



Projet de modélisation

Le Garage

On souhaite réaliser l'architecture logicielle d'un garage selon certaines règles.

Encadré par :

M. CHAUSSARD John

Réalisé par :

BENSAOULA Nidam

OLIVE Valentin

Table des matières

Introduction	2
I. Premier rendu (4 avril).....	3
A. Définition des bornes du système.....	3
B. Identification des acteurs primaires et secondaires.	4
C. Diagramme de cas d'utilisation du système.	5
D. Diagramme de classes du système	6
II. Second rendu (6 mai)	8
A. Nouveau diagramme de cas d'utilisation	8
B. Nouveau diagramme de classe	10
C. Diagramme d'activité du système.	11
D. Diagramme d'états-transitions du système.	13
E. Diagramme de séquence d'un scénario classique.	15
F. Diagramme de communication d'un scénario classique.	17
Conclusion.....	19

Introduction

Lors de notre sixième semestre de licence informatique, nous étudions la modélisation et plus particulièrement le langage UML. Pour évaluer cet apprentissage, un projet de modélisation nous a été donné. Ce projet est la modélisation d'un système de garage de réparation automobile.

Nous devons donc réaliser une modélisation de ce système en réalisant les différents diagrammes d'UML. Ces diagrammes sont les diagrammes de cas d'utilisation, de classe, de séquence, de communication, d'activité et d'état-transition.

Dans chacun de ces diagrammes il y a des informations très importantes concernant le système.

Le diagramme de cas d'utilisation recense les acteurs et les actions de chaque acteur. Le diagramme de classe recense les relations entre les différentes « entités » de notre système. Le diagramme de séquence est un diagramme qui montre le déroulement de chaque action selon un scénario donné.

Le diagramme de communication recense les mêmes informations que le diagramme de séquence, mais les montre sous une autre forme. Le diagramme d'activité montre l'enchaînement des actions sur « l'entité » principale de notre système.

Tandis que le diagramme d'état-transition permet de recenser les différents changements des objets importants de notre système.

Nous allons dans un premier temps réaliser les diagrammes de cas d'utilisation et de classe puis nous réaliserons par la suite les diagrammes que nous avons mentionnés plus tôt.

I. Premier rendu (4 avril)

A. Définition des bornes du système.

Les bornes du système sont d'une part ce que notre système sera en charge de gérer et d'autre part ce qu'il ne pourra pas gérer.

Notre système ne gèrera pas la solvabilité du client. En effet, lorsque le client payera par carte bleue ou par chèque, la banque se chargera d'effectuer le versement du montant des réparations même si le client n'est pas solvable.

Notre système ne gère pas non plus le fait que l'on peut faire appel à plusieurs fournisseurs avant de pouvoir commander les pièces. Puisque la disponibilité des pièces est faite de notre côté avec la base de données stock, mais nous ne savons pas si le fournisseur est en mesure de nous fournir les pièces qu'il nous manque.

Notre système gèrera par contre le fait que pour un client donné, nous avons en effet toutes ses informations tel que son adresse, son numéro de téléphone, son nom, son prénom. Notre système gèrera en plus le fait qu'un client peut posséder plusieurs voitures. Et nous enregistrerons dans la base de données client : le client, ses voitures et les réparations effectuées sur chacune de ses voitures.

Notre système ne gèrera pas l'effacement de la base de données client au bout de cinq ans. Cette fonction sera gérée par la base de données client elle-même.

B. Identification des acteurs primaires et secondaires.

Après une lecture en détail du sujet, nous en avons conclu qu'il y avait des acteurs primaires et des acteurs secondaires.

Les acteurs primaires sont au nombre de trois : le client, l'opérateur et le technicien.

Le client est un acteur primaire puisqu'il dépose sa voiture dans un garage et les réparations ne seront faites sur sa voiture seulement s'il a accepté le contrat (devis). Il inspecte sa voiture avec le technicien pour enregistrer l'état de sa voiture. Il doit aussi payer les réparations de ce fait c'est un acteur primaire.

Le technicien est aussi un acteur primaire puisque c'est cet acteur qui va répertorier l'état de la voiture, le temps d'immobilisation de la voiture, les pièces nécessaires aux réparations. Une fois toutes ces informations répertoriées, il les transmet à l'opérateur.

L'opérateur est également un acteur primaire au système puisque lui s'occupe d'enregistrer le client si celui-ci est nouveau. Il s'occupe de vérifier la disponibilité des pièces nécessaires aux réparations et de les commander si elles ne sont pas disponibles. C'est lui qui modifie l'état du stock présent. C'est également lui qui propose le devis au client.

Selon les choix de modélisation que nous avons effectués, nous avons donc quatre acteurs secondaires. Ces acteurs secondaires sont : la banque, la base de données client, la base de données stock et le fournisseur.

Le premier acteur secondaire est la banque. Nous avons mis une banque puisque le client peut payer ses réparations soit par espèces, soit par chèque, soit par carte bleue. Dans ces deux derniers cas, la banque est obligée de faire partie de notre système en tant qu'acteur secondaire pour gérer la transaction des paiements par carte bleue et par chèque.

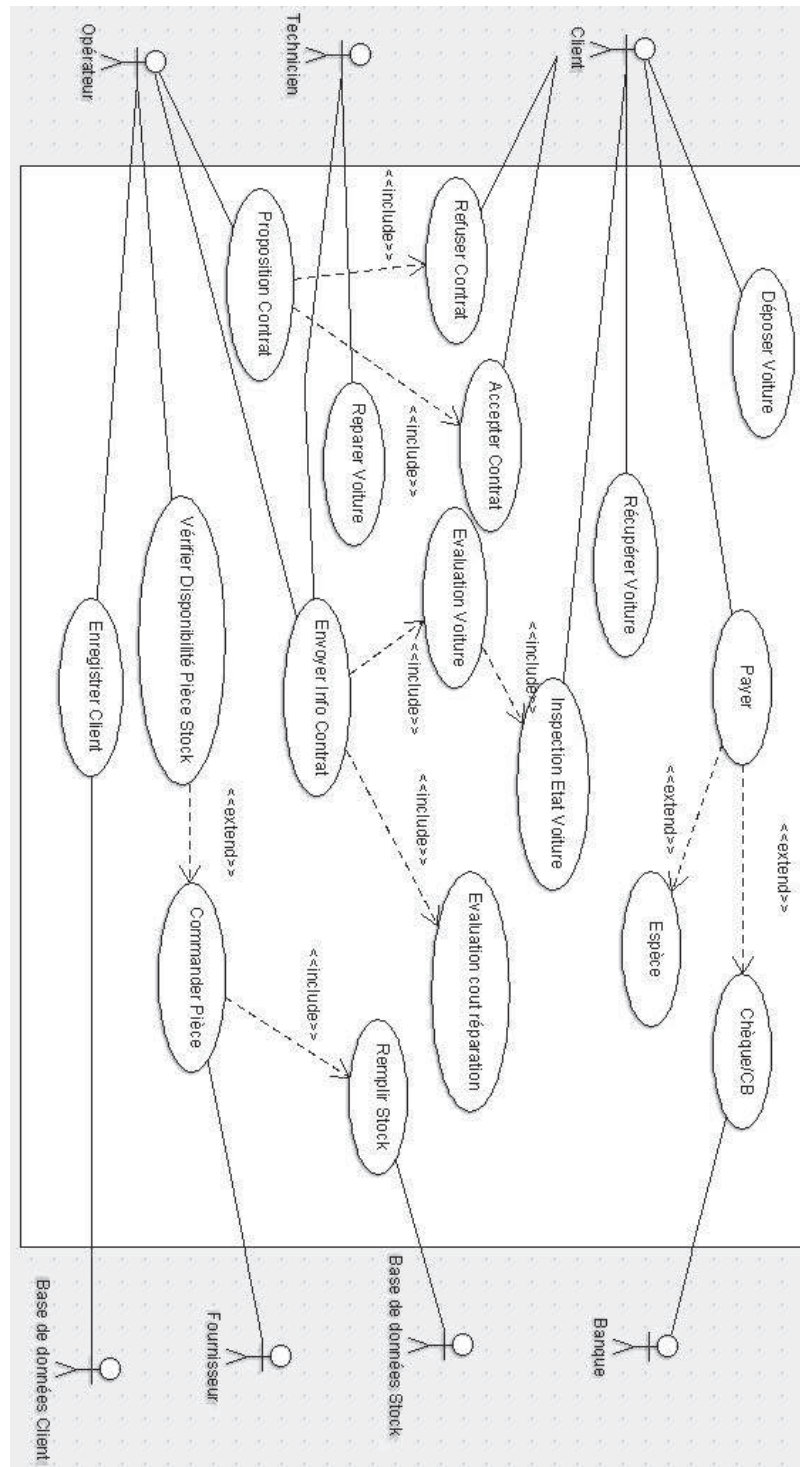
Le second acteur secondaire est la base de données client. Nous avons choisi de mettre en place une base de données client qui répertoriera tous les clients de notre garage. Lorsqu'un client se présentera dans notre garage, l'opérateur n'aura qu'à vérifier si le client existe dans la base de données et si ce n'est pas le cas, le créer.

Le troisième acteur secondaire est la base de données stock. Nous avons choisi de mettre en place une base de données pour la gestion de notre stock. Cela paraît plus simple de gérer notre stock à l'aide d'une base de données puisque l'opérateur pourra vérifier la disponibilité des pièces nécessaires aux réparations directement depuis notre système en interrogeant la base de données stock. Il pourra également à l'arrivée d'une commande modifier le stock en renseignant les quantités de pièces reçues.

Le dernier acteur secondaire est le fournisseur. Cet acteur n'apparaît dans notre système qu'au moment de commander des pièces. Il n'a donc aucun impact sur notre système.

C. Diagramme de cas d'utilisation du système.

Pour la réalisation du diagramme de cas d'utilisation, nous avons été amenés à faire des choix. En premier lieu, nous avons dû différencier les actions effectuées par le technicien de celles de l'opérateur. En effet, c'est l'opérateur seul qui s'occupera du logiciel, quant au technicien, il s'occupera de la voiture et de transmettre les informations relatives à celle-ci à l'opérateur, qui, après réception, établira un devis qu'il proposera au client.



D. Diagramme de classes du système

PanierCommande a été créé dans le but de décrire au mieux le processus de commande des pièces chez un fournisseur. La classe contrat a été créée dans le but de lier le client et sa voiture aux autres acteurs primaires.

En ce qui concerne les cardinalités, un contrat ne peut être lié qu'à une seule voiture à la fois, celui-ci est aussi lié à un seul client. Cependant, plusieurs opérateurs peuvent intervenir sur un même contrat pour modifier toutes sortes d'informations.

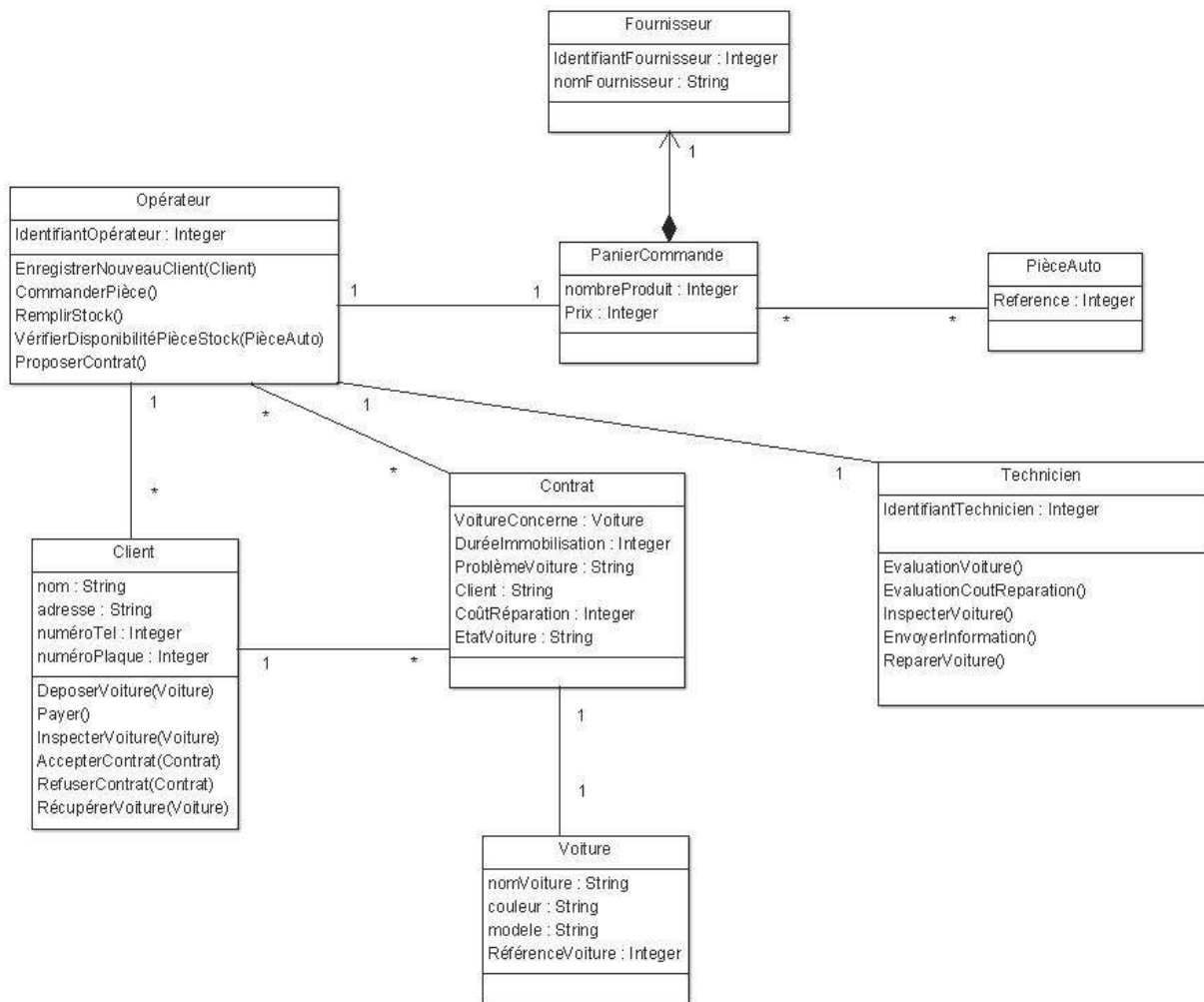
Un client peut avoir plusieurs voitures à faire réparer, ou bien réparer à plusieurs reprises une même voiture. Il devra pour cela effectuer plusieurs contrats.

Un opérateur peut s'occuper d'une multitude de clients, il sera amené à enregistrer un contrat par client, il peut donc s'occuper de plusieurs contrats.

Un technicien doit pouvoir transmettre des informations à une personne par rapport à un véhicule inspecté, il est donc lié à un seul opérateur.

Un opérateur est chargé de commander les pièces si nécessaire, il est donc lié à un unique panier de commande où toutes les pièces voulues sont ajoutées.

Comme vu en TD, nous avons décidé de lier un fournisseur et un panier de commande à l'aide d'une composition sachant que si l'on supprime un panier de commande, l'accès au fournisseur n'est plus disponible.



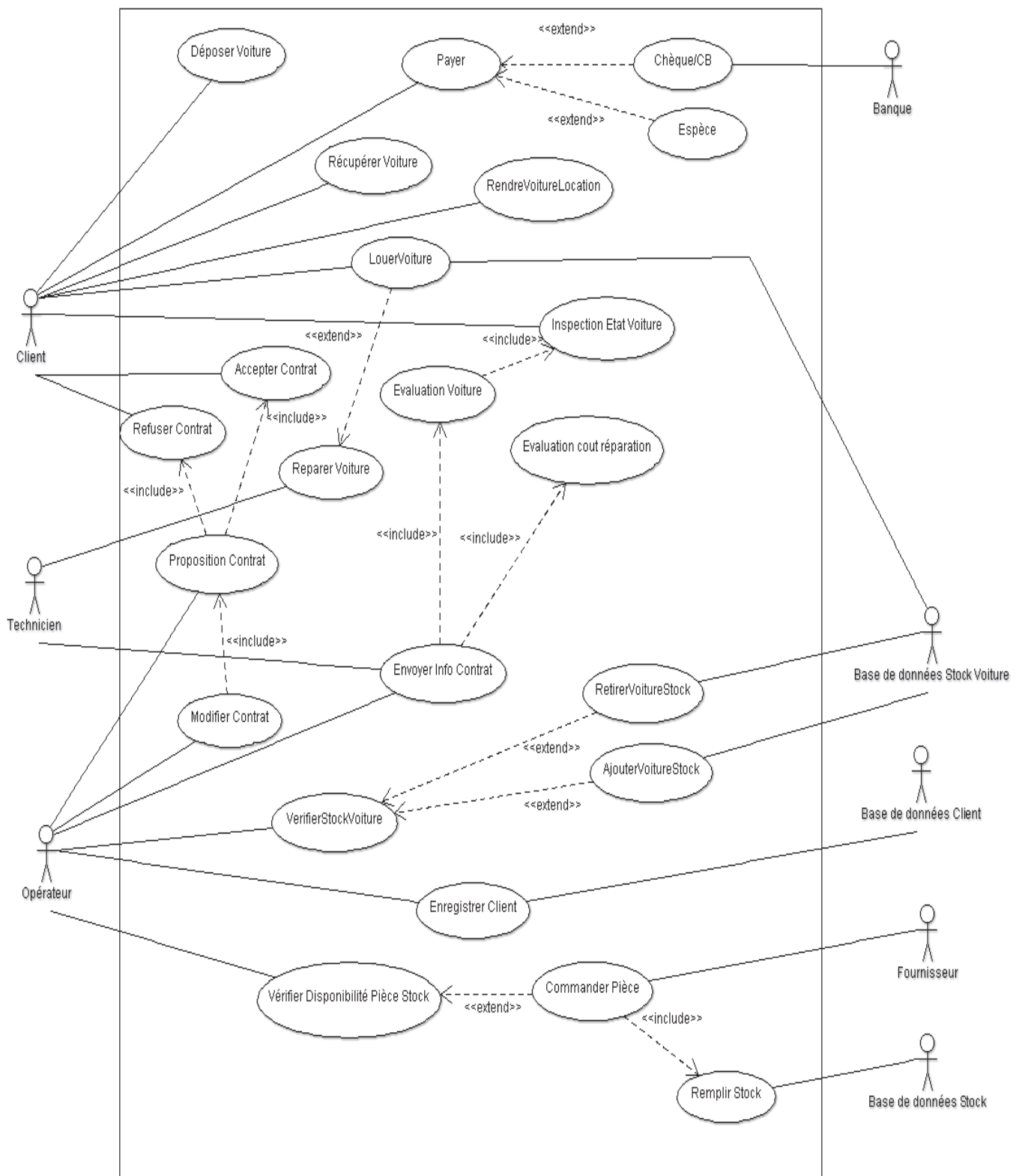
II. Second rendu (6 mai)

A. Nouveau diagramme de cas d'utilisation

Suite à une amélioration de notre système souhaitée par le client, nous avons dû modifier notre diagramme de cas d'utilisation. Pour pouvoir gérer le fait qu'un client du garage puisse avoir la possibilité de louer une voiture le temps que les réparations soient effectuées, nous avons créé des cas d'utilisation qui sont : LouerVoiture, RendreVoitureLocation, VérifierStockVoiture, RetirerVoitureStock et AjouterVoitureStock. Nous avons pris également la décision de créer un acteur secondaire supplémentaire qui est Base de données Stock Voiture.

Nous nous sommes rendus compte que nos liens extend n'étaient pas orientés dans le bon sens donc nous les avons modifié.

Voici notre nouveau diagramme de cas d'utilisation :

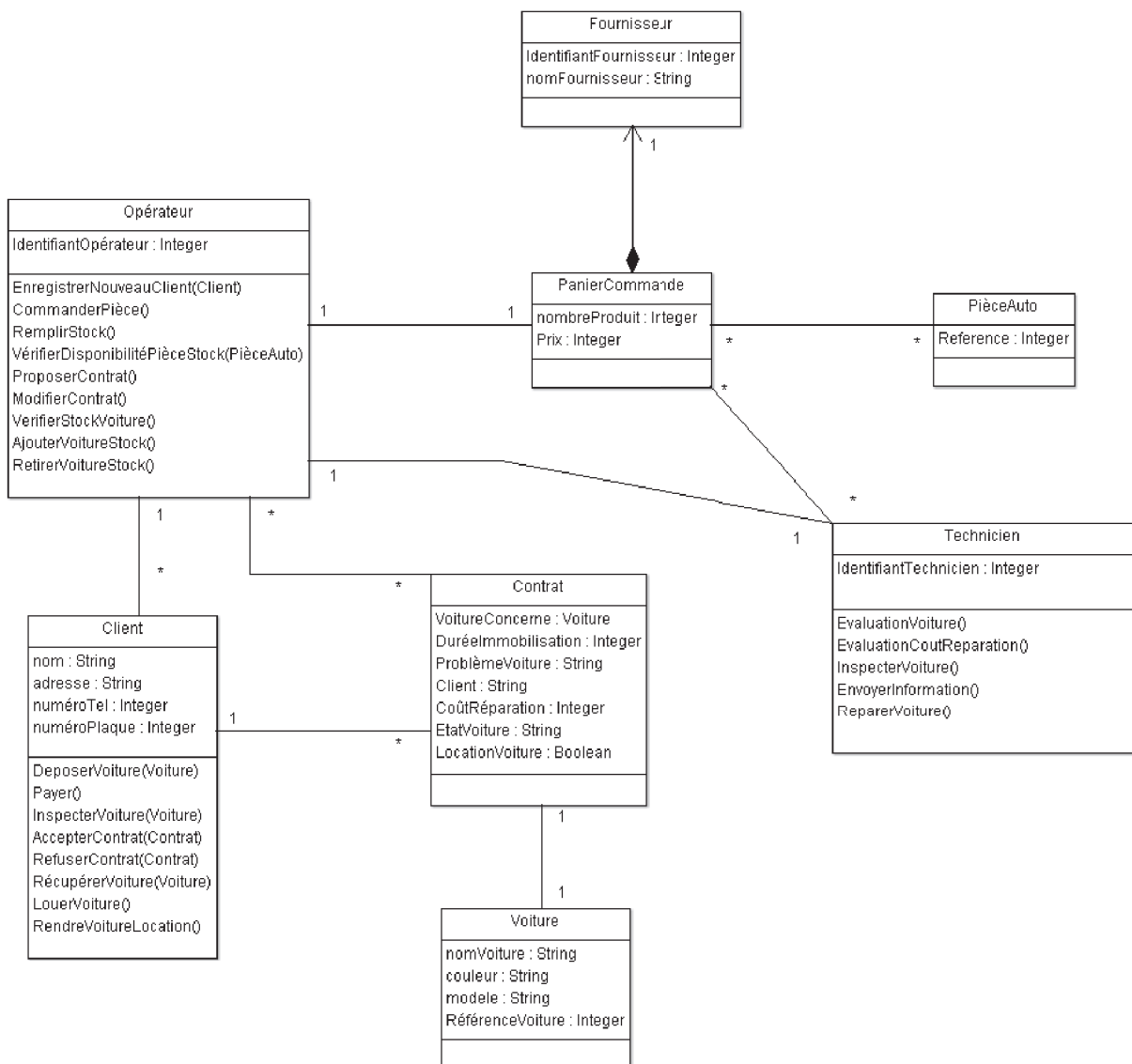


B. Nouveau diagramme de classe

Une fois notre nouveau diagramme de cas d'utilisation réalisé, nous avons modifié notre diagramme de classe. Nous avons seulement effectué des ajouts de méthodes et d'un attribut qui sont :

- l'attribut LocationVoiture qui est un booléen et qui est dans la classe Contrat,
- les méthodes pour le client : LouerVoiture(), RendreVoitureLocation(),
- les méthodes pour l'opérateur : VérifierStockVoiture(), AjouterStockVoiture(), RetirerStockVoiture().

Voici notre diagramme de classe modifié :



C. Diagramme d'activité du système.

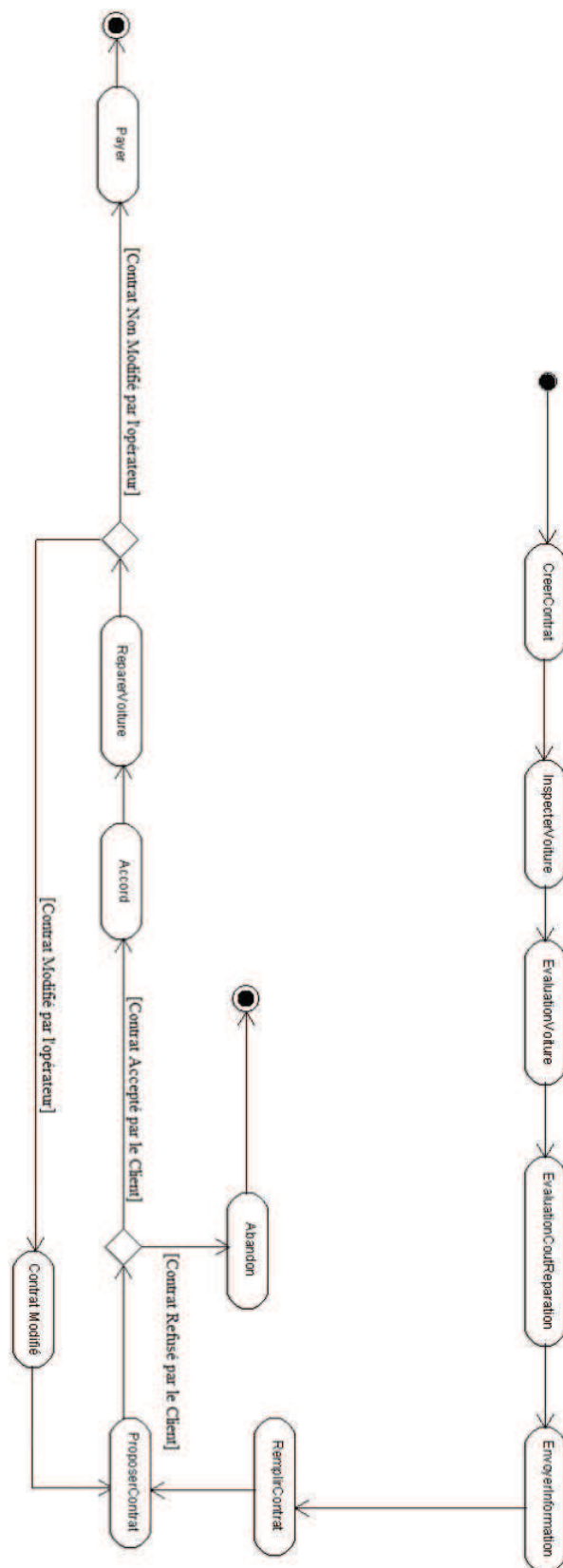
Pour l'élaboration du diagramme d'activité de notre système, nous nous sommes basés sur le diagramme de cas d'utilisation. Le cheminement des actions y est clair. Nous pouvons voir que l'on crée un contrat avant de faire toutes les autres tâches. Les réparations ne peuvent se faire uniquement si le client a accepté le contrat proposé. Si le contrat n'est pas accepté, le client récupère sa voiture sans qu'aucune réparation n'ait été effectuée.

Avant que le contrat soit proposé, certaines informations doivent être récoltées et envoyées pour son établissement.

Si les réparations sont effectuées, le client doit payer et il pourra récupérer sa voiture.

Le contrat peut être modifié à n'importe quel moment durant les réparations donc tant que les réparations ne sont pas terminées, nous avons la possibilité de modifier le contrat.

Voici notre diagramme d'activité :



D. Diagramme d'états-transitions du système.

Pour l'élaboration de notre diagramme d'états-transitions, nous nous sommes basés sur les diagrammes de classe et de séquence.

Nous en avons conclu que seul le contrat et la voiture changeaient d'état dans notre système.

Donc nous avons 5 états pour le Contrat qui sont :

- 0 : Contrat non créé
- 1 : Contrat créé
- 2 : Contrat proposé
- 3 : Contrat accepté
- 4 : Contrat refusé

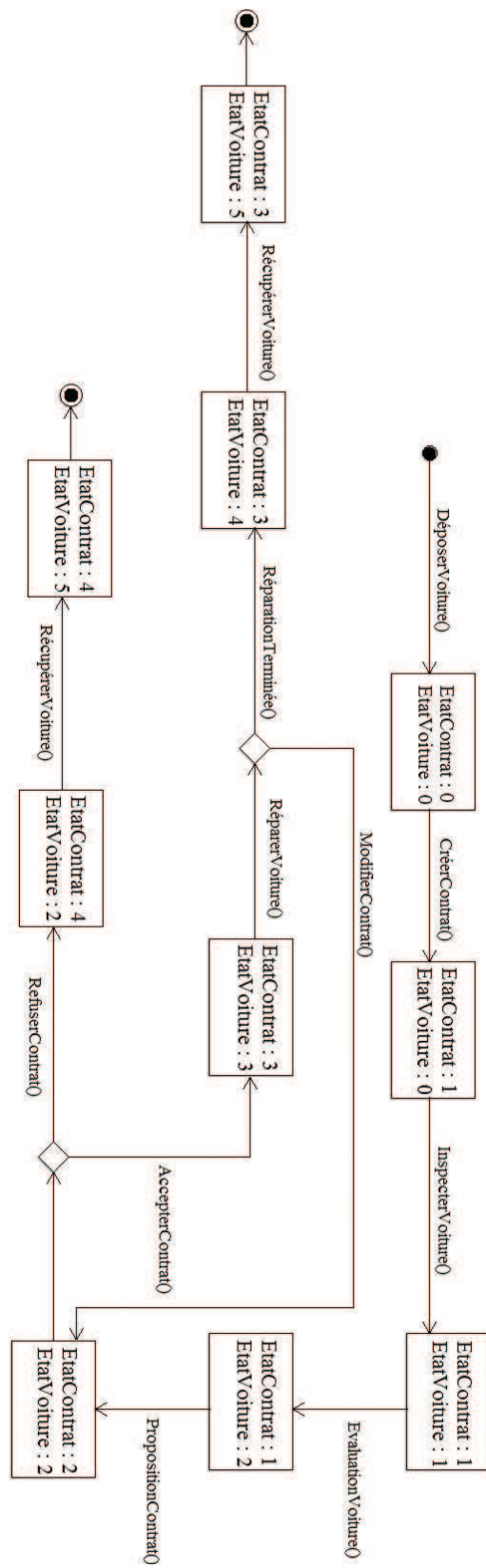
Pour la voiture, nous avons 6 états :

- 0 : Voiture déposée
- 1 : Voiture inspectée
- 2 : Voiture évaluée
- 3 : Voiture en cours de réparation
- 4 : Voiture réparée
- 5 : Voiture récupérée

Les transitions sont les méthodes que nous avons définies dans le diagramme de classe qui sont :

- DeposerVoiture()
- CréerContrat()
- InspecterVoiture()
- EvaluerVoiture()
- ProposerContrat()
- AccepterContrat()
- RefuserContrat()
- ReparerVoiture()
- RecupereVoiture()
- ModifierContrat()

Voici le diagramme d'états-transitions de notre système :



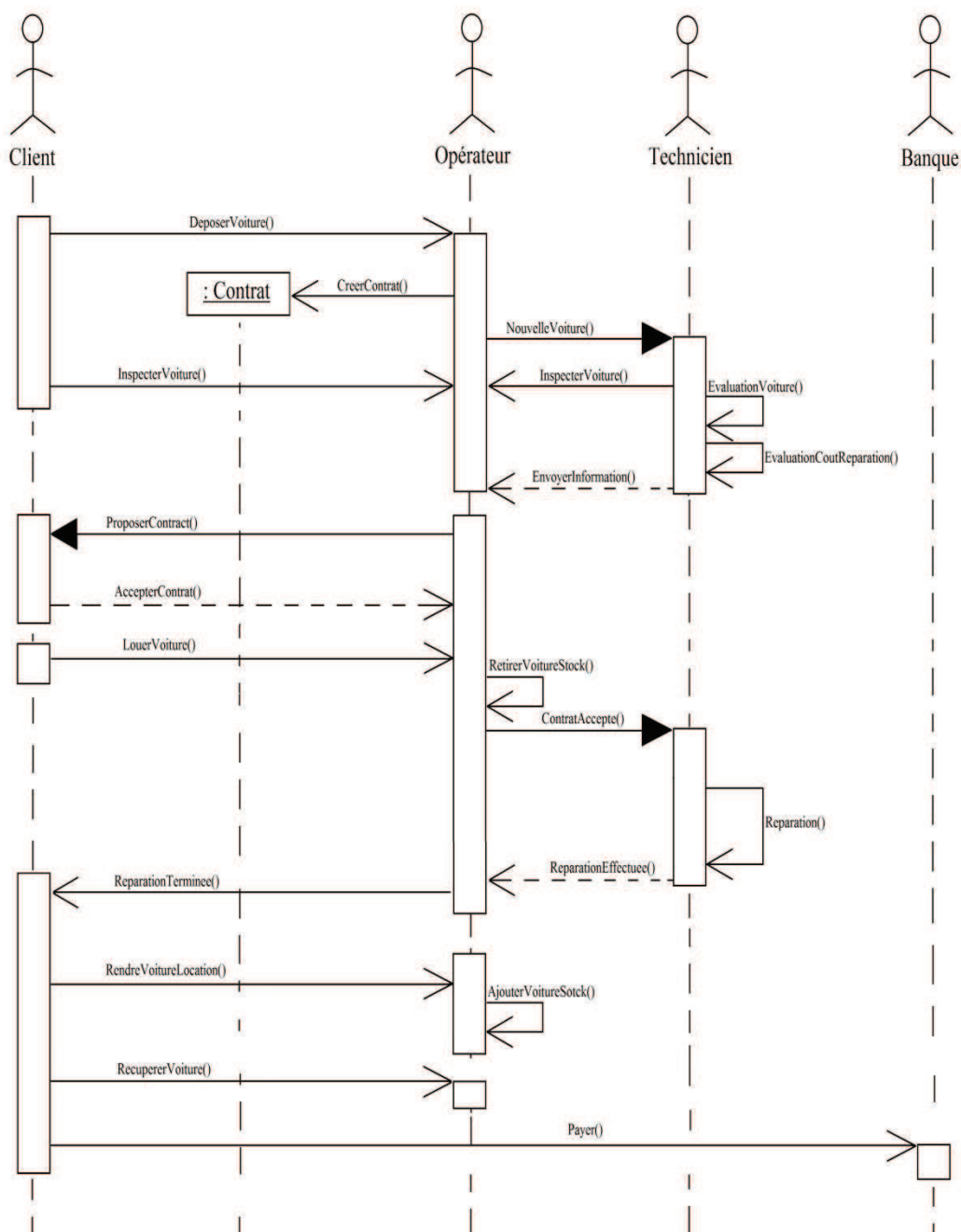
E. Diagramme de séquence d'un scénario classique.

Le scénario est le suivant : un client déjà enregistré vient faire réparer sa voiture ; il accepte le devis, et loue une voiture. Plus tard, il rendra la voiture de location, récupèrera sa voiture et règlera.

Selon notre diagramme de classe et celui de cas d'utilisation, nous avons à partir d'un scénario basique réalisé un diagramme de séquence.

Celui nous permet de voir le déroulement de notre système. Donc c'est la chronologie des événements de notre système dans ce scénario.

Voici le diagramme de séquence correspondant :

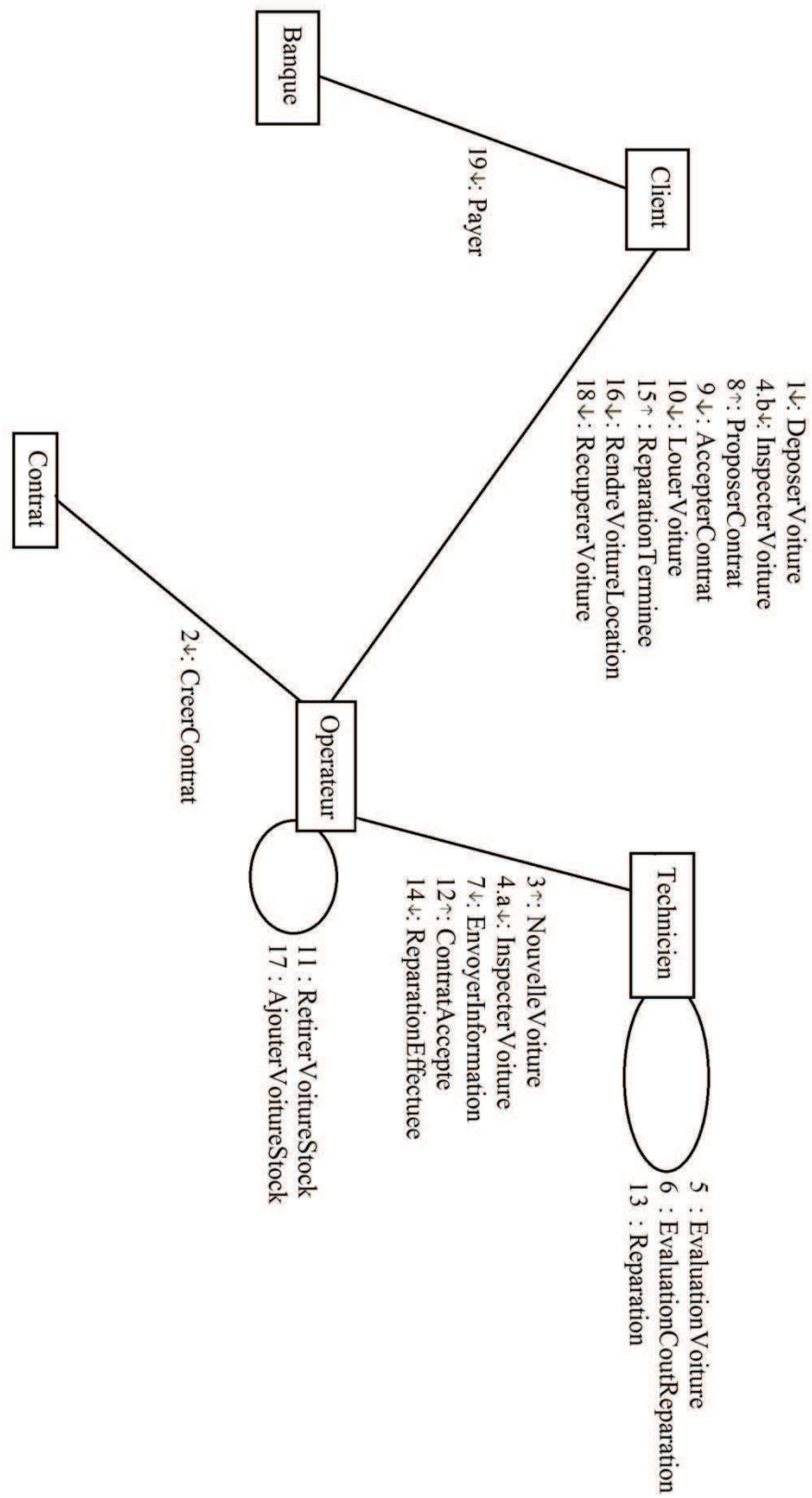


F. Diagramme de communication d'un scénario classique.

Le scénario est le suivant : un client déjà enregistré vient faire réparer sa voiture ; il accepte le devis, et loue une voiture. Plus tard, il rendra la voiture de location, récupèrera sa voiture et règlera.

Le diagramme de communication comporte les mêmes informations que le diagramme de séquence. Le scénario demeure le même donc nous avons une autre représentation de la chronologie de notre système pour ce scénario.

Voici notre diagramme de communication correspondant :



Conclusion

La réalisation de ce projet fut assez instructive, de par le fait que nous effectuions nos propres choix et que la plupart de nos choix peuvent correspondre avec ce que le client demandait.

Il faut seulement savoir bien expliqué nos choix et bien analyser le sujet avant de commencer. Le fait d'avoir changé le sujet après le premier rendu n'est pas une mauvaise chose.

Cependant, nous avions déjà réalisés la suite de notre projet donc nous avons dû adapter nos diagrammes avec ce que le client demandait en plus. Nous avons pu nous apercevoir que nos diagrammes étaient bien « pensés » puisque nous n'avons pas eu besoin de changer tous nos diagrammes mais seulement rajouter les actions supplémentaires que le client souhaite pouvoir faire à l'aide de son système.