

Concernant la base de donnée :

Table Commande:

```
CREATE TABLE [Commande]
(
    [ID] INT NOT NULL PRIMARY KEY identity(1, 1),
    [Guitare] int not null,
    [Status] varchar(32) not null default 'En cours de traitement',
    foreign key (Guitare) references Guitare(ID) on delete cascade
)
```

Création d'une nouvelle table 'Devis' contenant l'ID du devis ainsi que la date du devis, le délais de validité ainsi que le commentaire possible

Voici alors la table Devis proposé

(relation 0,1) Commande - Devis

```
CREATE TABLE [Devis]
(
    [ID] INT NOT NULL PRIMARY KEY identity(1, 1),
    [DateCreation] date not null,
    [DateExpiration] date not null,
    [Commentaire] varchar(128),
    [Prix] int not null,
    [Accepte] bit not null default 0
)
```

Une version alternative de cette table peut être envisagée afin de faire référence a la commande

(relation 0,n) Commande - Devis

```
CREATE TABLE [Devis]
(
    [ID] INT NOT NULL PRIMARY KEY identity(1, 1),
    [DateCreation] date not null,
    [DateExpiration] date not null,
    [Commentaire] varchar(128),
    [Prix] int not null,
    [Accepte] bit not null default 0,
    [Commande] int not null,
    foreign key (Commande) references Commande(ID) on delete cascade
)
```

En vue du fonctionnement interne de l'application la première version sera envisagée, cela nécessite alors de mettre à jour la table Commande afin d'y rajouter la clé étrangère faisant référence au Devis,

Voici alors la table Commande à jour :

```
CREATE TABLE [Commande]
(
    [ID] INT NOT NULL PRIMARY KEY identity(1, 1),
    [Guitare] int not null,
    [Devis] int,
    [Status] varchar(32) not null default 'En cours de traitement',
    foreign key (Guitare) references Guitare(ID) on delete cascade
    foreign key (Devis) references Devis(ID) on delete cascade
)
```

Concernant la librairie:

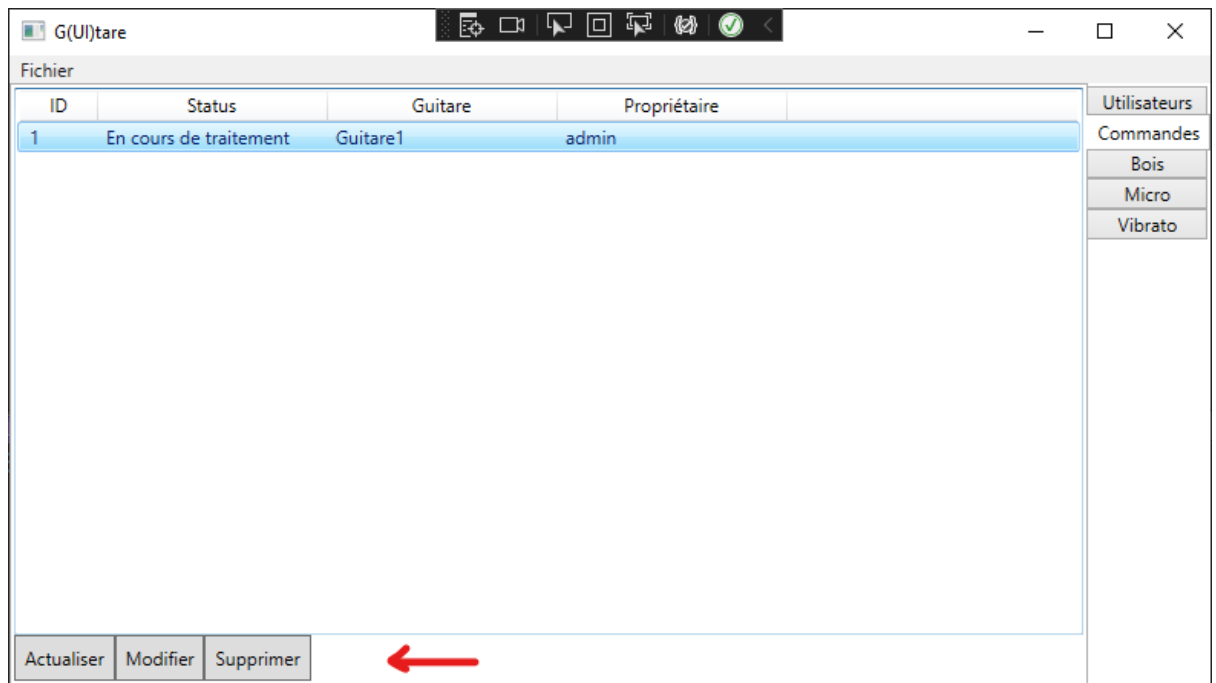
La création d'un nouvel objet (DTO) représentatif d'un devis ainsi que les fonctions internes de création, lecture, édition et suppression

Concernant l'API REST:

Un nouveau contrôleur devra être créé afin de permettre d'interagir avec les devis (création, lecture, édition, suppression) à travers la librairie à jour

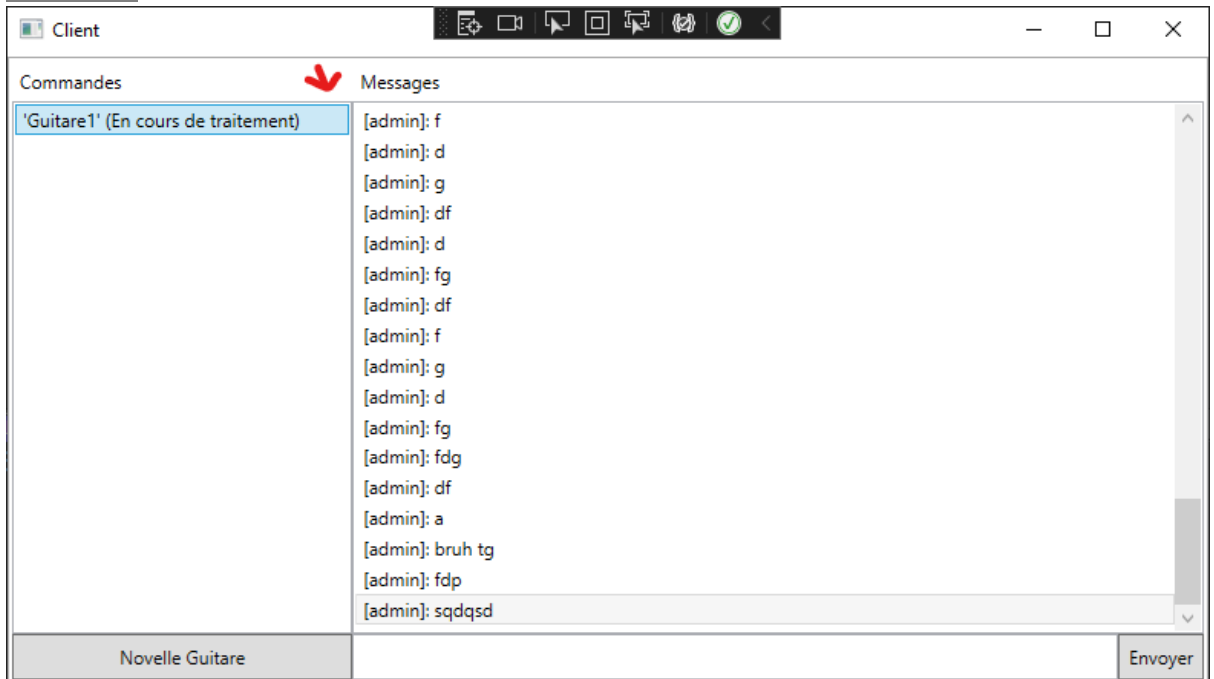
Concernant l'interface:

Coté luthier :



L'ajout d'un nouveau bouton 'Devis' dans l'onglet des commandes ouvrant une nouvelle fenetre permettant de proposer/éditer un devis

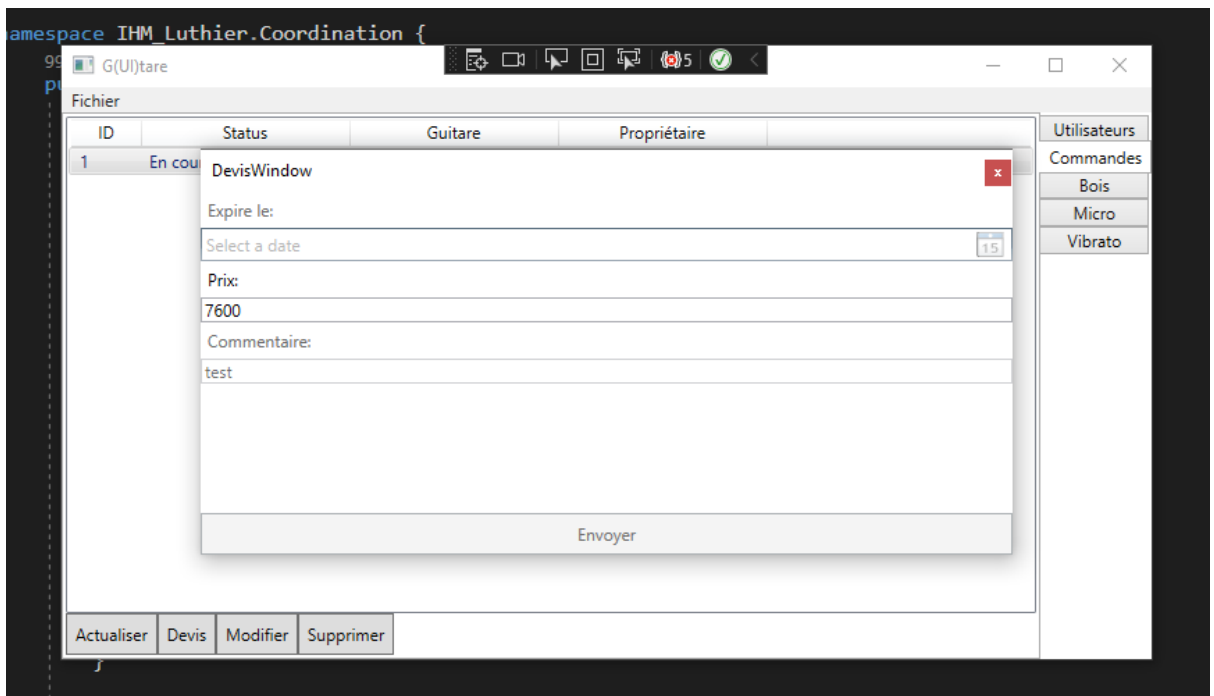
Coté client:



Tout d'abord élargir la colonne contenant les commandes afin de rajouter un bouton 'Devis' à côté de chaque commande de la liste, ce bouton permettra d'afficher le devis proposé par le luthier et permettra ainsi de l'accepter ou de le refuser, une fois accepté le devis ne sera plus modifiable.

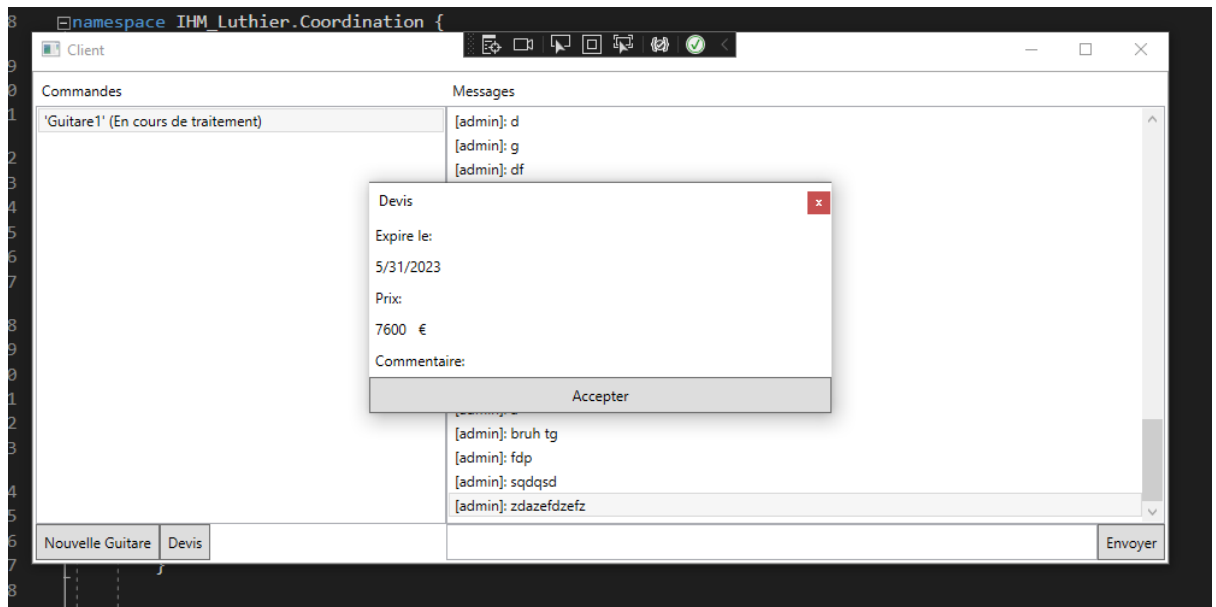
Interface finale:

Luthier :



Le luthier peut proposer un devis et l'envoyer au client, une fois accepté par le client le devis ne peut plus être édité (la fraude c'est pas bien)

Client :



Le client reçoit le devis et peut l'accepter ou non, un chat est à sa disposition afin de pouvoir négocier avec le luthier

Au niveau de l'API REST :

Commande		^
GET	/api/Commande/GetAllCommande	▼
GET	/api/Commande/GetCommandeByUtilisateurID	▼
GET	/api/Commande/GetCommandeByID	▼
POST	/api/Commande/AddCommande	▼
POST	/api/Commande/RemoveCommandeByID	▼
PUT	/api/Commande/UpdateCommande	▼
Devis		^
GET	/api/Devis/GetAllDevis	▼
GET	/api/Devis/GetDevisByID	▼
POST	/api/Devis/AddDevis	▼
POST	/api/Devis/RemoveDevisByID	▼
PUT	/api/Devis/UpdateDevis	▼

```

Commande ▾ {
  id                integer($int32)
  status            string
                   nullable: true
  guitare           integer($int32)
  guitareObj        > {...}
                   nullable: true
  devis             integer($int32)
  devisObj          > {...}
                   nullable: true
}

```

```

Devis ▾ {
  id                integer($int32)
  dateCreation       string($date-time)
  dateExpiration     string($date-time)
  commentaire        string
                   nullable: true
  prix              integer($int32)
  accepte            boolean
}

```

Le DTO (Data Transfer Object) correspondant au devis a été rajouté et celui correspondant à la commande a été mis à jour, les routes de l'API REST nécessaires ont été rajoutées (/api/Devis/*)

Permettant les opérations basiques CRUD (Create, Read, Update, Delete) nécessaire à la manipulation des devis.

Répo : <https://github.com/Nephys/TPGuitare> (privé, demandez l'accès)