Lycée TOUCHARD WASHINGTON

Section des Techniciens Supérieurs
Systèmes Numériques option
Informatique et Réseaux

TD2 - Socket sous Qt

Distributeur Automatique de Billets ***- côté client***

- *Date : novembre 2022*
- *Version : 2*
- *Référence : TD2 – Socket sous Qt (DABClient).odt*

1. Objectif

- *Utilisation des sockets avec la bibliothèque **Qt***
- *Comprendre le fonctionnement des sockets en mode événementiel*
- *Codage de l'information*
- *Communication réseau*
- *Flux de données **QDataStream***

2. Conditions de réalisation

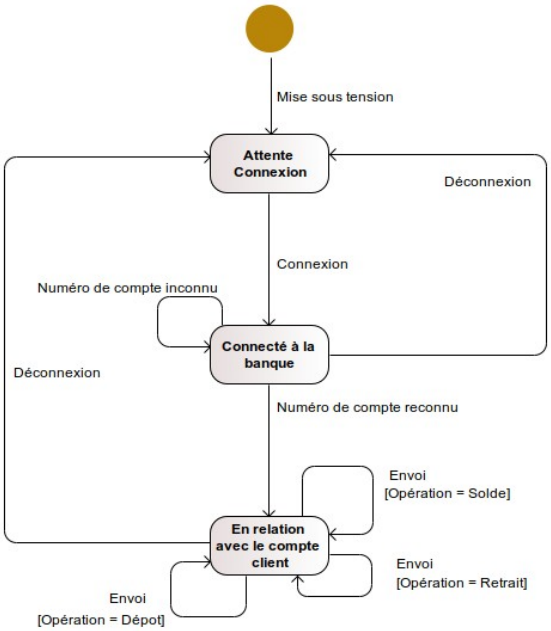
- *Ce fichier contient des liens hypertextes.*
- *Ressources utilisées :*
  - *Un PC sous Linux*
  - *Un serveur est disponible*
  - *Qt-creator*
  -

3. Ressources

Les classes socket dans la technologie QT, consulter le site https://doc.qt.io/qt-6/qtnetwork-programming.html et plus particulièrement la classe, ***QtcpSocket*** *https://doc.qt.io/qt-6/qtcpsocket.html.*

4. Le besoin

Les distributeurs automatiques de billets (DAB) permettent de se connecter à une banque. Ils offrent la possibilité d'interroger un compte pour en connaître le solde, de déposer de l'argent et d'en retirer. Les fonctionnalités attendues côté client sont décrites par le diagramme d'état du système suivant :

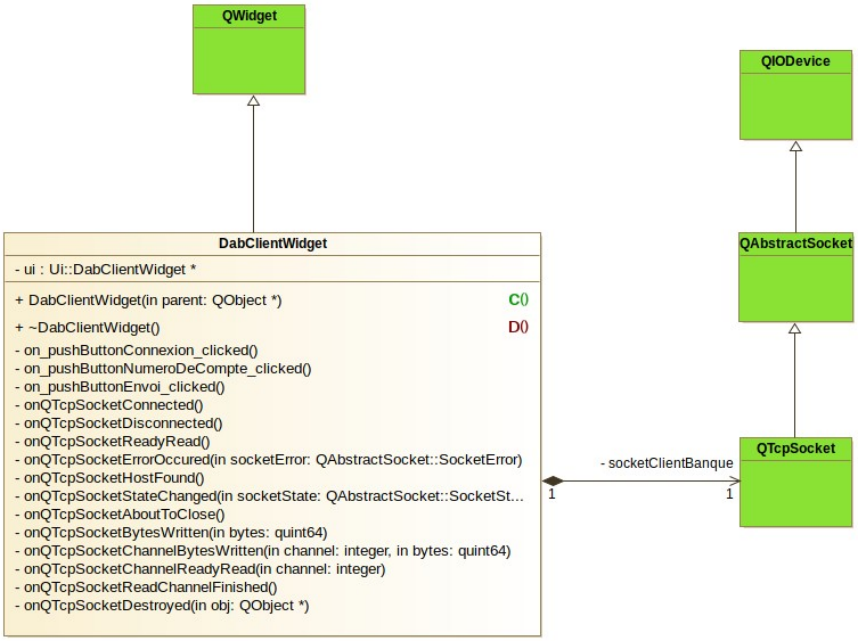


Le diagramme à états ci-contre indique que c'est seulement lorsque le client est connecté à sa banque et que son numéro de compte est reconnu que les opérations vers la banque peuvent être réalisées. La déconnexion reste possible à chaque état.

5. Réalisation

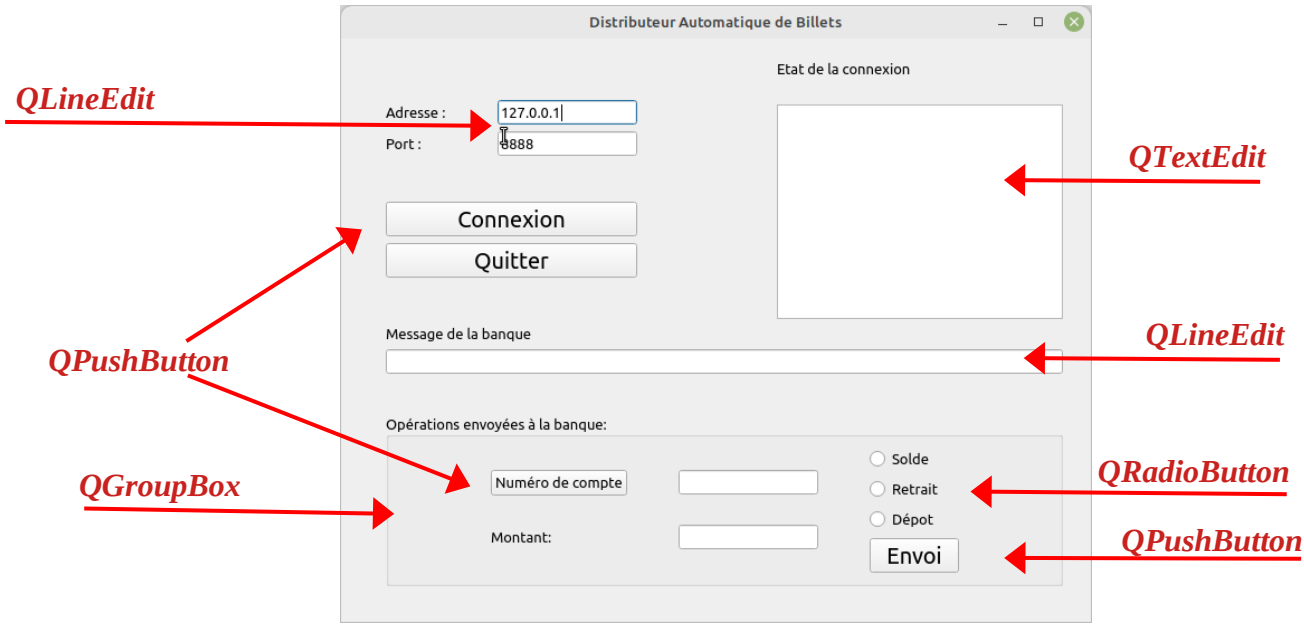
5.1. Création du projet

Créez un projet **ClientDAB** de type Application Qt utilisant les Widgets. La classe principale se nomme ***DabClientWidget***. Elle hérite de ***QWidget*** comme le montre le diagramme de classes partiel ci-dessous. C'est la seule classe où vous aurez à intervenir.



5.2. Création de l'IHM

L'interface attendue possède l'aspect suivant :



1. Nommez chaque Widget avec la convention de nommage habituelle.
2. Comme indiquer dans le diagramme à état du système, tant que la connexion avec la banque n'est pas établie, les opérations vers la banque ne sont pas possibles (le **groupBox** relatif aux opérations vers la banque doit-être désactivé, la zone d'édition pour les messages de la banque est en lecture seule. La zone d'édition contenant l'état de la connexion avec la banque indique les différents états de la connexion et les messages d'erreur système. Le bouton Connexion devient Déconnexion lorsque la connexion est établie.

6. Utilisation de la classe QTcpSocket

6.1. Traitement des signaux

3. Codez la relation entre les classes **DabClientWidget** et **QtcpSocket** de manière automatique.
4. Reliez les différents signaux en provenance du socket client à la classe **DabClientWidget**, en respectant les conventions de nommage vues en cours.
5. Réalisez les traitements correspondant à chaque signal en réalisant la mise à jour de la zone d'édition retraçant l'état de la socket.
6. Réalisez le codage pour la connexion et la déconnexion de la socket.

6.2. Réalisation des trames d'échange entre la banque (côté serveur) et le DAB (côté client).

La trame à transmettre pour fournir le numéro de compte se décompose ainsi :

Taille des données	Commande	Valeur
De type **quint16**	'N' - Un caractère sous la forme d'un **QChar**	Un entier de type **int**

Pour forger cette trame, nous utiliserons un **QBuffer** et un **QDataStream**. Voici un exemple façon de procéder :

Envoi du numéro de compte — **dabclientwidget.cpp**

```
void DabClientWidget::on_pushButtonNumeroDeCompte_clicked()
{
    quint16 taille = 0;
    QChar commande = 'N';
    int numCompte = ui->lineEditNumeroDeCompte->text().toInt();
    QBuffer tampon;
    if(tampon.open(QIODevice::WriteOnly))
    {
        QDataStream sortie(&tampon);
        sortie << taille << commande << numCompte ;
        taille = tampon.size() - sizeof(taille);
        tampon.seek(0);
        sortie << taille;
        socketClientBanque.write(tampon.buffer());
    }
}
```

7. Réalisez le code nécessaire pour l'envoi du numéro de compte vers la banque. Le numéro est transmis s'il est supérieur à 0.

Tous les messages en provenance de la banque sont de la forme :

Taille des données	Message
De type **quint16**	Une chaîne de caractère sous la forme d'une **QString**

Comme il n'est pas possible de connaître à l'avance le nombre d'octets que le client reçoit, procédez par étape en utilisant les flux :

Réception d'un message de la banque — **dabclientwidget.cpp**

```
void DabClientWidget::onQTcpSocketReadyRead()
{
    quint16 taille = 0;
    QString message;
    // si le nombre d'octets reçu est au moins égal a celui de la taille
    // de ce que l'on doit recevoir
    if(socketClientBanque.bytesAvailable() >= static_cast<qint64>(sizeof(taille)))
    {
        // association de la socket au flux d'entrée
        QDataStream entree(&socketClientBanque);
        // extraction de la taille de ce que l'on doit recevoir
        entree >> taille;
        // si le nombre d'octets reçu est au moins égal a celui de ce que l'on doit
        // recevoir
        if (socketClientBanque.bytesAvailable() >= static_cast<qint64>(taille))
        {
            entree >> message;
            // complétez pour afficher dans la QLineEdit
        }
    }
}
```

8. Réalisez le code nécessaire pour recevoir la réponse de la banque et l'afficher dans la zone d'édition prévue à cet effet.

Autres opérations vers la banque :

Lors de l'affichage de l'IHM du distributeur, on souhaite que le bouton radio Solde soit coché par défaut.

9. Réalisez le code nécessaire dans la méthode qui vous semble la plus appropriée.

L'appui sur le bouton **Envoi** doit envoyer à travers la socket le type d'opération souhaité, suivi éventuellement du montant de la transaction suivant le format de trame :

Taille des données	**Commande**	**Valeur**
De type **quint16**	Pour le retrait : 'R' - Un caractère sous la forme d'un **QChar**	Un réel de type **float**
De type **quint16**	Pour le dépôt : 'D' - Un caractère sous la forme d'un **QChar**	Un réel de type **float**
De type **quint16**	Pour le solde : 'S' - Un caractère sous la forme d'un **QChar**	Sans objet

Pour la réponse de la banque si tout va bien vous n'avez rien à faire...