

Projet L3 CDAA 2021 : Application de gestion de Contacts

Introduction

Dans ce projet il est question de réaliser une application QT capable de dialoguer avec une base de données SQLite afin de gérer des contacts. Chaque contact possède comme attributs un nom, un prénom, une entreprise, une adresse mail, un numéro de téléphone et une photo. On lui raccorde également une liste d'interactions qui stocke toutes les interactions du contact (y compris les modifications sur ce dernier), elle-même composée d'une liste de todo qui sont des tâches à réaliser par le contact.

Nous allons donc voir la réalisation de ce projet en trois parties. Dans un premier temps nous verrons la représentation C++ de l'application, les différents moyens mis en œuvre pour représenter les différentes fonctionnalités demandées. Ensuite nous verrons la représentation de l'interface graphique. Pour terminer nous verrons la façon dont nous avons réalisé la base de données.

Modélisation C++ de l'application

Les Classes :

GestionContact	Contact	Interaction	Todo	Date
Classe regroupant une liste de Contact ainsi que la date de dernière suppression d'un Contact	Classe représentant un contact, possédant tous les attributs cités plus haut	Classe possédant un contenu, une date et une liste de todo	Classe possédant un contenu (à faire) et une date	Classe représentant une date à l'aide de 3 entiers

Chaque classe a de nombreuses redéfinitions des opérateurs de base. Pour en citer quelques-uns les opérateurs $+$, $-$, $+=$ et $-=$ de *GestionContact* permettent l'ajout ou la suppression d'un *Contact*. L'opérateur $<<$ de *Contact* permet un affichage de tous ces attributs ainsi que de sa liste d'*Interaction* et de *Todo* triées par ordre croissant de date. L'opérateur $<$ de *Date* permet de savoir si une date est antérieure ou non à une autre.

Ci-dessous vous pouvez voir le diagramme de classe représentant les liens entre les différentes classes de la partie C++ de l'application :

Projet L3 CDAA 2021 : Application de gestion de Contacts

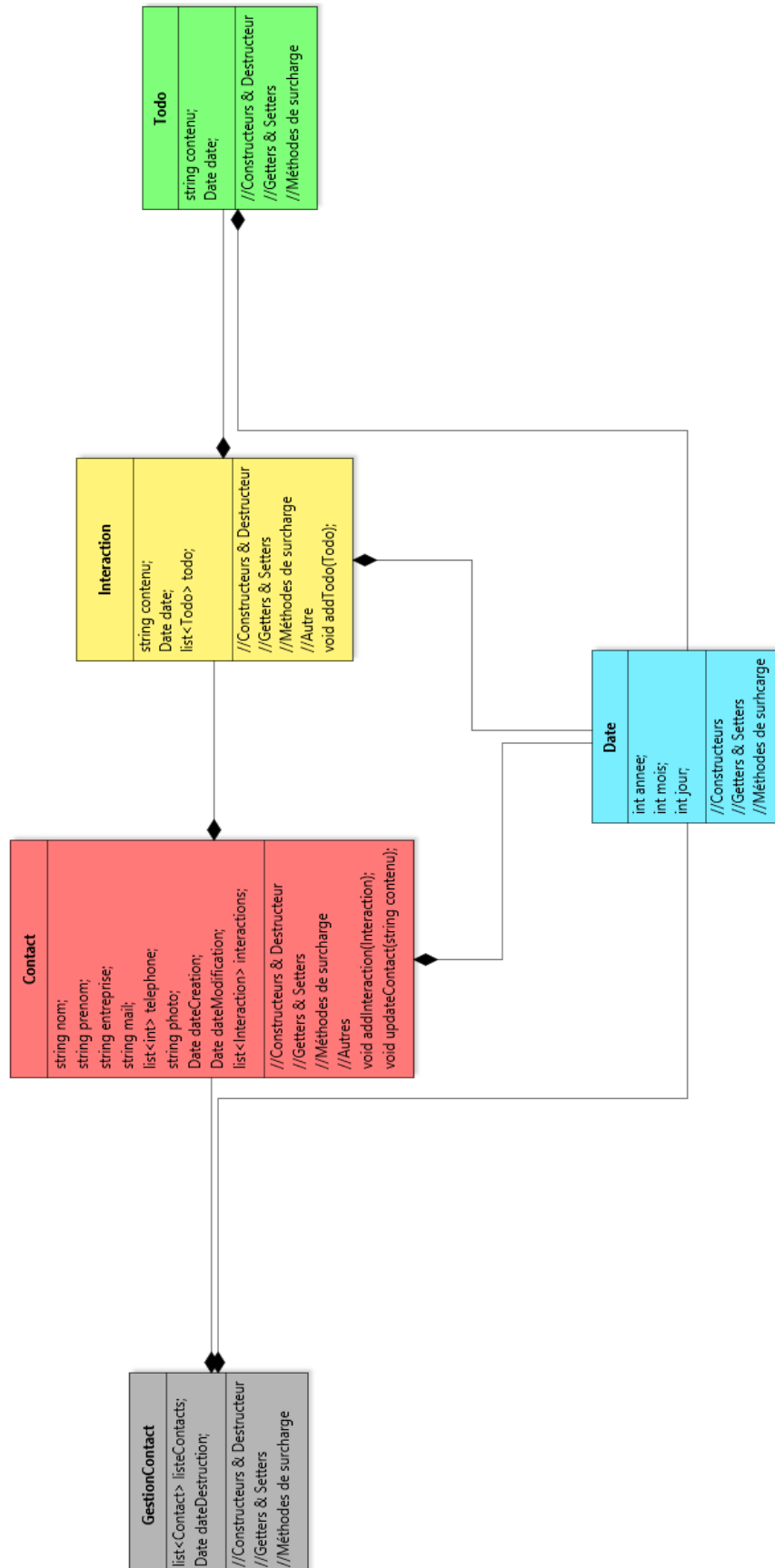


Diagramme de classes de la partie C++ de l'application

Projet L3 CDAA 2021 : Application de gestion de Contacts

Réalisation de l'interface graphique

Les Classes :

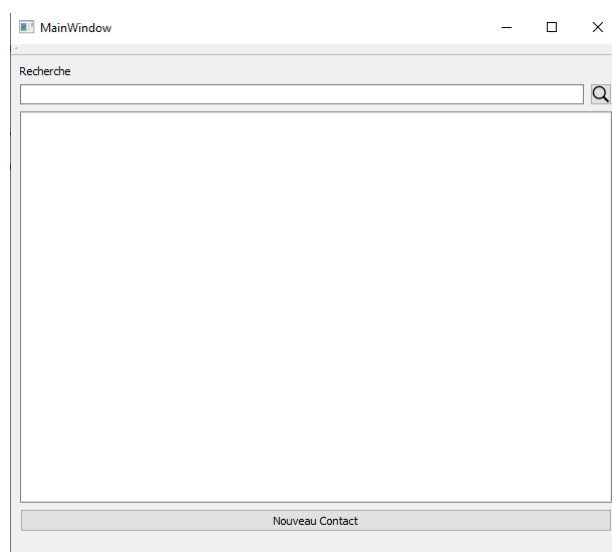
MainWindow	HomePage	NouveauContact	FeuilleContact	AjoutInteraction
Fenêtre principale de l'application possédant l'attribut GestionContact	Widget principale correspondant à l'affichage au lancement du programme	Widget permettant d'ajouter un contact à la liste de contacts	Widget permettant de modifier, voire de supprimer, un contact déjà existant	Widget permettant d'ajouter ou de supprimer une interaction

AjoutTodo

Widget permettant d'ajouter, de modifier ou de supprimer un todo

L'interface graphique est composée d'une fenêtre principale représentée par la classe *MainWindow*. Cette dernière possède en attribut un pointeur sur chaque autre classe de l'IHM, ainsi qu'un attribut de *GestionContact* sur lequel elle travaille. Sur cette fenêtre principale on affiche 2 Widgets, le Widget *HomePage* et le Widget *NouveauContact*.

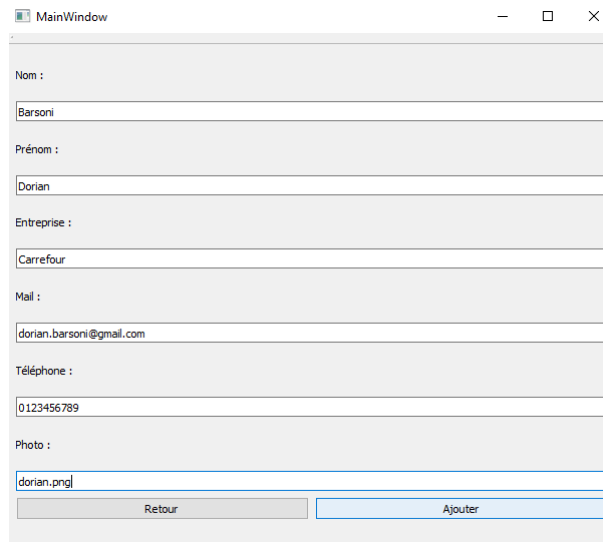
La classe *HomePage* est le Widget affiché par défaut lors du lancement de l'application. Elle possède une barre de recherche (non fonctionnelle par manque de temps) ainsi qu'une *QListWidget* dans laquelle on affiche les contacts, puis un bouton « Nouveau Contact » qui permet de changer le central widget de *MainWindow* en affichant à la place *NouveauContact*.



HomePage

Projet L3 CDAA 2021 : Application de gestion de Contacts

La classe *NouveauContact* sert à ajouter un nouveau contact. Elle est quant à elle composée de *QLineEdit* et de boutons. On a une *QLineEdit* pour le nom, le prénom, l'entreprise, l'adresse mail, le numéro de téléphone et la photo. Puis on a deux boutons, un bouton « Retour » pour revenir sur *HomePage*, et un bouton « Ajout » pour ajouter ledit contact.

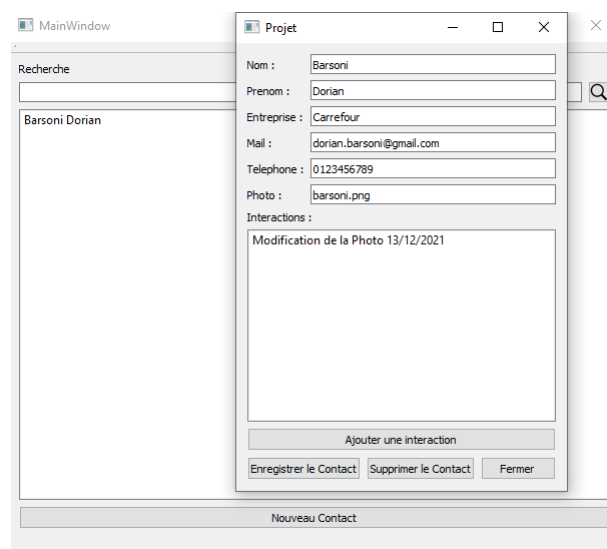


The screenshot shows a window titled "MainWindow" with a form for adding a new contact. The form contains the following fields and buttons:

- Nom : Barsoni
- Prénom : Dorian
- Entreprise : Carrefour
- Mail : dorian.barsoni@gmail.com
- Téléphone : 0123456789
- Photo : dorian.png
- Buttons: Retour, Ajouter

NouveauContact

Une fois le contact ajouté il apparaît dans la *QListWidget* de *HomePage*. On peut alors cliquer dessus ce qui fait apparaître une nouveau Widget, *FeuilleContact*. *FeuilleContact* est composé de toutes les informations du contact qu'on retrouve dans des *QLineEdit*. On y retrouve également des boutons pour enregistrer les modifications, supprimer le contact ou fermer le Widget. On retrouve en plus de cela une *QListWidget* dans laquelle on stocke les interactions du contact, ainsi qu'un bouton « Ajouter une interaction ». Dans l'exemple ci-dessous on modifie l'adresse URI de la photo du contact pour observer l'ajout dans la liste d'interaction.



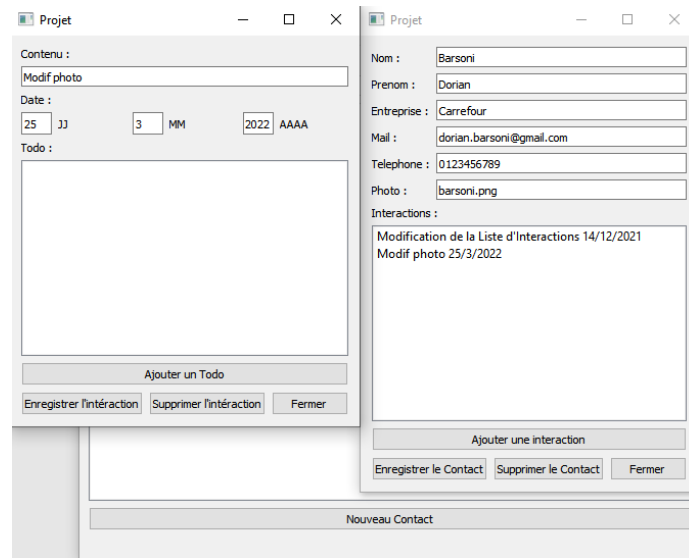
The screenshot shows a window titled "Projet" with a form for editing a contact. The form contains the following fields and buttons:

- Nom : Barsoni
- Prénom : Dorian
- Entreprise : Carrefour
- Mail : dorian.barsoni@gmail.com
- Téléphone : 0123456789
- Photo : barsoni.png
- Interactions : Modification de la Photo 13/12/2021
- Buttons: Ajouter une interaction, Enregistrer le Contact, Supprimer le Contact, Fermer

FeuilleContact

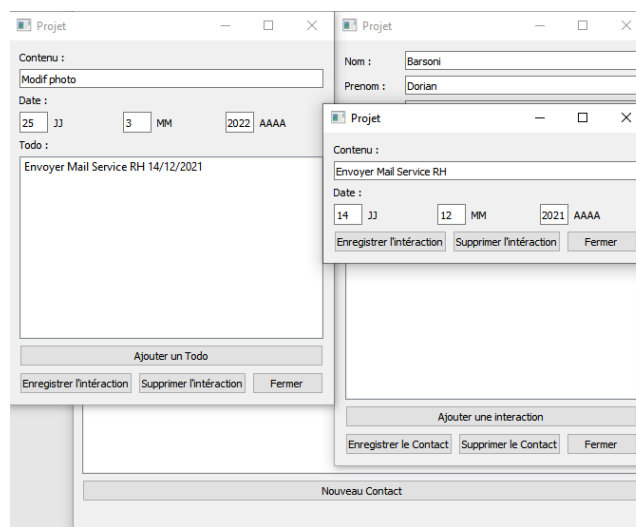
Projet L3 CDAA 2021 : Application de gestion de Contacts

On peut cliquer sur la nouvelle interaction apparue. Cela nous ouvre un autre Widget, *AjoutInteraction*. Dans ce widget on retrouve des *QLineEdit* où se trouvent le contenu et la date de l'interaction, mais également une autre *QListWidget* contenant les *Todo* de l'interaction. Dans l'exemple ci-dessous on modifie le contenu de l'interaction pour démontrer sa mise à jour dans la *QListWidget* de *FeuilleContact* (on notera également que la modification de la photo a été signalé par l'ajout d'une interaction « Modification de la liste d'interaction »).



AjoutInteraction

Enfin on peut cliquer sur « Ajouter un Todo » afin d'ajouter un *Todo* à cette interaction. Cela ouvrira le widget *AjoutTodo*, composé uniquement de *QLineEdit* dans laquelle on peut noter les informations du *Todo*. On peut de même que précédemment cliquer sur le *Todo* dans La *QListWidget* des *Todo* pour l'ouvrir. Ci-dessous on a ajouté un *Todo* qui est automatiquement mis à jour dans le *QListWidget*.



AjoutTodo

Projet L3 CDAA 2021 : Application de gestion de Contacts

Base de Données

Malgré plusieurs efforts je n'ai pas réussi à implémenter la base de données dans le programme. Vous retrouverez dans ce rapport l'implémentation théorique que je souhaitais réaliser.

Les Classes :

DataBase
Classe permettant d'enregistrer dans une base de données toutes les informations de l'application

Afin de réaliser une base de données capable de stocker les informations dont traite l'application il suffit d'utiliser 3 tables :

- **Contact**(id, nom, prenom, entreprise, mail, telephone, photo, dateCrea, dateModif)
- **Interaction**(idContact, idInteraction, id_todo, contenu, date)
- **Todo**(idContact, idInteraction, idTodo, contenu, date)

A l'aide de ces trois tables on peut reconstruire toutes les informations. On peut recréer tous les *Contact*, recréer toutes les *Interactions* et les lier à leur *Contact*, recréer tous les *Todo* et les lier à leur *Interaction*.

Afin de sauvegarder les informations nous avons 2 solutions :

- Soit enregistrer directement chaque création, suppression, modification dans la base
- Soit enregistrer dans la base de données l'attribut *GestionContact* à la fermeture du programme

La solution une semblant plus sur pour éviter la perte de données (en cas de coupure de courant par exemple) c'est celle que nous décidons d'implanter.

Ainsi, après chaque appuie sur les boutons d'enregistrement, de création ou de suppression, on a une émission de signal qui permet d'effectuer, le cas échéant, un INSERT INTO, UPDATE ou DELETE.

Enfin afin que le programme soit à jour à son lancement on exécute une fonction permettant de lire l'entièreté de la base de données afin de remplir l'attribut *GestionContact* avec lequel on travaille dans *MainWindow*.

Projet L3 CDAA 2021 : Application de gestion de Contacts

Conclusion

Durant ce compte rendu nous avons vu la manière dont nous avons réalisé une application de gestion de contacts.

La modélisation de l'application a été réalisé en C++ à l'aide de 5 classes : *GestionContact* permettant de gérer une liste de contact, *Contact* représentant un contact, *Interaction* représentant les interactions d'un contact, *Todo* représentant les choses à faire pour une interaction d'un contact, *Date* représentant une date. Ces classes sont imbriquées ensembles comme vu dans le diagramme de classes ci-dessus.

Concernant la partie interface graphique nous avons implémenté 6 nouvelles classes : *MainWindow* qui est la fenêtre principale du projet et qui possède tous les attributs sur lesquels nous travaillons (*GestionContact*, *Database...*), *HomePage* qui est le widget dans lequel apparait la liste de contact, *NouveauContact* qui est le widget d'ajout d'un *Contact*, *FeuilleContact* qui est le widget permettant de modifier la fiche d'un *Contact*, *AjoutInteraction* qui permet de créer/modifier/supprimer une *Interaction*, et *AjoutTodo* qui permet de créer/modifier/supprimer un *Todo*.

Enfin concernant la partie base de données nous avons utilisé uniquement trois tables pour représenter l'entièreté de nos données, la table **Contact**, **Interaction** et **Todo**. Chaque enregistrement/suppression dans l'application entraine immédiatement une mise à jour dans la base de données ce qui permet d'éviter une perte de données suite à un éventuel problème. Au lancement cette même base de données est lue en intégralité pour remplir l'attribut *GestionContact* de *MainWindow* avec lequel nous travaillons.