médecins conformément à l’énoncé, nous avons utilisés plusieurs méthodes apprises en cours notamment la technique de programmation orientée objet dans la mise en place de notre code, la conception des interfaces graphiques avec l’outil WPF combiné à l’algorithmique KNN.

Les technologies utilisées sont Visual Studio et la librairie externe CSVHelper qui nous avons installé dans les Tools avec le gestionnaire Nuget pour l’upload des fichiers.

# Modélisation UML

Dans la modélisation UML, nous avons recenser les classes et entités qui entrent en jeu.

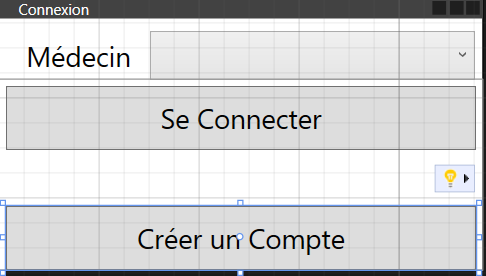
1. La classe Medecin
2. La classe Patient
3. La classe Diagnostic
4. La classe KNN
5. La classe IDiagnostic

# Aperçu de l’application (GUI)

Notre application destinée aux médecins dispose de plusieurs interfaces graphiques interconnectés conformément à notre énoncé.

***Interface de Connexion***

Nous avons créé la première interface graphique où nous avons la possibilité de se connecter à l’application ou création un compte pour le médecin.



Sur nos deux boutons, nous avons fait un binding à deux variables (Connect & CreateAccount) que nous avons créé dans le ViewModel.

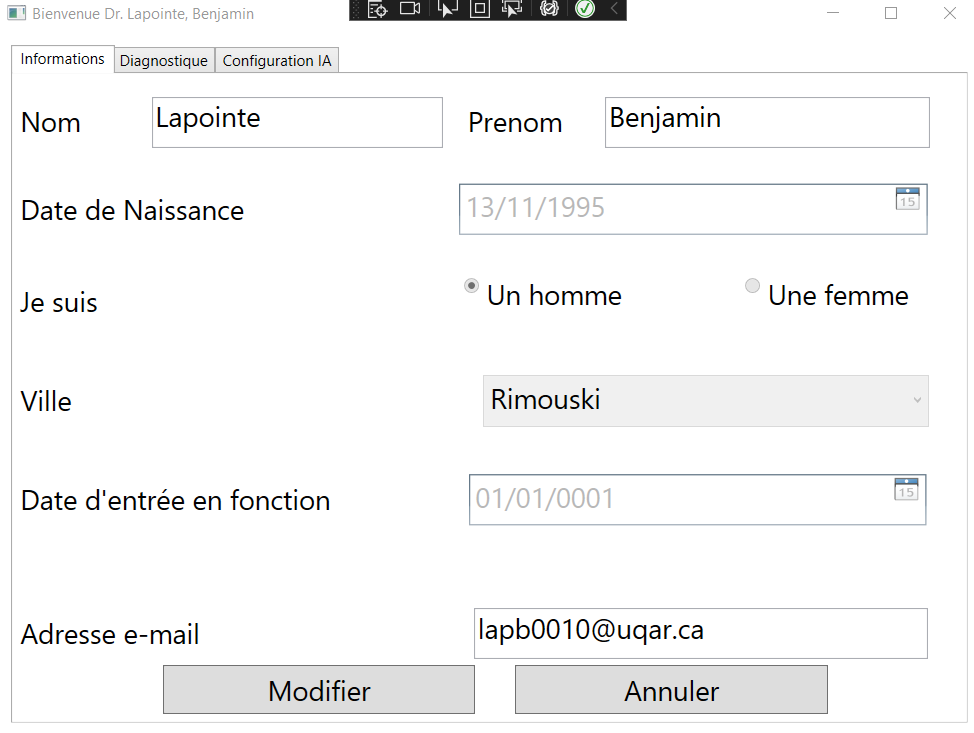
Le premier aspect consiste à créer un compte afin de remplir ses informations. Puis, vous aurez la possibilité d’ajouter le médecin puis se connecter.



Une fois connecté, vous allez sur l’interface Accueil.

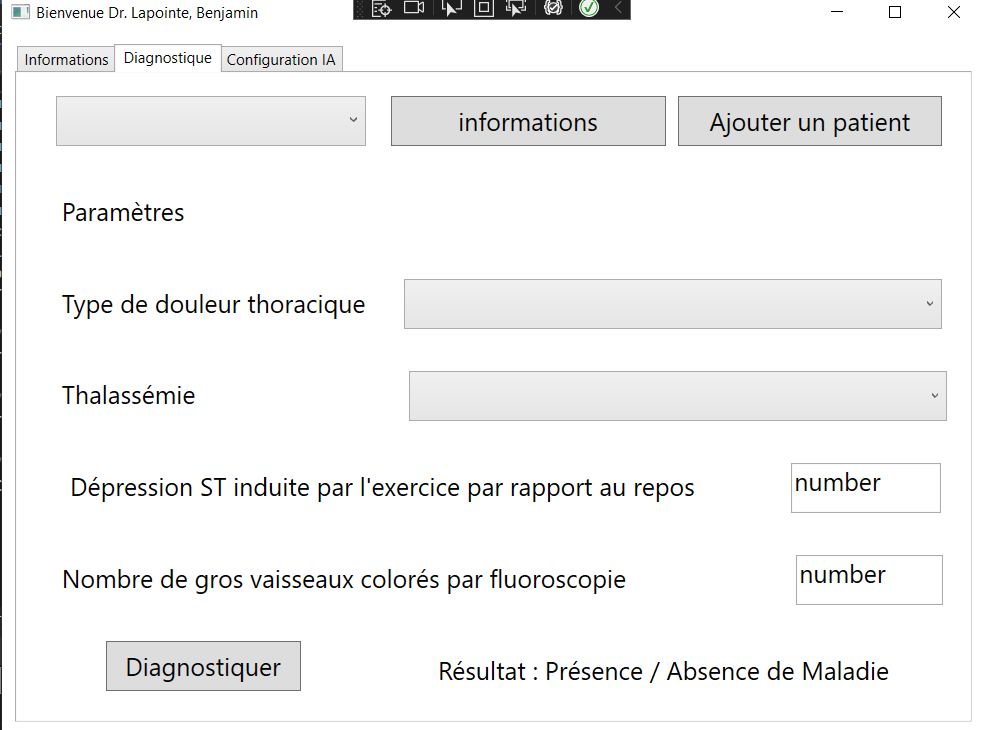
***L’interface Accueil.***

Elle représente l’interface de la bienvenue où nous avons trois 3 onglets : *informations, diagnostique, configuration IA.*

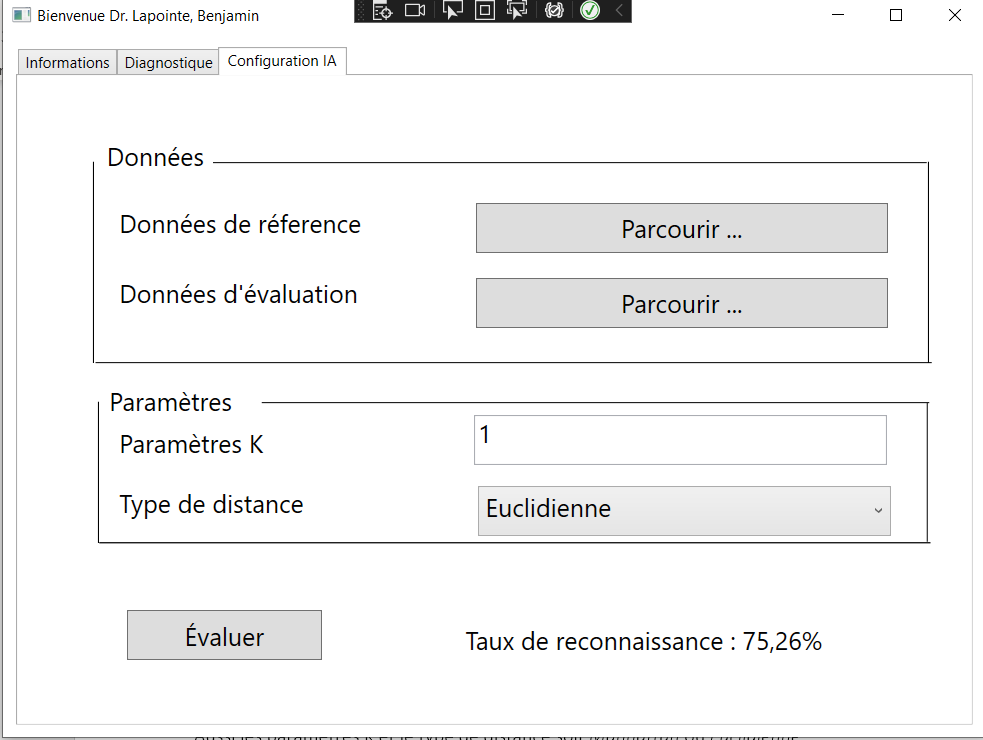
**

Sur l’onglet ***Informations*** nous avons les informations du médecin où nous avons la possibilité de modifier ou annuler.

Sur l’onglet ***Diagnostique*** où nous avons des boutons pour trouver les informations du médecin, ajouter un patient, définir le type de douleur thoracique, Thalassémie dans des Combobox. Définir aussi le nombre de dépression ST induite et le nombre de gros vaisseaux.



Sur l’onglet ***Configuration IA*** où nous avons deux boutons pour ajouter nos deux fichiers ***train.csv*** et ***test.csv.*** Il suffit de cliquer sur Parcourir et les charger depuis votre ordinateur. Aussi les paramètres K et le type de distance soit *Manhattan* ou *Euclidienne*.



A faire : sur l’onglet Configuration IA, une fois cliquer sur Evaluer, qu’est ce qui doit se passer ? ça donne le résultat du taux de reconnaissance en pourcentage.

A Faire : faire fonctionner le bouton *Informations, ajouter patient, Diagnostiquer* dans l’onglet **Diagnostique (Diagnostic classe et IDiagnostic)**.

*Interface Créer Patient*

# Algorithme du -NN et Résultats

L’implémentation de l’algorithme KNN connu sous le nom de K-Nearest Neighbors – les plus proches voisins – a connu plusieurs étapes.

KNN est un algorithme qui utilise une technique simple et a pour objectif est de faire des prédictions. Les données d'entrainement sont conservées, puis on localise les K enregistrements dedans les plus similaires, puis à partir de ses voisins on fait une prédiction.

Comment Calculer la distance euclidienne pour un échantillon donné ?

Comment Calculer le Manhattan ?

**Implémenter KNN en 3 étapes**

**Étape 1 : Calculer la distance euclidienne entre deux lignes**

**Étape 2 : Localiser les voisins les plus proches**

**Etape 3 - Faire des prédictions**

Nous utiliserons les voisins similaires collectés à partir de notre dataset d'apprentissage pour faire la prédiction.

**Etape 4 : Résultat final**

# Problèmes et Difficultés rencontrées

Dans la réalisation de notre projet (TP1), nous nous sommes confrontés à plusieurs problèmes, notamment

L’interconnexion des interfaces avec le binding

Le survol d’une fenêtre à une autre

Malgré les difficultés, les occupations, nous avons pu nous dépasser pour aboutir à ce résultat. La documentation sur internet en a largement contribué.

# Conclusion

Nous remercions M. Yacine pour ce TP1 qui nous fait largement découvrir le contour de ce langage ainsi l’outil WPF pour la conception des interfaces graphiques qui demandent beaucoup de soins et de concentration.

*A faire une vidéo de présentation*

# Références