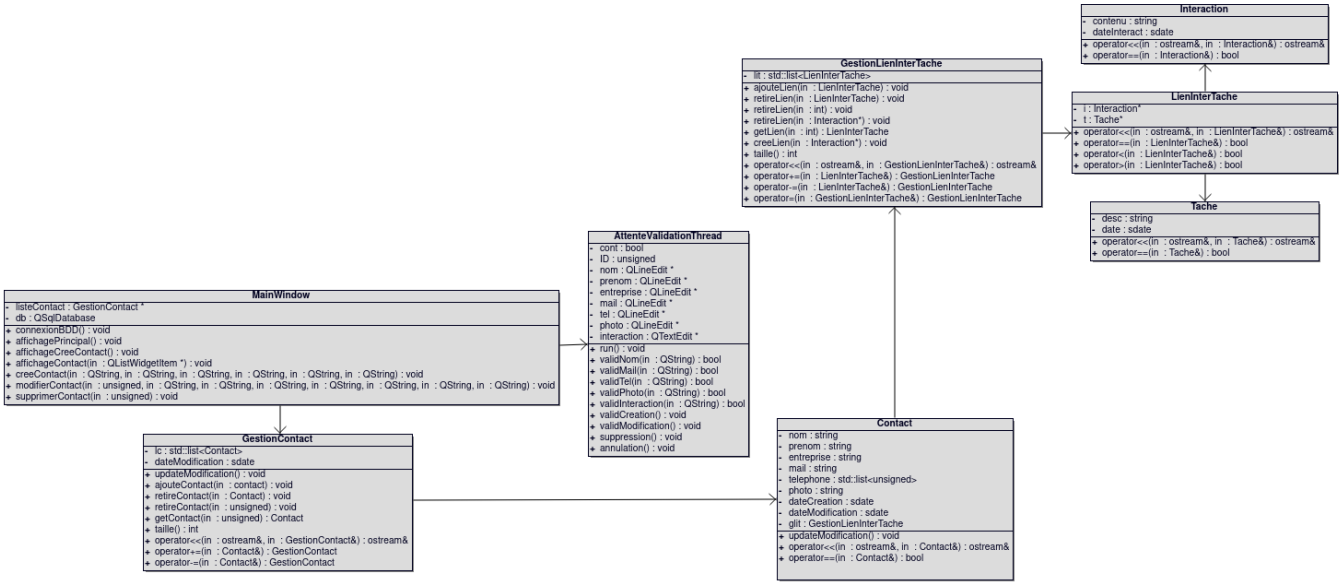
Jorge Alexandre (TD1-TP2)
Robert Quentin (TD1-TP1)

Compte rendu projet CDAA

Sommaire :

Diagramme UML :



Ce diagramme a été fait avec BOUML, vous pouvez retrouver les fichiers sources dans le dossier “diagrammeUML” ou alors vous avez également la capture d’écran “diagrammeUML.png”.

classe Contact :

La classe contact contient tous les attributs qui caractérisent un contact (nom, prenom, etc…). Elle contient aussi un attribut de type GestionLienInterTache qui est une liste de liens entre ses interactions et les potentielles tâches qui lui sont liées.

classe LienInterTache :

Cette classe fait le lien entre une interaction et la tâche qui peut lui être liée. On y retrouve donc le pointeur de l’interaction et le pointeur de la tâche le cas échéant sinon nullptr.

Le lien est créé dans le classe GestionLienIntertache qui possède une liste de liens et gère la création de ces liens. C’est ici que l’on retrouve le filtrage des @todo ou @date et donc la création ou non d’une tâche.

classe MainWindow :

C’est la fenêtre de l’application, elle contient la base de données ainsi que la liste des contacts de l’application. Elle gère tous les affichages ainsi que la création ou la modification des contacts dans sa liste. Elle utilise également la classe AttenteValidationThread dont les détails sur son fonctionnement sont donnés ci-dessous.

classe AttenteValidationThread :

Cette classe utilisée uniquement par MainWindow est un thread qui se lance lors de l'affichage de la création ou de la modification d'un contact. Son intérêt est que comme il n'est pas possible de passer des arguments au signal clicked() d'un bouton, pour récupérer les informations saisie dans les QLineEdit etc il aurait fallu avoir tout les champs de saisie en attributs de la MainWindow et encore cela aurait posé d'autres problèmes. La solution que nous avons choisie est de lancer un thread qui prend en paramètres les champs de saisie et qui attend le clique sur un bouton. Si c'est un bouton pour confirmer la création/modification alors il y a une vérification de la conformité des données saisie et si tout est correct alors les données sont renvoyées à la MainWindow et le thread s'arrête. Si c'est un bouton pour supprimer le contact, on renvoie l'indice du contact dans la liste et le thread s'arrête. Enfin, si c'est un bouton retour, le thread dit à la MainWindow de revenir sur l'affichage principal et s'arrête.

Base de données :

CONTACT (ID_contact, nom, prenom, entreprise, mail, telephone, photo, dateCreation, dateModification)

INTERACTION (ID_interaction, contenu, dateInteraction)

TACHE (ID_tache, descTache, dateTache)

LIENINTERTACHE (ID_lienInterTache, *ID_contact*, *ID_interaction*, *ID_tache*)

Logiquement, les tables "CONTACT", "INTERACTION" et "TACHE" contiennent les mêmes attributs que les classes correspondantes, on ajoute seulement un identifiant dans les tables de la base données qui servira à la fois de clé primaire pour ces tables mais aussi de clé étrangère pour la table LIENINTERTACHE.

La table LIENINTERTACHE tout comme la classe du même nom, fait le lien entre les interactions et les tâches. Là où dans la classe nous retrouvons les pointeurs de l'interaction et de sa tâche liée, dans la table nous retrouvons les identifiants de cette même interaction et de sa tâche. Dans la table, on retrouve également l'ID du contact auquel cette interaction est liée.

Ainsi, il nous est très facile de retrouver quelle interaction est liée à quel contact et si cette interaction est liée à une tâche, laquelle.

Signaux / Slots :

Classe MainWindow :

signaux :

- toAffichagePrincipal() : ce signal est connecté au slot affichagePrincipal().

slots :

- affichageCreeContact() : ce slot est connecté au signal clicked() du bouton de création d'un nouveau contact de l'affichage principal. Il change donc l'affichage dans la fenêtre pour y afficher une interface de création de contact avec des champs de saisie.

- affichageContact() : ce slot est connecté au signal itemClicked() de la QListWidget qui contient la liste des contacts. Lors du clique sur un contact, on obtient un nouvel affichage où l'on retrouve toute les informations de ce contact ainsi que ses interactions et sa liste de tâche. A noter que tous les champs sont éditables directement et que l'on retrouve un bouton pour valider tout changement éventuel. On y retrouve également un bouton pour supprimer ce contact.

- creeContact(...) : ce slot est connecté au signal toCreeContact(...) de la classe AttentevalidationThread. Il prend en paramètre 6 QString qui sont les informations saisies dans la création de contact. Ce slot va donc créer un nouveau contact à partir de ces informations et l'ajouter dans la liste.

- affichagePrincipal() : ce slot est connecté au signaux toAffichagePrincipal() des classes MainWindow et AttenteValidationThread. Il affiche les éléments de la fenêtre principal donc la liste de contact ainsi qu'un bouton pour en créer un nouveau.

- modifierContact(...) : ce slot est connecté au signal toModifierContact(...) de la classe AttenteValidationThread. Il prend en paramètre un unsigned et 7 QString, l'unsigned pour l'indice du contact à modifier et les QString sont les nouvelles informations sur ce contact. Ce slot va donc remplacer les informations du contact en question par celle saisie et passé en paramètre.

- supprimerContact(unsigned) : ce slot est connecté au signal toSupprimerContact(unsigned) de la classe AttenteValidationThread et passe l'indice du contact à supprimer. Ce slot supprime donc ce contact.

Classe AttenteValidationThread :

signaux :

- toCreeContact(...) : ce signal est connecté au slot creeContact(...) de MainWindow.
- toAffichagePrincipal() : ce signal est connecté au slot affichagePrincipal() de MainWindow.
- toModifierContact(...) : ce signal est connecté au slot modifierContact(...) de MainWindow.
- toSupprimerContact(unsigned) : ce signal est connecté au slot supprimerContact(unsigned) de MainWindow.

slots :

- validCreation() : ce slot est connecté au signal clicked() du bouton de validation de la création du contact de l'affichage de création de contact de MainWindow. Ce slot vérifie si les informations saisies sont correctes puis renvoie les données à MainWindow.

- annulation() : ce slot est connecté aux signaux clicked() des boutons de retour des affichage de création et de modification de contact de MainWindow. Ce slot dit simplement à MainWindow de revenir à l'affichage principal et arrête le thread.

- validModification() : ce slot est connecté au signal clicked() du bouton de validation de la modification du contact de l'affichage de modification de contact de MainWindow. Ce slot vérifie si les informations saisies sont correctes puis renvoie les données à MainWindow.

- suppression() : ce slot est connecté au signal clicked() du bouton de suppression du contact de l'affichage de modification de contact de MainWindow. Ce slot renvoie l'indice du contact à supprimer à MainWindow.