

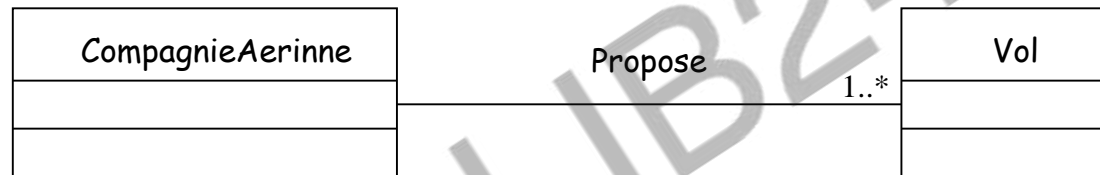
Soit le cas *"Réservation de vols dans une agence de voyage"*

- 1° Des compagnies aériennes proposent différents vols.
- 2° Un vol est ouvert à la réservation et fermé sur ordre de la compagnie.
- 3° Un client peut réserver un ou plusieurs vols, pour des passagers différents.
- 4° Une réservation concerne un seul vol, et un seul passager.
- 5° Une réservation peut être annulée ou confirmée.
- 6° Un vol a un aéroport de départ et un aéroport d'arrivée.
- 7° Un vol a un jour et une heure de départ et un jour et une heure d'arrivée.
- 8° Un vol peut comporter des escales dans des aéroports
- 9° Une escale a une heure d'arrivée et une heure de départ.
- 10° Chaque aéroport dessert une ou plusieurs villes

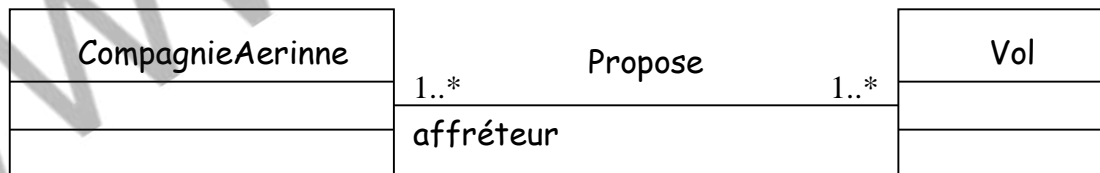
➤ Modélisation de la phrase :

1° Des compagnies aériennes proposent différents vols.

CompagnieAerienne et *Vols* sont 2 objets métiers : 2 classes

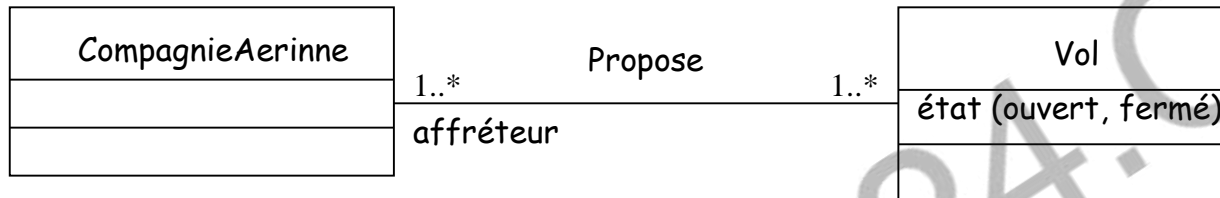


- Un vol est réalisé par une seule compagnie mais partagé par plusieurs affréteurs

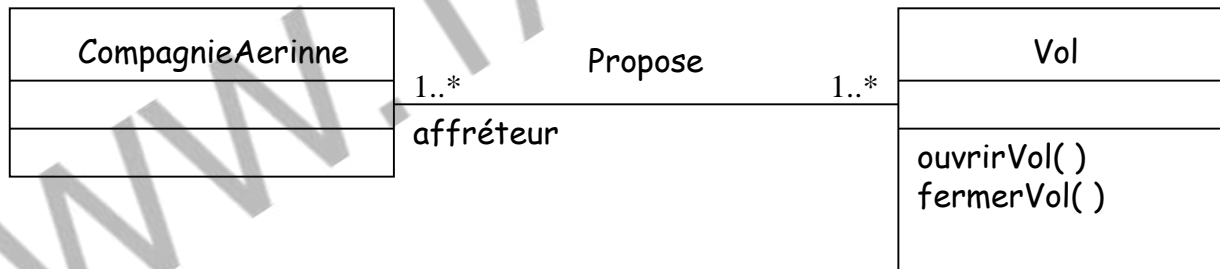


➤ Modélisation de la phrase :

2° Un vol est ouvert à la réservation et fermé sur ordre de la compagnie.



- Tout objet peut avoir un état (diagramme d'états).
- Dans un diagramme de classes tout concept dynamique est modélisé en opération.
- Il faut représenter la 2° phrase par 2 opérations : **ouvrirReservation()** et **fermerReservation()**
- Dans quelle classe ? Responsabilité d'une classe

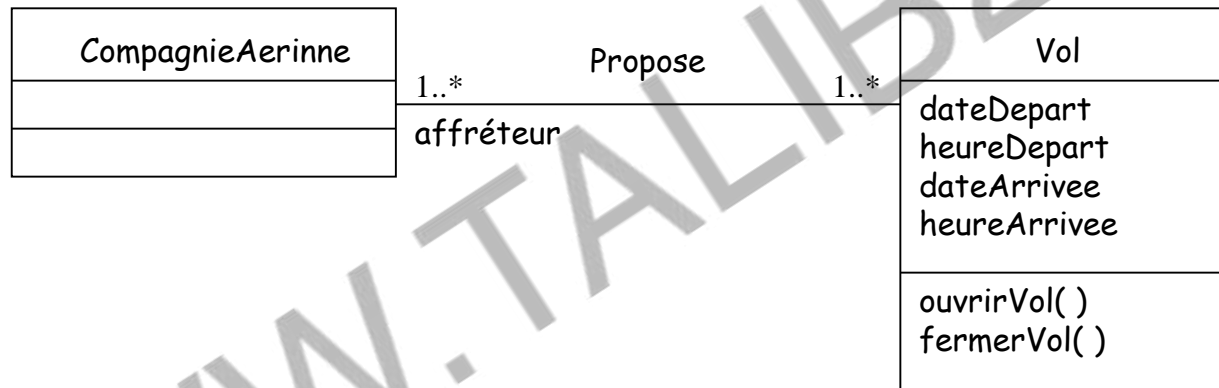


- Les opérations sont déclarées dans l'objet dans lequel elles doivent s'exécuter
- Les autres pourront déclencher ces opérations par envoi de messages
- Le classe CompagnieAerienne a une association avec la classe vol.

➤ Modélisation des phrases :

7° Un vol a un jour et une heure de départ et un jour et une heure d'arrivée.

➤ Les dates et les heures de départ et d'arrivée ne représentent que des valeurs : attributs.



➤ Pour savoir si un élément doit être représenté en attribut ou en objet :

➤ S'il n'y a que sa valeur qui est intéressante : c'est plutôt un attribut.

➤ Si plusieurs questions peuvent concerner l'élément, alors il faut le représenter en objet.

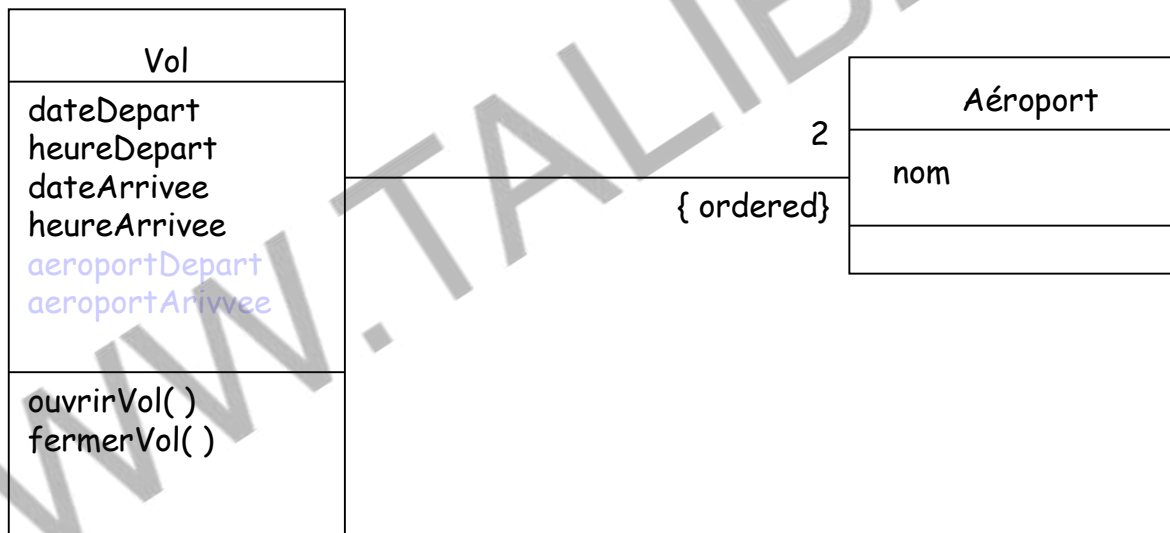
➤ Modélisation des phrases :

6° Un vol a un aéroport de départ et un aéroport d'arrivée.

➤ Par quoi peut-on représenter l'élément "Aéroport" ?

3 réponses sont envisageables :

1. Soit avec une classe et une association de multiplicité 2

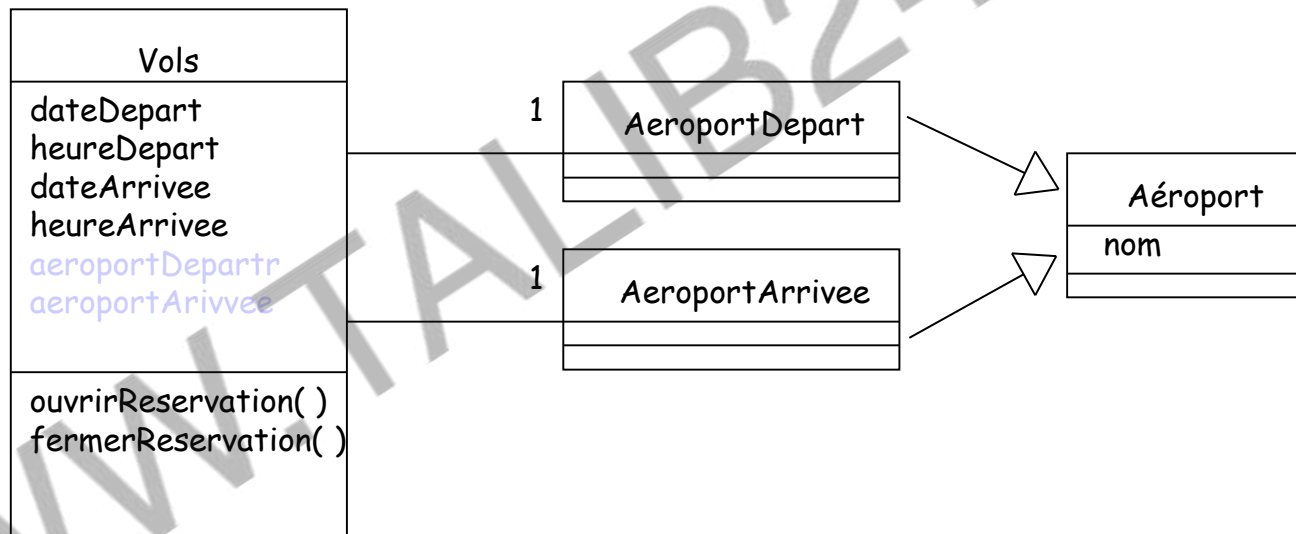


Modélisation peu parlante.

➤ Modélisation des phrases :

6° Un vol a un aéroport de départ et un aéroport d'arrivée.

2. Soit avec 2 classes

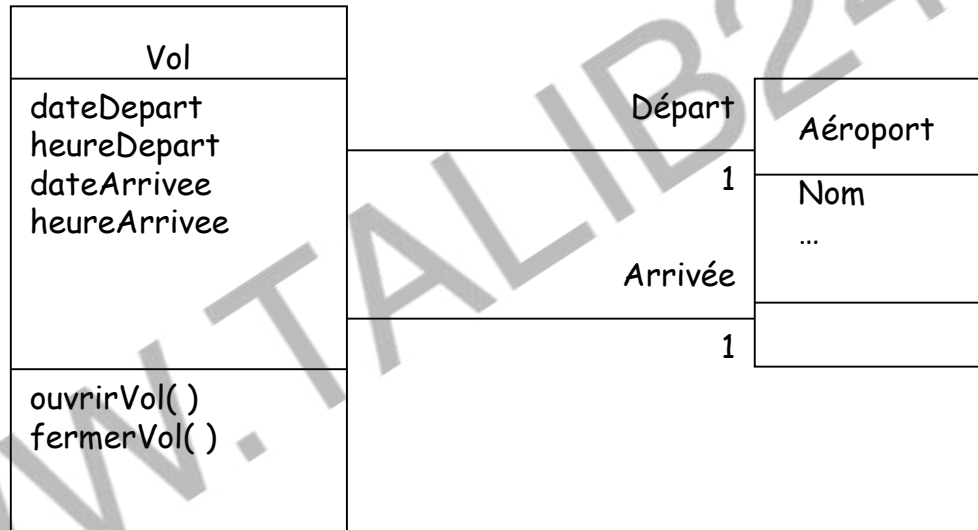


Modélisation non correcte. Tout aéroport peut être de départ et d'arrivée.

➤ Modélisation des phrases :

6° Un vol a un aéroport de départ et un aéroport d'arrivée.

2. Soit avec 2 associations



Le rôle de chaque association précise son sens.

➤ Modélisation des phrases :

10° Chaque aéroport dessert une ou plusieurs villes

➤ On ne peut pas savoir la multiplicité de "Aéroport"



➤ Si on considère que desservir une ville signifie l'aéroport le plus proche, il n'y en a qu'un :
la multiplicité est de 1

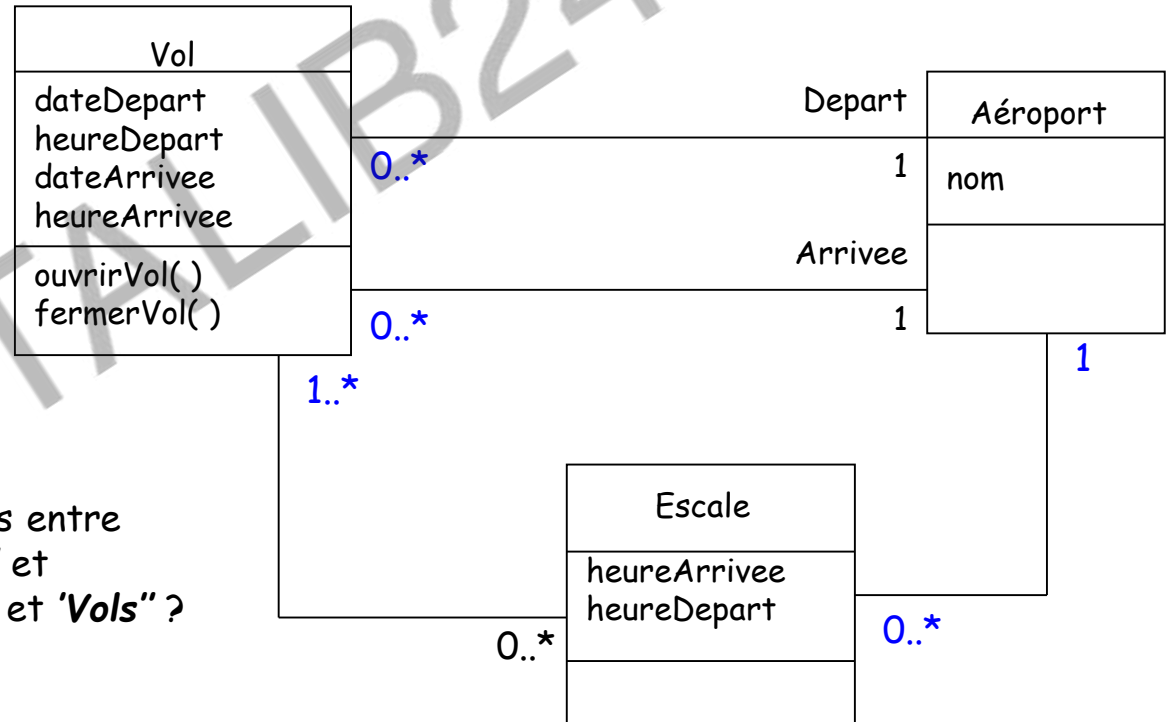
➤ Si on considère que desservir une ville signifie les aéroports dans un rayon de 35 km :
la multiplicité est de 0..*

➤ Modélisation des phrases :

8° Un vol peut comporter des escales dans des aéroports

9° Une escale a une heure d'arrivée et une heure de départ.

➤ Une escale a les propriétés heure d'arrivée et heure de départ, c'est donc un objet.



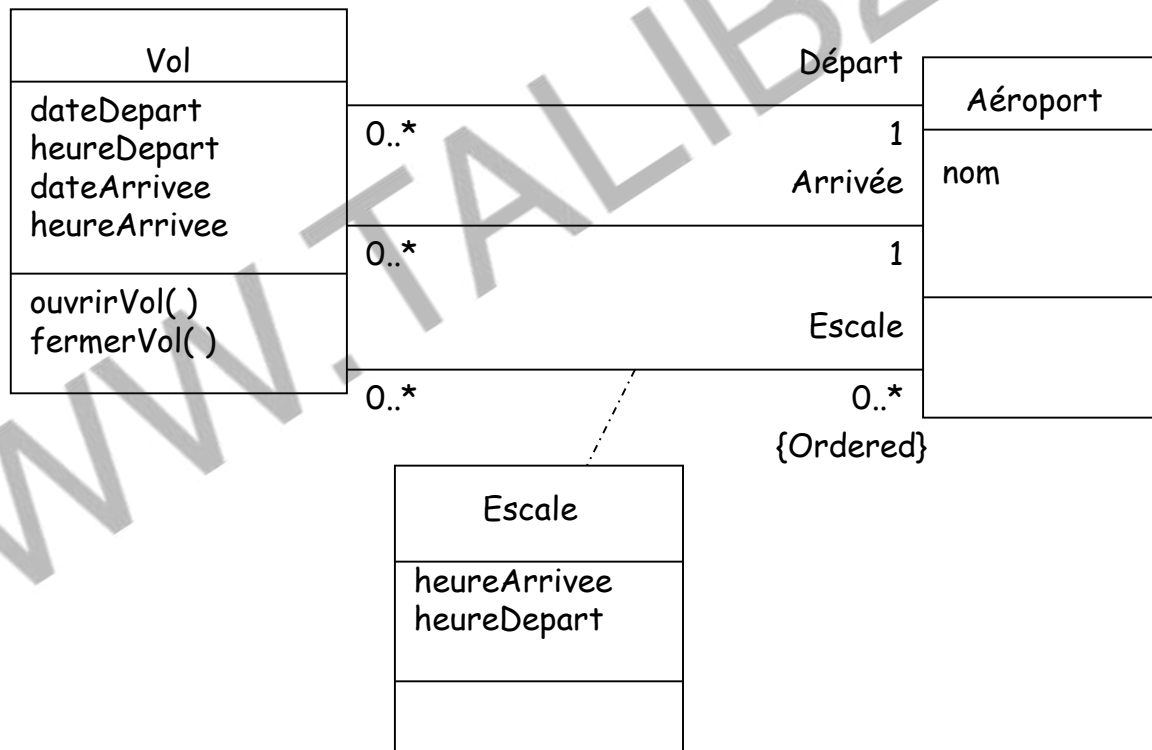
➤ Quelles sont alors les multiplicités entre "Vols" et "Escale", entre "Escale" et "Aéroport" et entre "Aéroport" et "Vols" ?

➤ **Modélisation des phrases :**

8° Un vol peut comporter des escales dans des aéroports

9° Une escale a une heure d'arrivée et une heure de départ.

- **"Escale"** a peu d'informations propres. Elle n'est qu'une partie de **"Vol"**.
- On peut la représenter comme une spécialisation de **"Aéroport"**. Mais elle n'est pas totalement un aéroport
- La meilleure solution serait de la modéliser comme une classe d'association entre **"Vols"** et **"Aéroport"**.

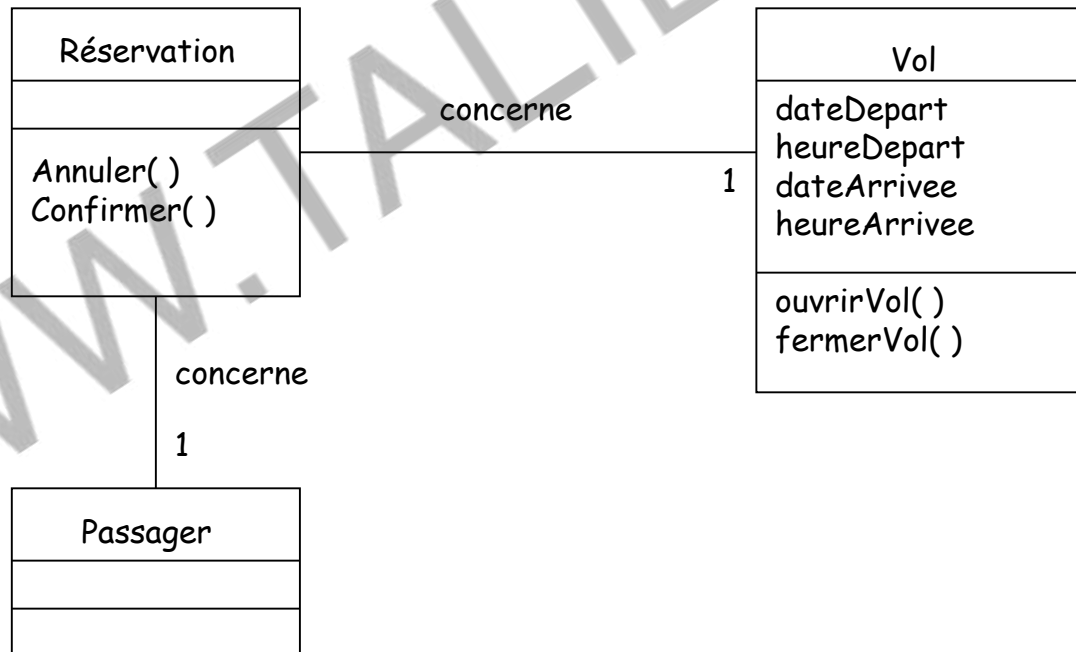


➤ Modélisation des phrases :

4° Une réservation concerne un seul vol, et un seul passager.

5° Une réservation peut être annulée ou confirmée.

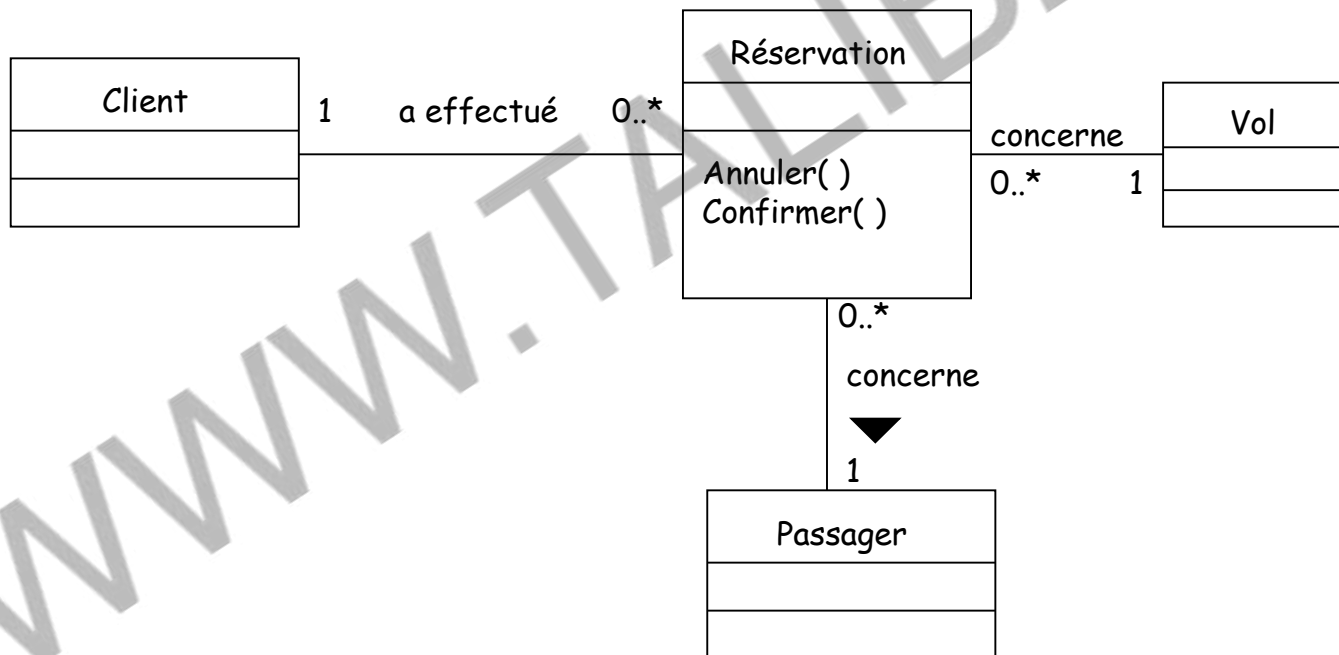
- La réservation et le passager sont 2 concepts métier : 2 classes d'objets
- Un réservation concerne un seul vol et un seul passager: donc 2 associations entre "Vol" et "Réservation" et entre "Réservation" et "Passager".
- La 5° phrase se traduit par l'ajout de 2 opérations *annuler()* et *confirmer()* dans "Reservation".



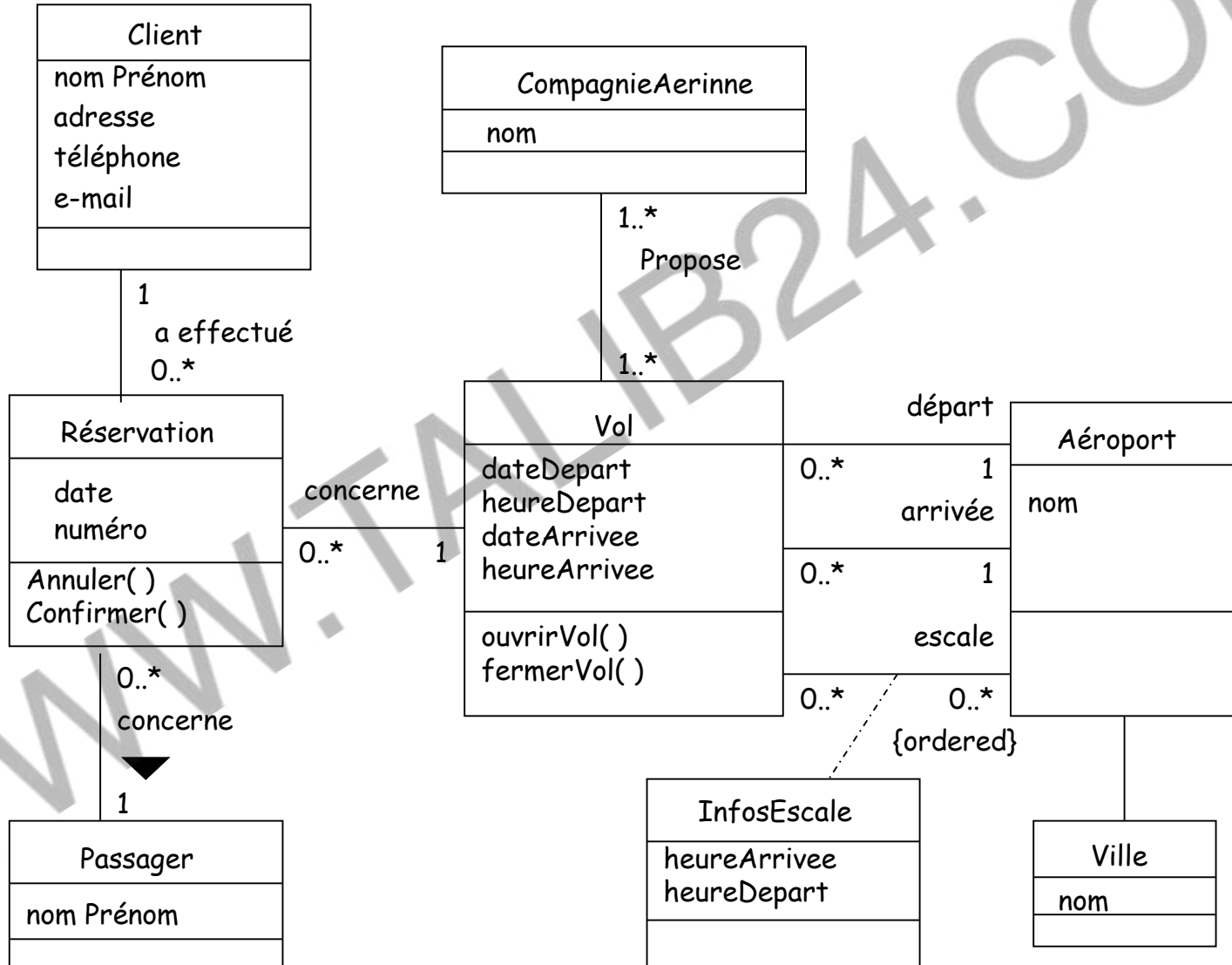
➤ Modélisation des phrases :

3° Un client peut réserver un ou plusieurs vols, pour des passagers différents.

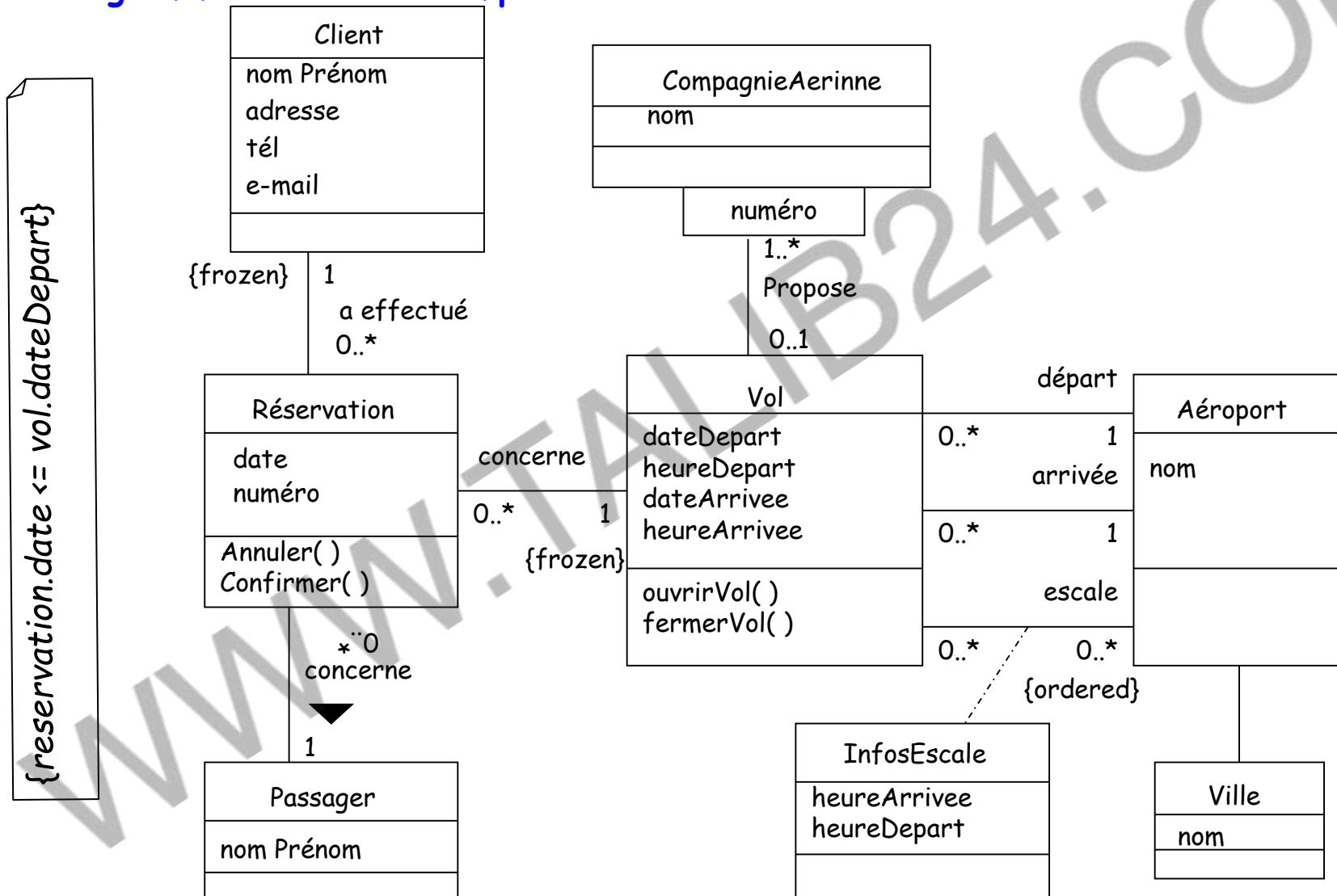
➤ Il faut discerner un client d'un passager



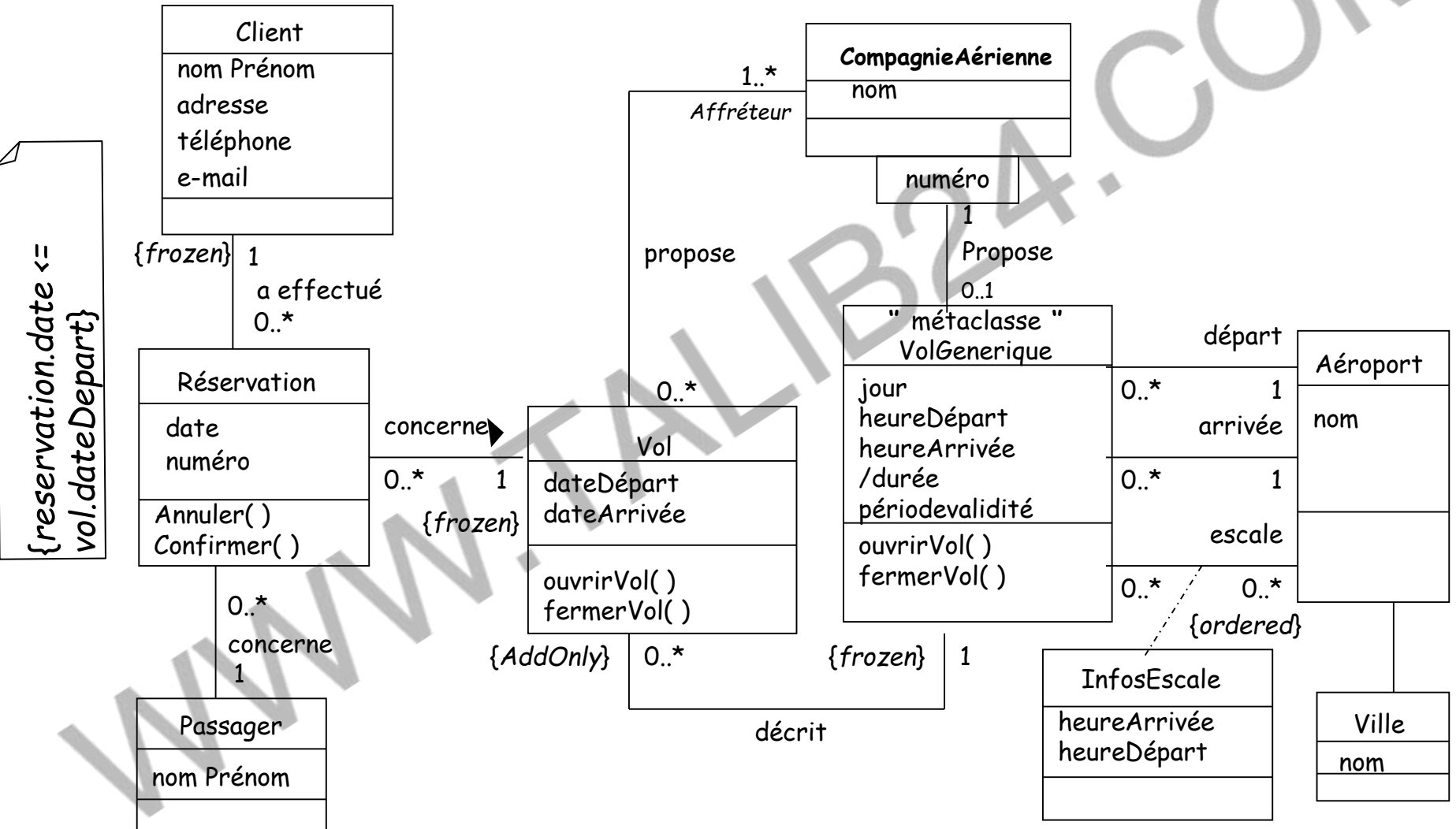
➤ Le diagramme des classe complet est :



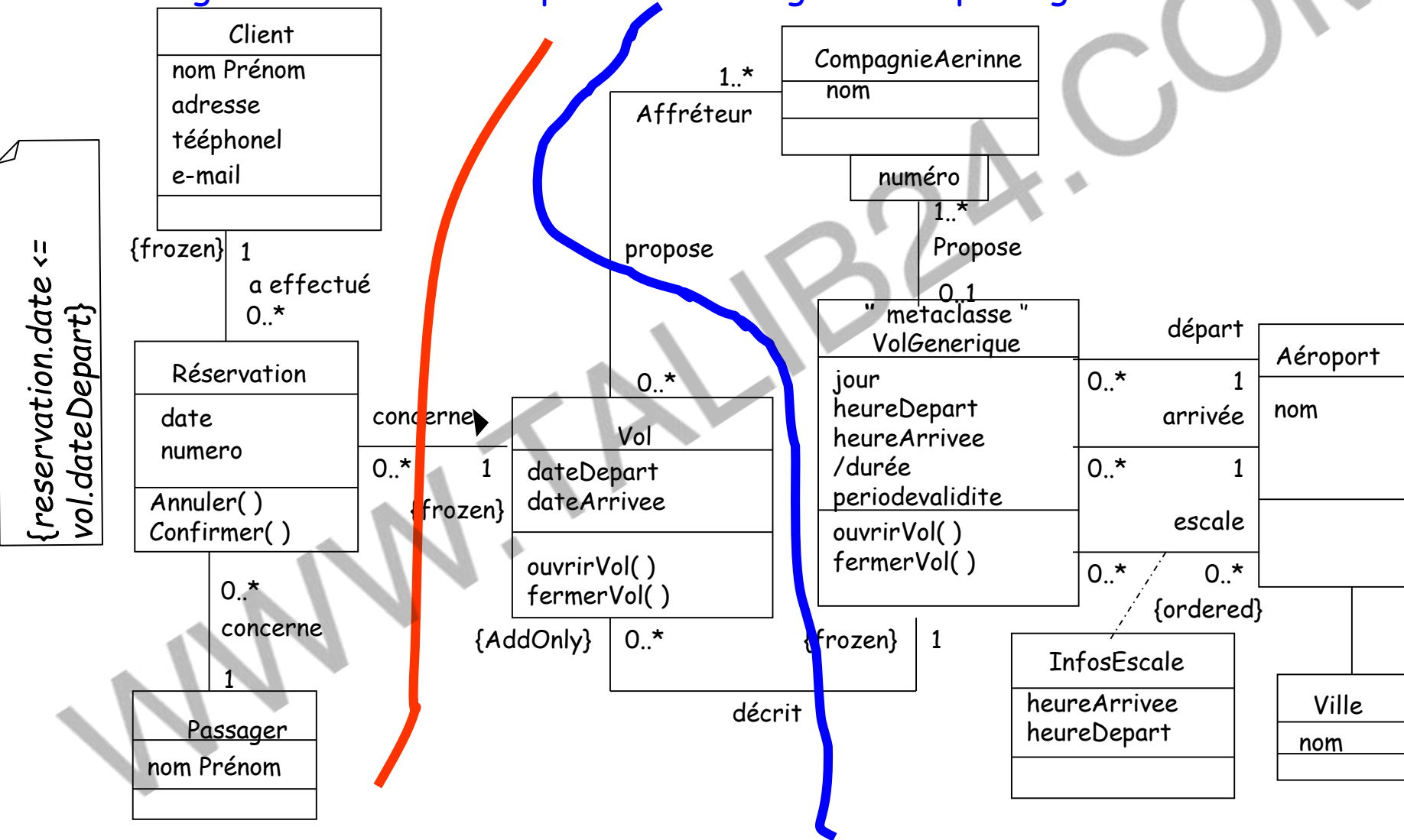
➤ Diagramme des classe complet et annoté

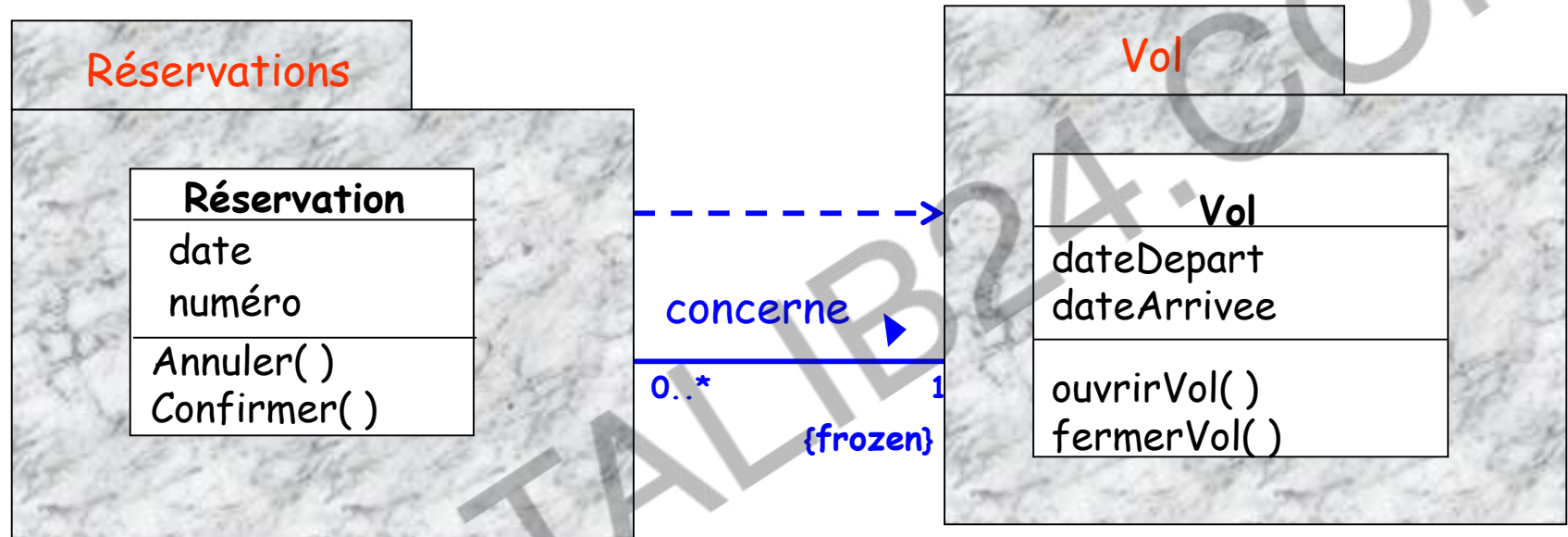


➤ Le diagramme des classe complet devient :



➤ Le diagramme des classes peut être réorganisé en packages

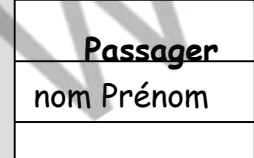
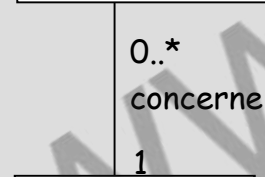
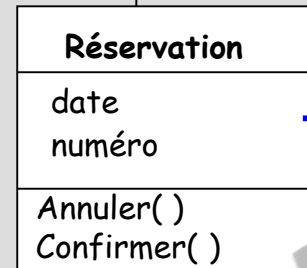
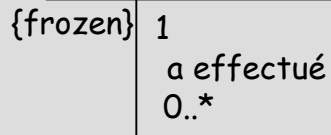
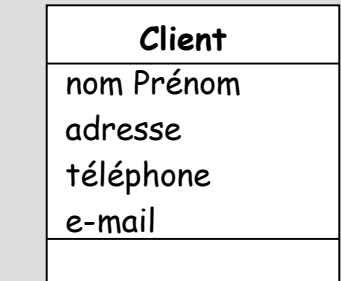




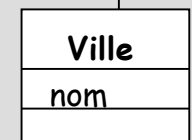
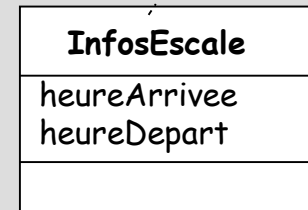
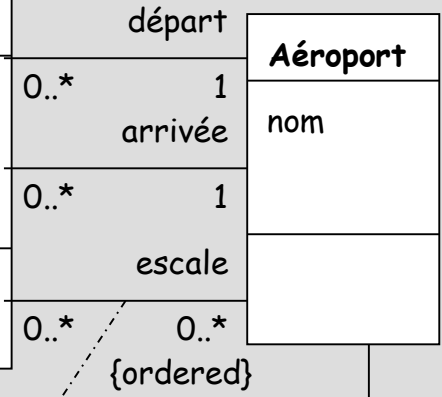
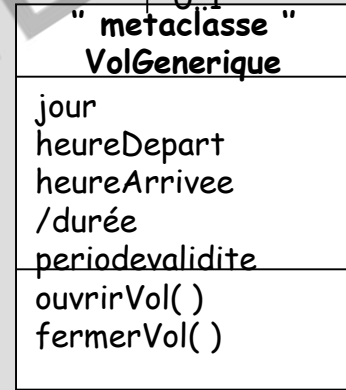
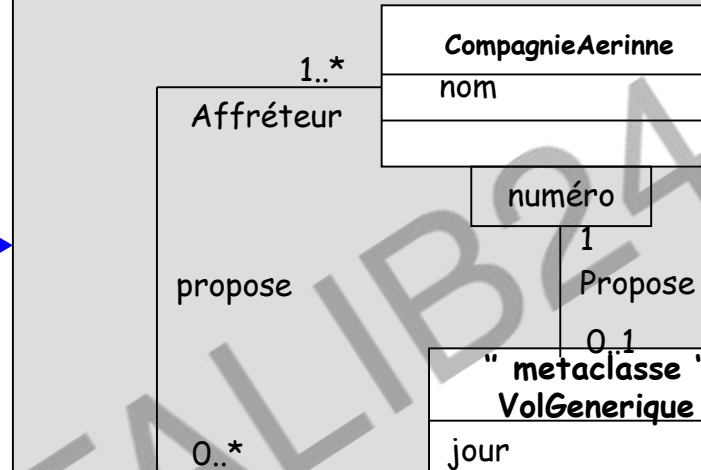
- Réduire la dépendance mutuelle afin d'augmenter la modularité et l'évolutivité d'une application

Réservations

Vol

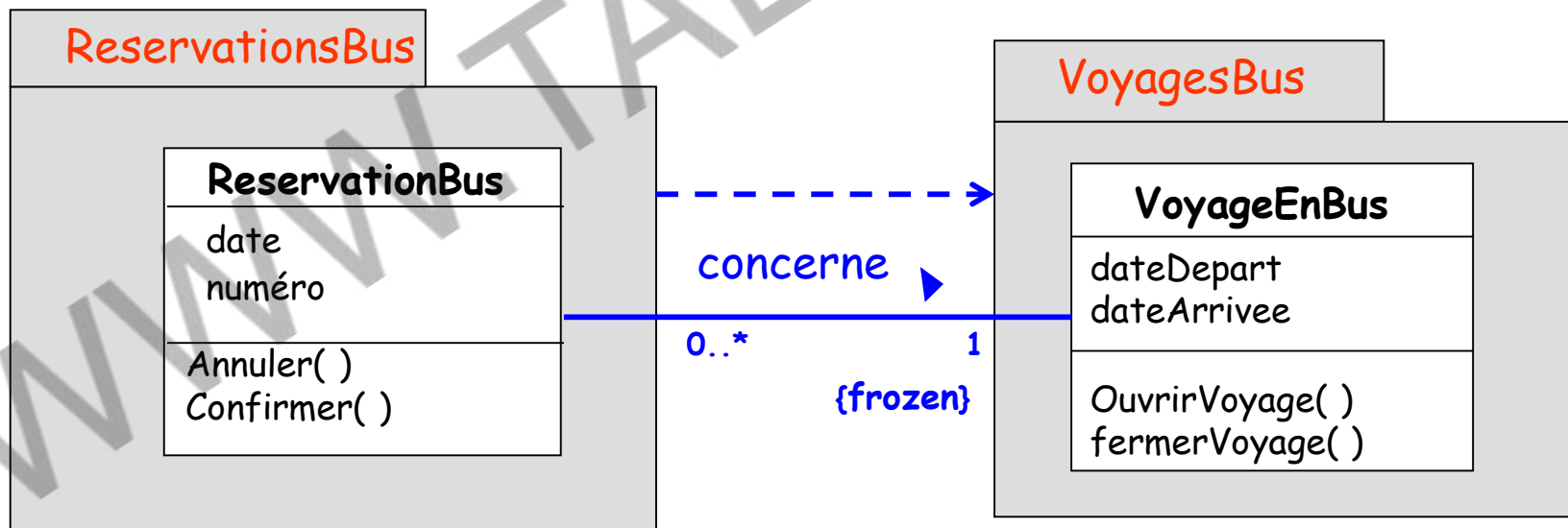


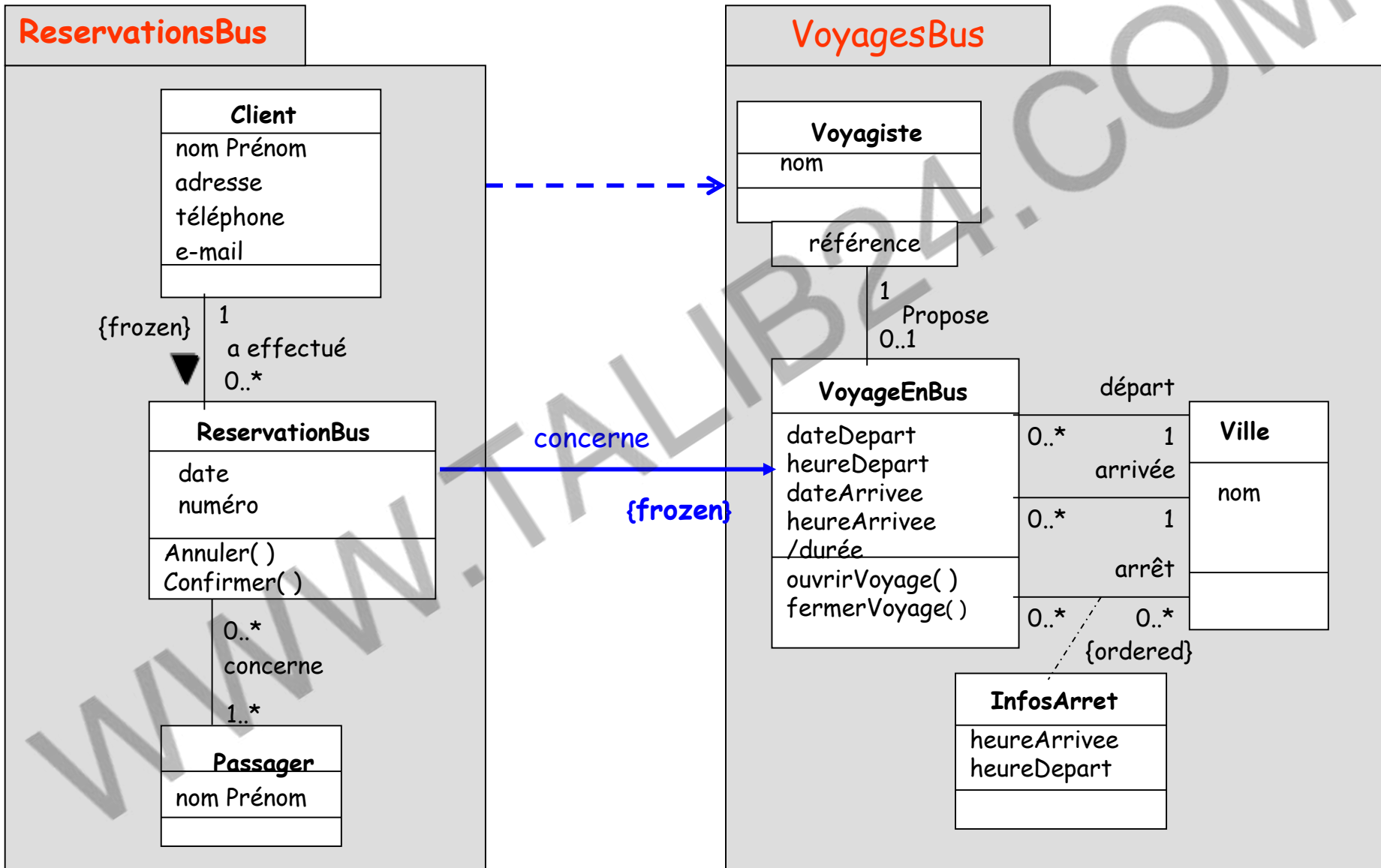
concerne



Généralisation et réutilisation

- On veut élargir ce domaine aux voyages par bus que des transporteurs assurent.
- Un voyage en bus à une ville de départ et un ville d'arrivée avec des dates et des heures associées.
- Un trajet peut comporter des arrêts dans des villes intermédiaires.
- Un client peut réserver un ou plusieurs voyages pour un ou plusieurs passagers

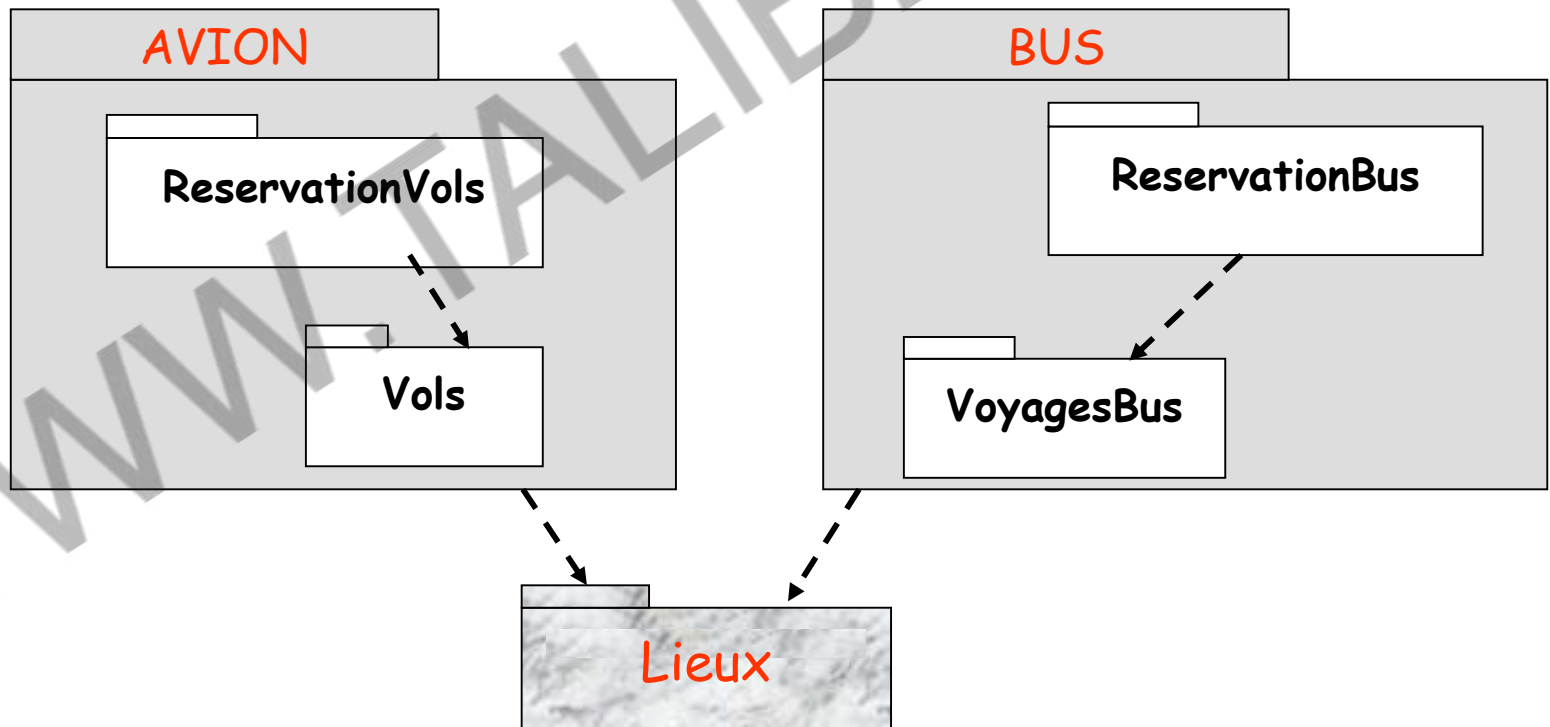




Fusion des 2 modèles

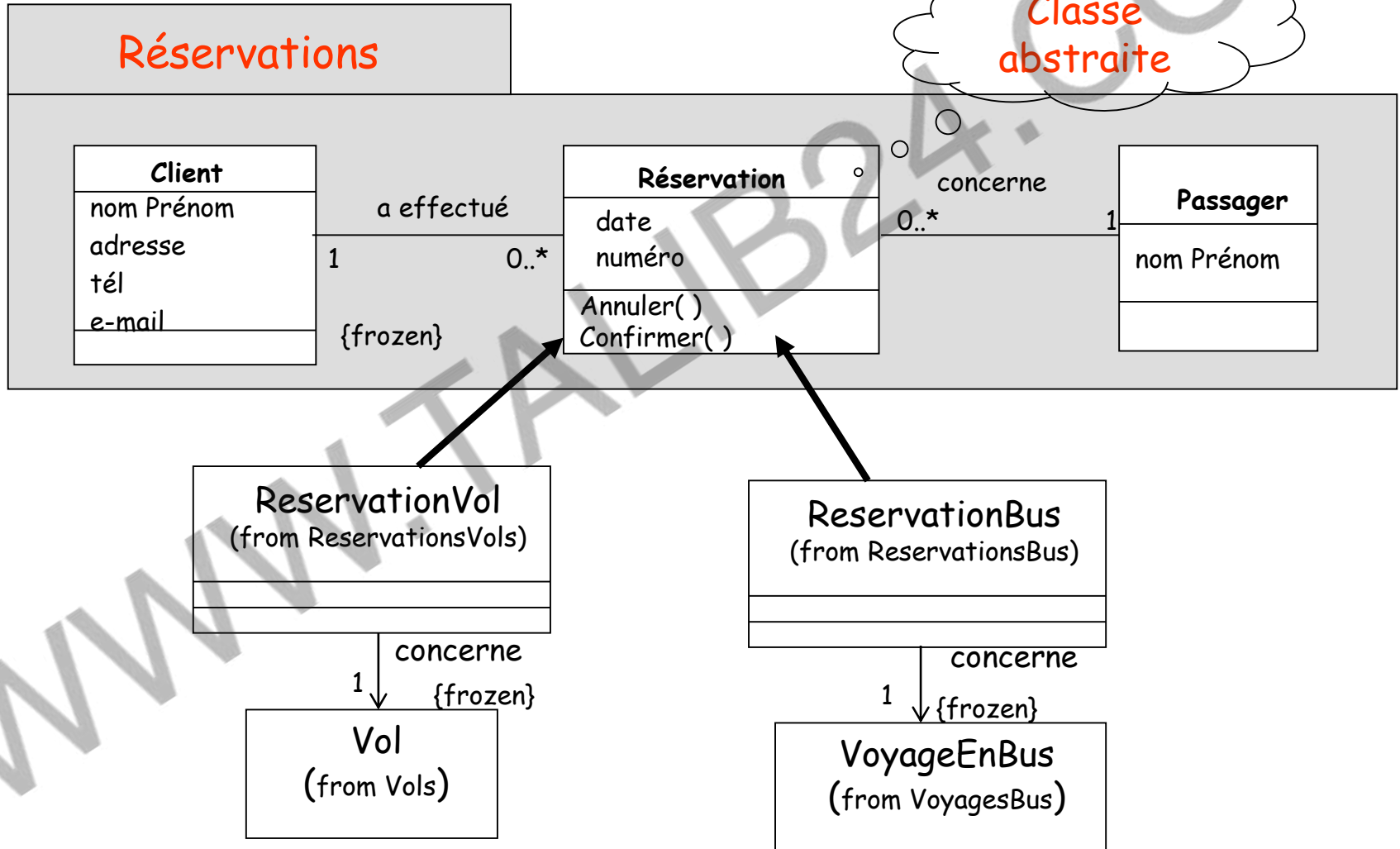
1. Il faut isoler les classes communes dans des packages
2. Il faut factoriser les propriétés communes

1



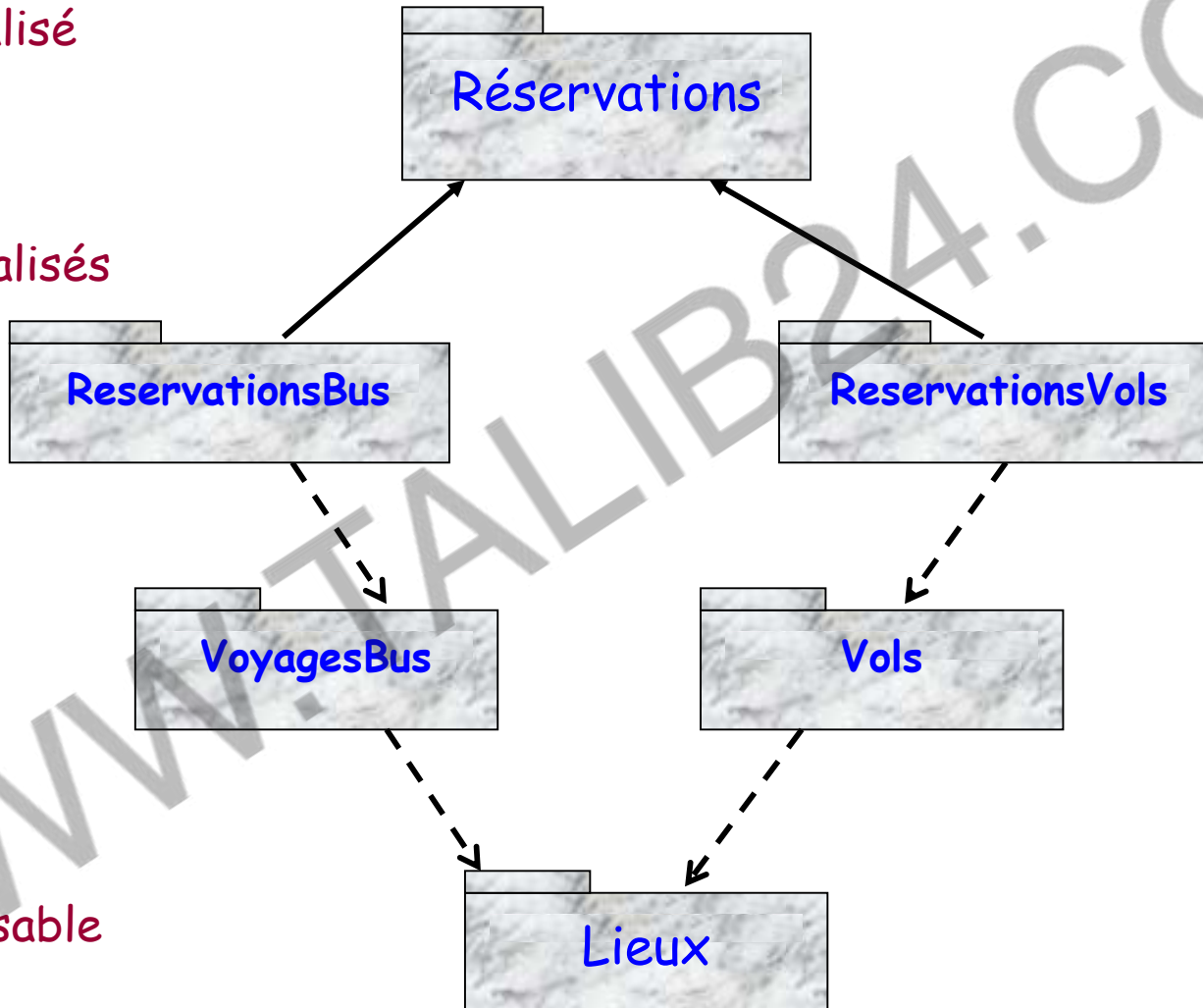
2

Il faut isoler les classes communes dans des packages



Package généralisé

Packages spécialisés



Package réutilisable

TD1 : Diagramme de cas d'utilisation UML

1 Identification des acteurs et de cas d'utilisation simples

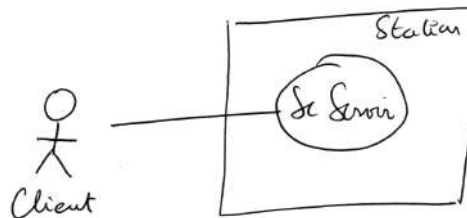
Considérons une station-service de distribution d'essence. Les clients se servent de l'essence et le pompiste remplit les cuves.

Question : Le client se sert de l'essence de la façon suivante : il prend un pistolet accroché à une pompe et appuie sur la gâchette pour prendre de l'essence. Qui est l'acteur du système? Est-ce le client, le pistolet ou la gâchette?

C'est le client. Un acteur est toujours extérieur au système. Définir les acteurs d'un système, c'est aussi en définir les bornes.

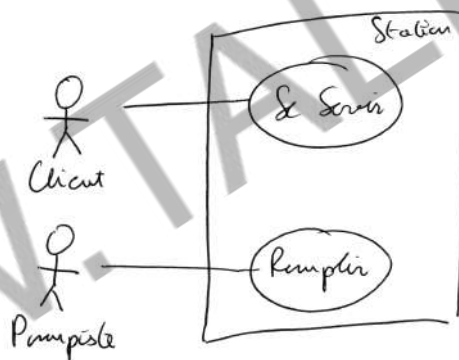
Question : Jojo, dont le métier est pompiste, peut se servir de l'essence pour sa voiture. Pour modéliser cette activité de Jojo, doit-on définir un nouvel acteur? Comment modélise-t-on ça?

Jojo est ici considéré comme un client. Pour définir les acteurs, il faut raisonner en termes de rôles.



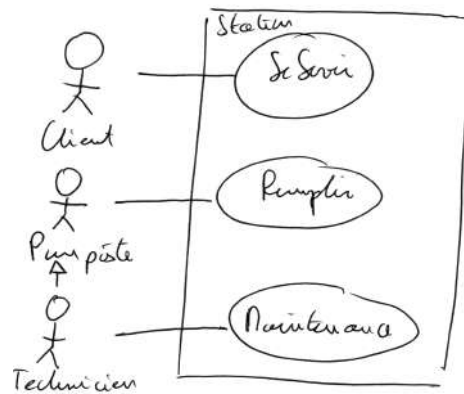
Question : Lorsque Jojo vient avec son camion citerne pour remplir les réservoirs des pompes, est-il considéré comme un nouvel acteur? Comment modélise-t-on cela?

Jojo est ici considéré comme pompiste.



Question : Certains pompistes sont aussi qualifiés pour opérer des opérations de maintenance en plus des opérations habituelles des pompistes telles que le remplissage des réservoirs. Ils sont donc réparateurs en plus d'être pompistes. Comment modéliser cela?

La seule relation possible entre deux acteurs est la généralisation. Elle permet de spécifier des usages particuliers.



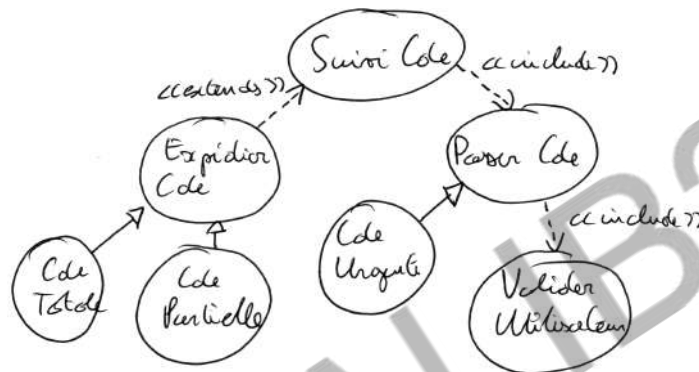
2 Relations entre cas d'utilisation

Soient les cas d'utilisation suivants :

- Passer une commande
- Passer une commande urgente
- Suivre une commande
- Valider l'utilisateur
- Passer une commande
- Expédier commande totale ou partielle

Le suivi de la commande désigne le processus complet, du passage à l'expédition. Il peut toutefois arriver qu'une commande passée ne soit pas envoyée. Passer une commande urgente est un cas particulier de passer une commande. Pour passer une commande, il faut nécessairement valider l'utilisateur.

Question : Donner le diagramme de cas d'utilisation sans représenter les acteurs

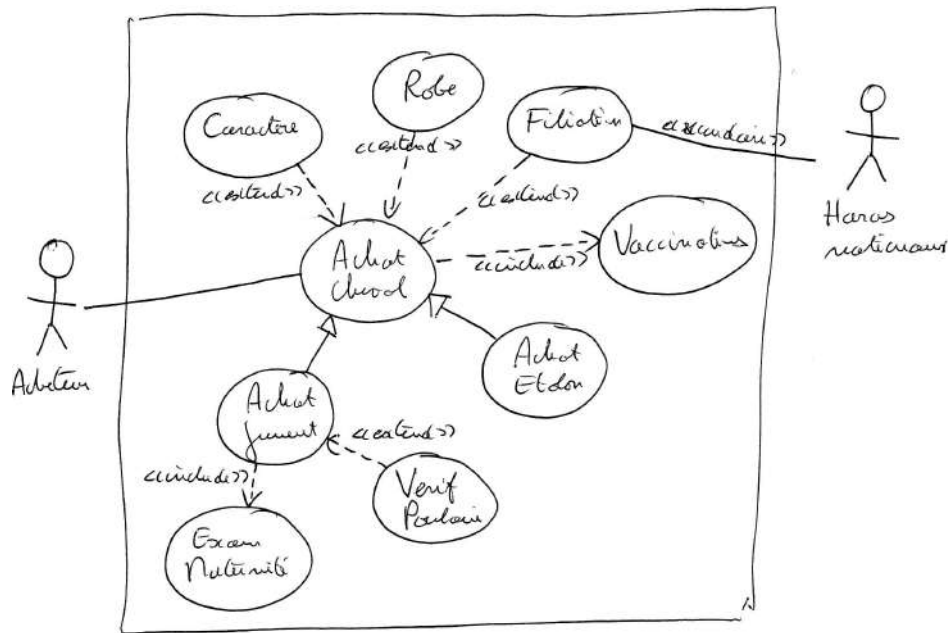


3 Spécification des besoins avec un diagramme de cas

3.1 Préparation de l'achat de chevaux

Un système informatique doit permettre à des acheteurs potentiels de préparer l'achat de chevaux (mais pas l'achat proprement dit). L'achat d'un cheval concerne soit une jument soit un étalon. Dans le premier cas, on doit impérativement examiner l'état de maternité du cheval, et éventuellement vérifier que la jument n'a pas un jeune poulain en ce moment. Que l'on souhaite acheter un étalon ou une jument, on doit effectuer un examen des vaccinations. En outre l'acheteur peut souhaiter, lors de la préparation de son achat, consulter le caractère du cheval ou bien en connaître la robe. Toutes les informations en rapport à la filiation d'un cheval sont obtenues en consultant la base de données externe des haras nationaux.

Question : Donnez un diagramme de cas d'utilisation pour le système de préparation avant achat.



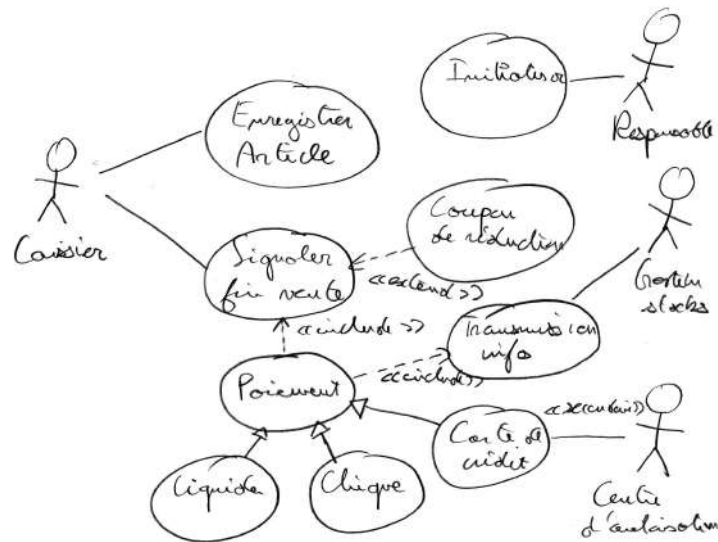
Question : Donnez une description textuelle d'un cas d'utilisation qui vous sera désigné en TD.

3.2 Caisse enregistreuse

Le déroulement normal d'utilisation d'une caisse enregistreuse est le suivant :

1. Un client arrive à la caisse avec des articles
2. Le caissier enregistre le numéro d'identification de chaque article, ainsi que la quantité si celle-ci est supérieure à 1
3. Un client arrive à la caisse avec des articles
4. Le caissier enregistre le numéro d'identification de chaque article, ainsi que la quantité si celle-ci est supérieure à 1
5. La caisse affiche le prix de chaque article et son libellé
6. Lorsque tous les articles ont été enregistrés, le caissier signale la fin de la vente
7. La caisse affiche le total des achats
8. Le client choisit son mode de paiement :
 - Liquide : le caissier encaisse l'argent et la caisse indique le montant éventuel à rendre au client
 - Chèque : le caissier note l'identité du client et la caisse enregistre le montant sur le chèque
 - Carte de crédit : un terminal bancaire fait partie de la caisse, il transmet la demande à un centre d'autorisation multi-banques
9. La caisse enregistre la vente et imprime un ticket
10. Le caissier transmet le ticket imprimé au client
11. Un client peut présenter des coupons de réduction avant le paiement. Lorsque le paiement est terminé, la caisse transmet les informations relatives aux articles vendus au système de gestion des stocks. Tous les matins, le responsable du magasin initialise les caisses pour la journée.

Question : Donnez un diagramme de cas d'utilisation pour la caisse enregistreuse.



Dans cet exo, il faut bien comprendre que les acteurs ne sont pas tout ce qui bouge. Ici, par exemple, le client n'est pas un acteur puisqu'il n'interagit directement pas avec la caisse. c'est le caissier qui enregistre les articles. Les cas ne modélisent pas non plus les séquences d'action. le fait que l'on boucle sur les articles et qu'on sorte de là en signalant la fin de l'enregistrement est modélisé simplement par un cas qui pourra se répéter, et un autre qu'on déclenchera quand le moment sera venu. En outre, un diagramme de cas ne doit pas représenter le moindre truc qui se passe. Ici, par exemple, on fait un cas « enregistrer article » pour la saisie du numéro, de la quantité et l'affichage du libellé et du prix.

Question : Donnez une description textuelle d'un cas d'utilisation qui vous sera désigné en TD.

Exemple pour l'enregistrement d'un article.

- Identification :

- Nom du cas : *Enregistrement d'un article*
- Objectif : *Détaille les étapes permettant au caissier d'enregistrer un article.*
- Acteurs : *Caissier*
- Date : *<date du jour>*
- Responsables : *<vous, cher collègue>*
- Version : *1.0*

- Séquencements :

- *Le cas d'utilisation commence lorsqu'un client présente un article au caissier*
- *Pré-condition : Il y a un article à saisir*
- *Enchaînement nominal*
 1. *Le caissier enregistre le numéro de l'article*
 2. *La caisse affiche le libellé et le numéro*
- *Enchaînements alternatifs*
 3. *Après 1, si la quantité est supérieure à 1, le caissier enregistre la quantité*
- *Post-conditions : ...*

UML

2 : ANALYSE FONCTIONNELLE

Diagrammes de cas d'utilisation, de séquence, d'activités Corrigé des exercices

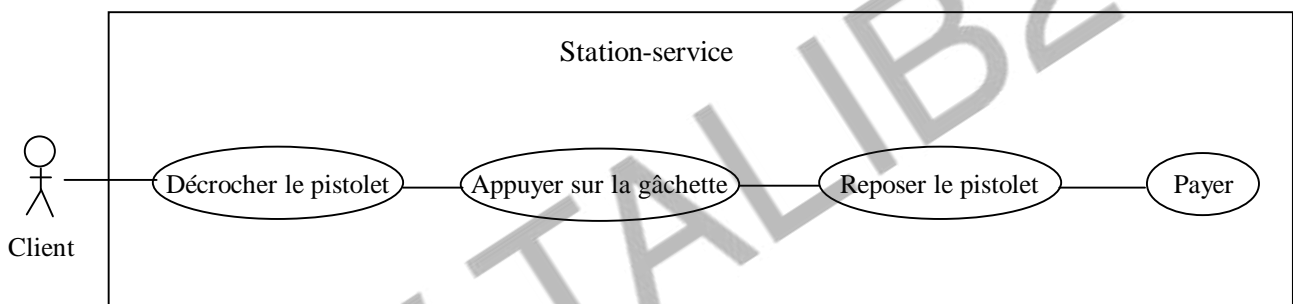
Exercices

1 : la station-service

Soit un système informatique qui gère une station-service :

- Le client peut utiliser des pompes manuelles et payer à la caisse du gérant ou utiliser des pompes automatiques.
- Le gérant de la station utilise le système informatique pour ses opérations de gestion (particulièrement le bilan des opérations de vente d'essence).
- Le gérant peut se servir de l'essence pour sa voiture.
- La station-service a un petit atelier d'entretien de véhicules. Le gérant est aussi mécanicien.

1) Que pensez-vous du diagramme présenté ci-dessous ?



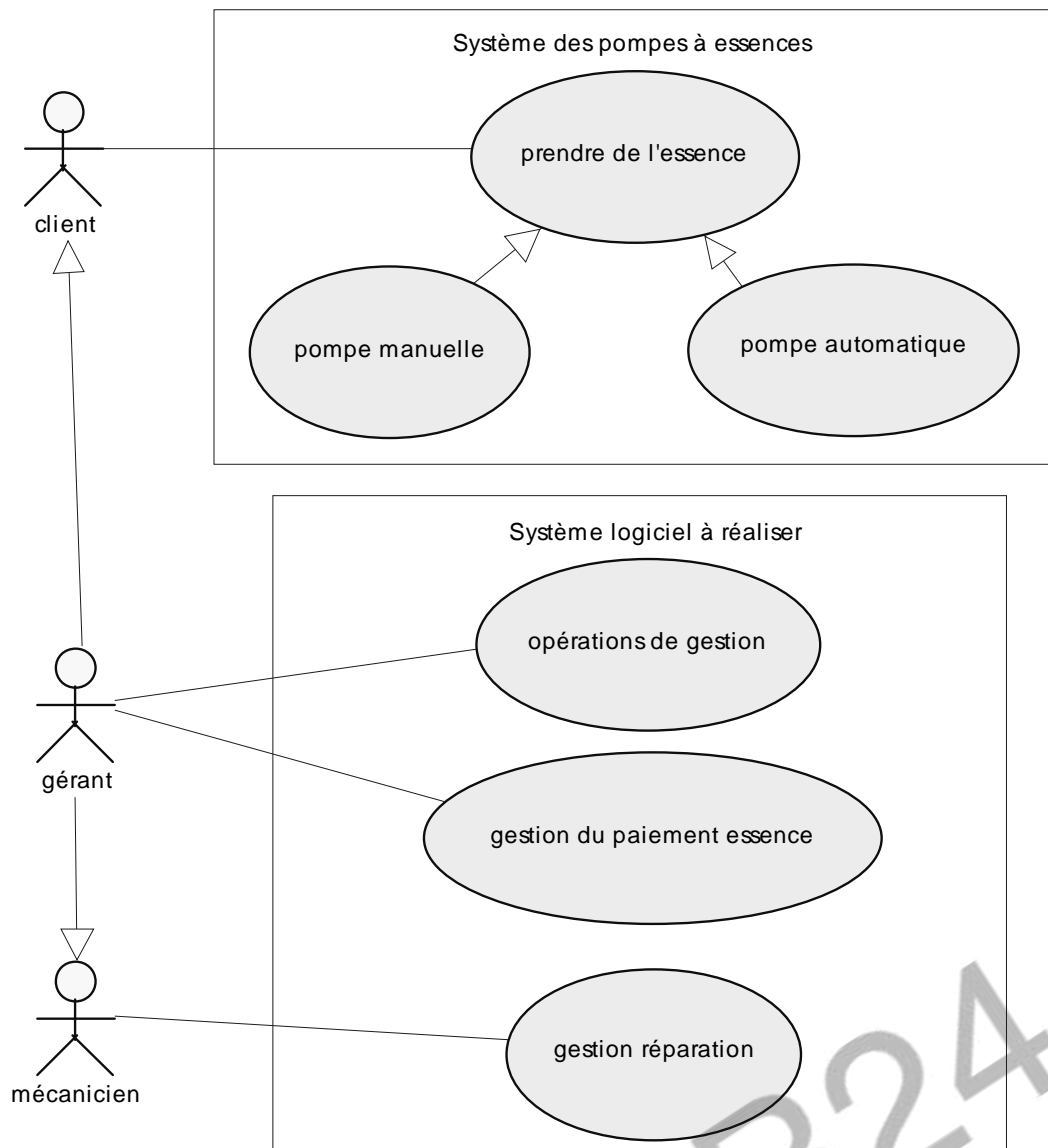
➤ Réponse :

Ce schéma n'est pas un diagramme de cas d'utilisation.

Il correspond à peu près à un diagramme d'activités (il faudrait orienter les liaisons) : il décrit la séquence des actions effectuées par le client. De plus, il fusionne deux usages : prendre de l'essence (ce qui se fait à la pompe) et payer (ce qui se fait à la caisse).

2) Faites le diagramme des cas d'utilisation.

➤ *1^{ère} solution (pas la bonne !)* :



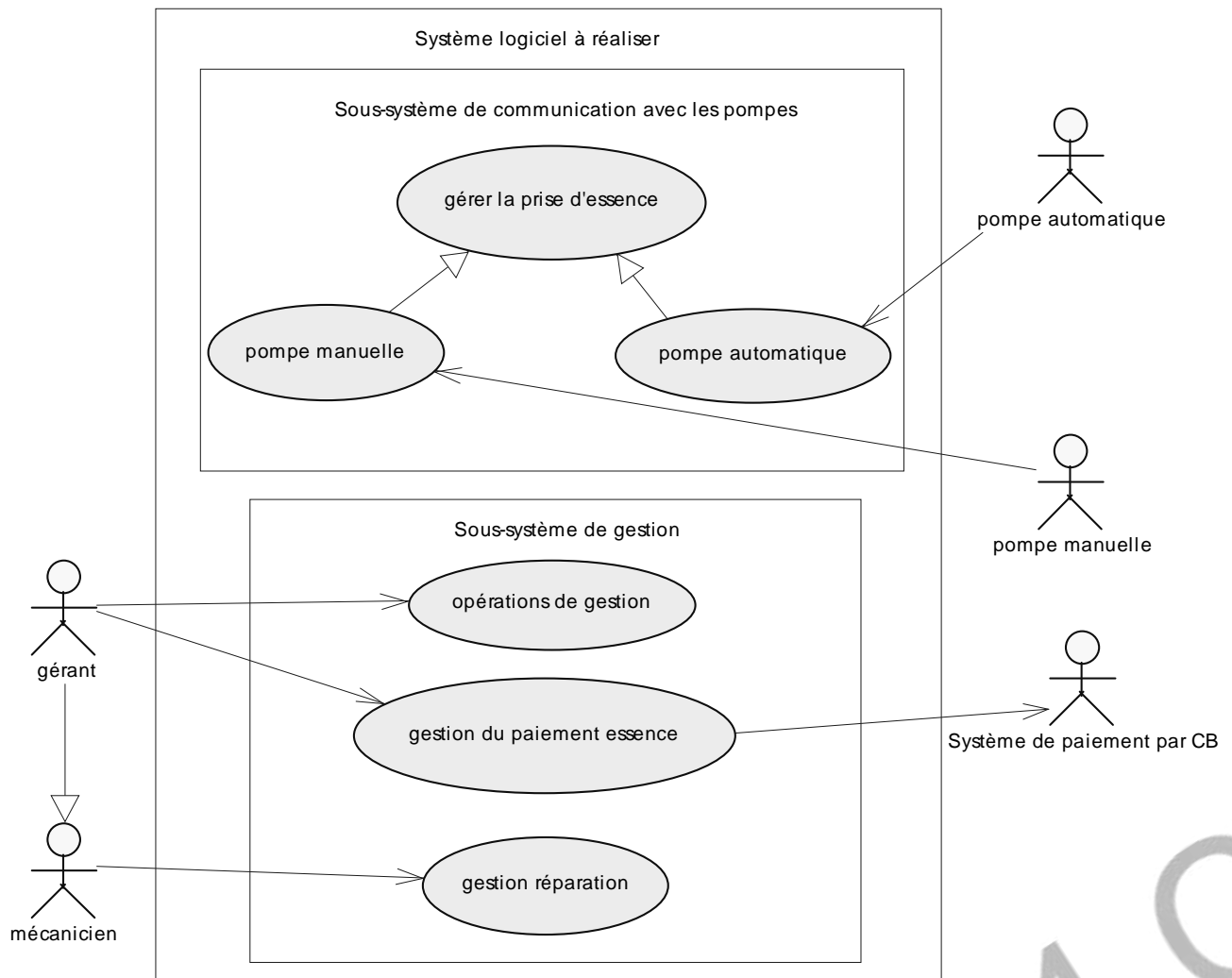
Remarques :

Il y a un héritage entre gérant et mécanicien : cela veut dire que le logiciel qu'utilise le gérant lui permet aussi de faire ce que permet de faire le logiciel qu'utilise le mécanicien. Cet héritage est un choix de conception.

Il n'y a pas d'héritage entre gérant et client : le logiciel qu'utilise le gérant ne permet pas de prendre de l'essence ! Quand le gérant prend de l'essence, il n'est plus gérant, mais un client comme un autre.

On peut mettre au jour deux sous-systèmes : le logiciel et le matériel (la pompe à essence). La pompe à essence ne relève pas du développement à réaliser. **Il faut donc revoir notre diagramme !**

➤ 2^{ème} solution (la bonne !):



Remarques :

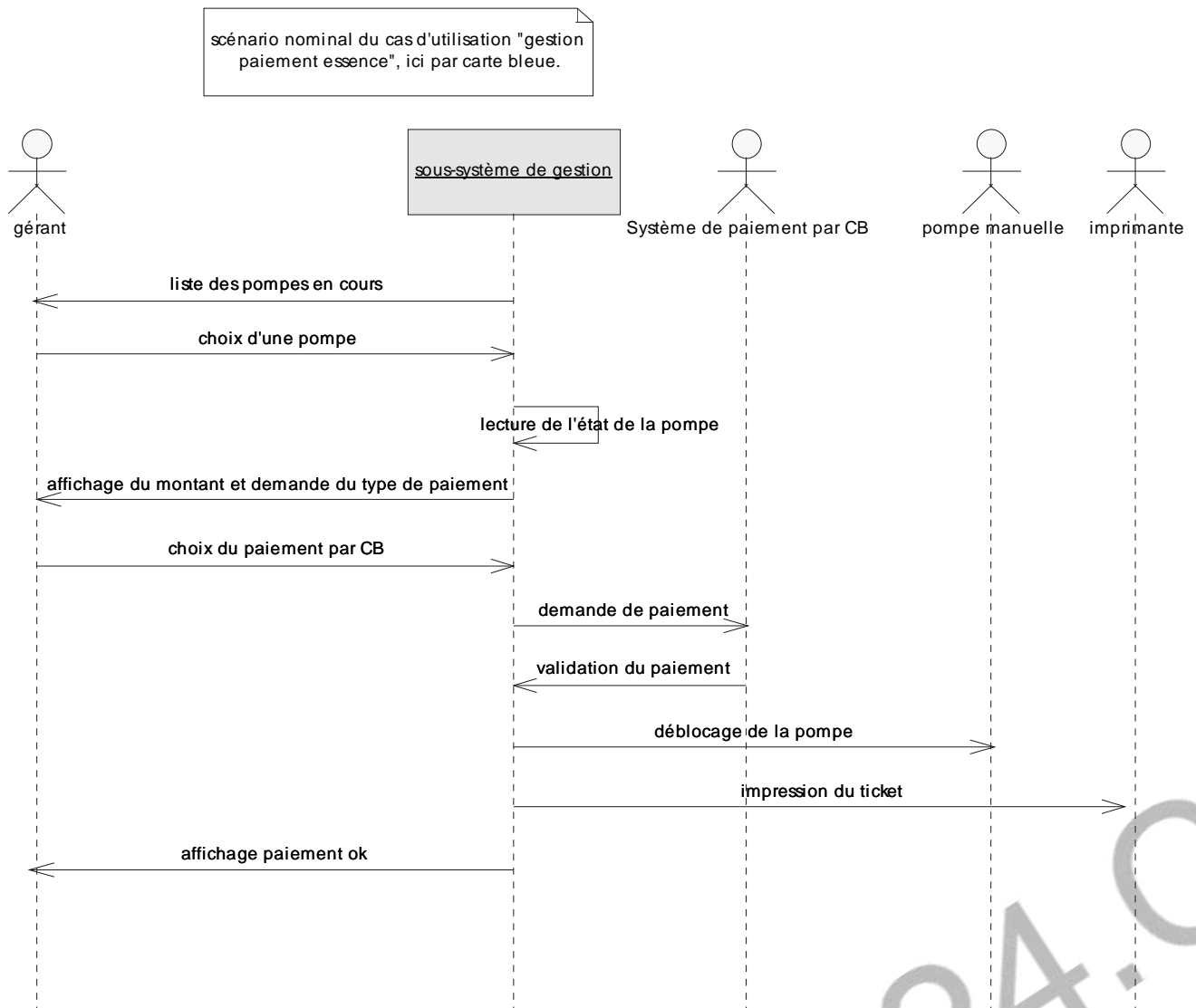
La prise d'essence, manuelle ou automatique, génère pour le logiciel à réaliser un cas d'utilisation qui est le fait de récupérer les informations de cette prise d'essence (quelle pompe, quel carburant, quel prix, quel client et quelle transaction s'il y a lieu, etc.).

Ce sont les deux machines : « pompe automatique » et « pompe manuelle » qui envoient ces informations à notre système.

Ainsi, on a bien décrit les usages du logiciel que nous avons à réaliser : gestion globale, gestion des paiements en caisse, gestion des réparation, gestion de la prise d'essence.

A noter que le diagramme des cas d'utilisation permet aussi de commencer le travail d'architecture organique au sens de la mise au jour des sous-systèmes.

3) Faire des diagrammes de séquence système



Remarques :

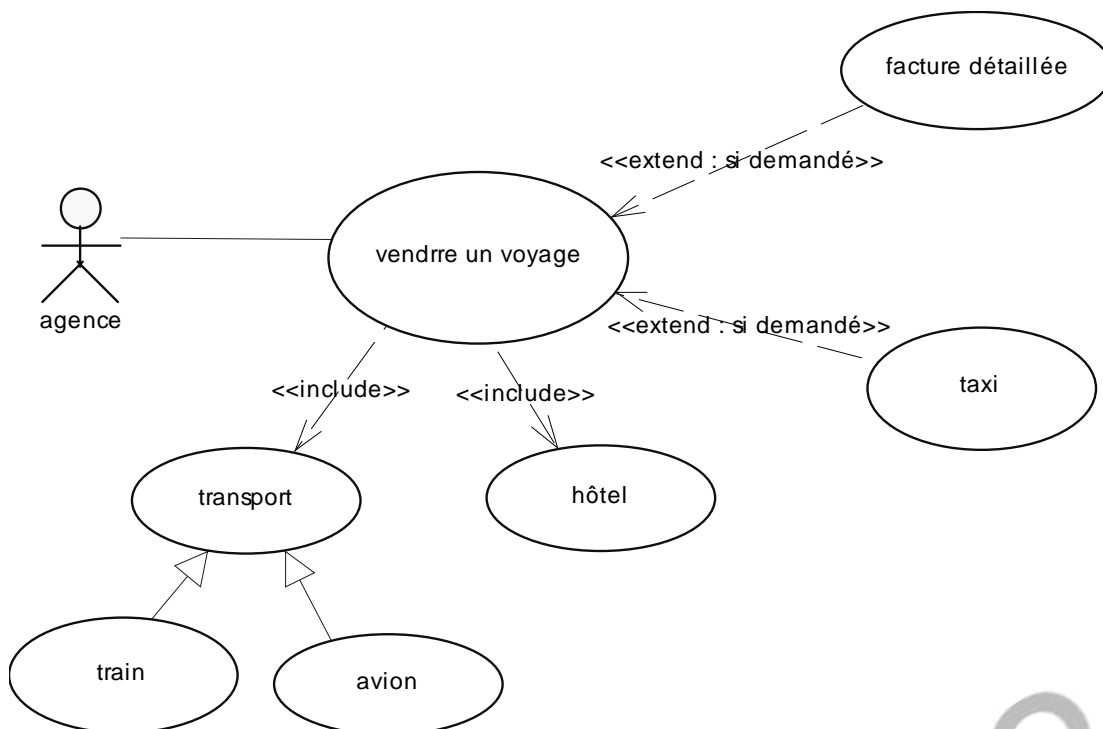
l'appel réflexif sert à montrer la communication avec un autre système ou sous-système. Ici, l'état de la pompe est une information que le système pompe nous aura communiqué.

Déblocage de la pompe : on a considéré que la pompe est bloquée tant que le paiement n'est pas validé (c'est un choix de conception à discuter avec le client !).

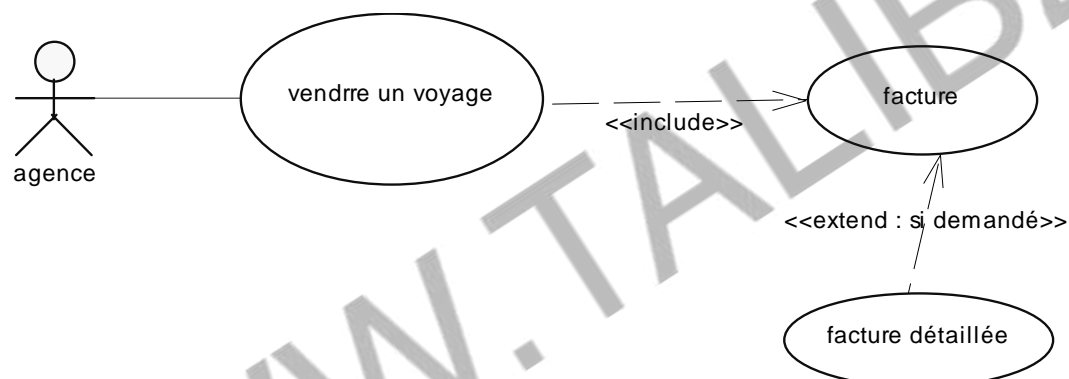
2 : l'agence de voyage

- Une agence de voyage organise des voyages et gère le transport, l'hébergement et offre la possibilité à ses clients de disposer d'un taxi à l'arrivée du voyage pour se rendre à l'hôtel. Qui est acteur du système ? Modéliser les cas d'utilisation.
- Détailler les cas d'utilisation trouvés.
- Certains clients demandent des factures détaillées. Les voyages peuvent se faire soit par train, soit par avion. Compléter le diagramme des cas d'utilisation.
- Faire le diagramme de séquence système de l'activité principale. On utilisera des modules sans les détailler. On détaillera ensuite un module au choix.

Diagramme des cas d'utilisation



➤ Variante 1



➤ Variante 2

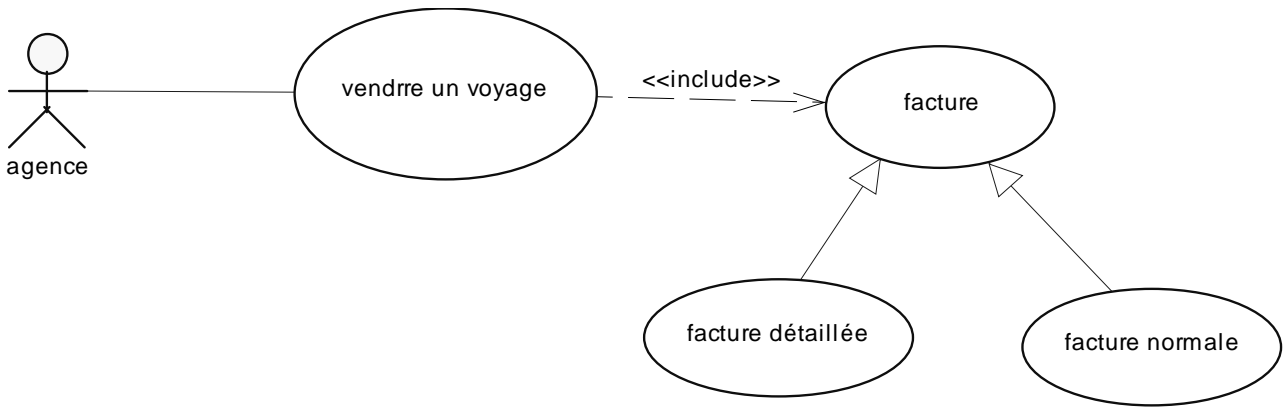
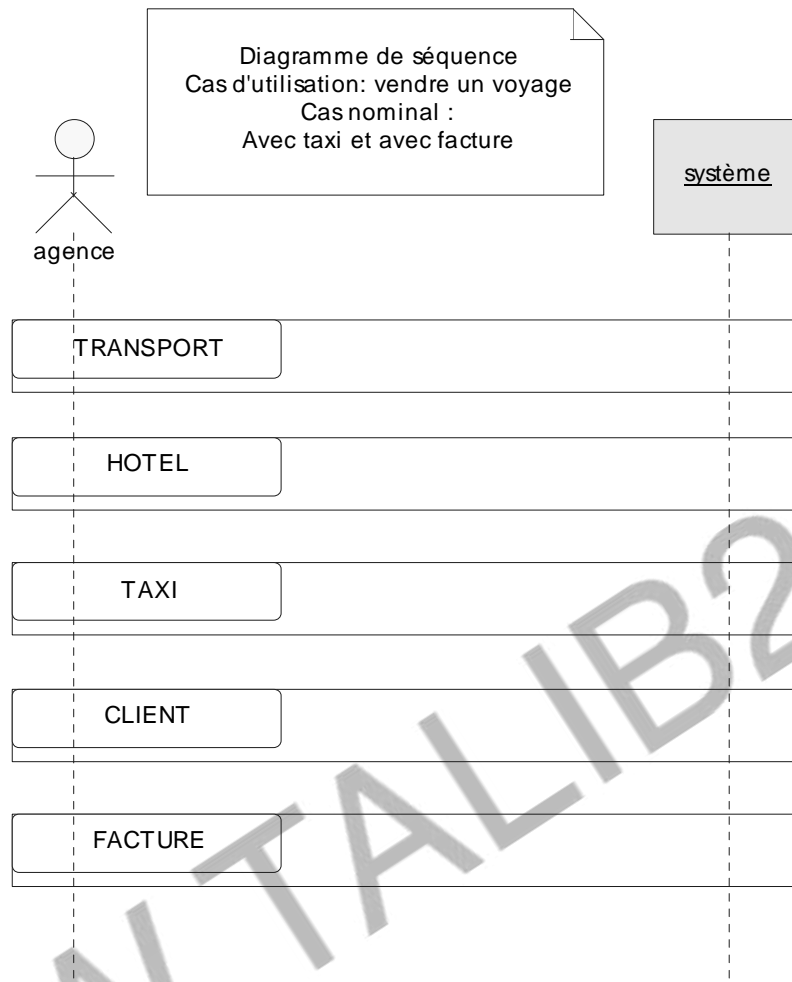


Diagramme de séquence système

➤ Cas d'utilisation « vendre un voyage »



WWW.TALIB24.COM

➤ Cas d'utilisation « transport »

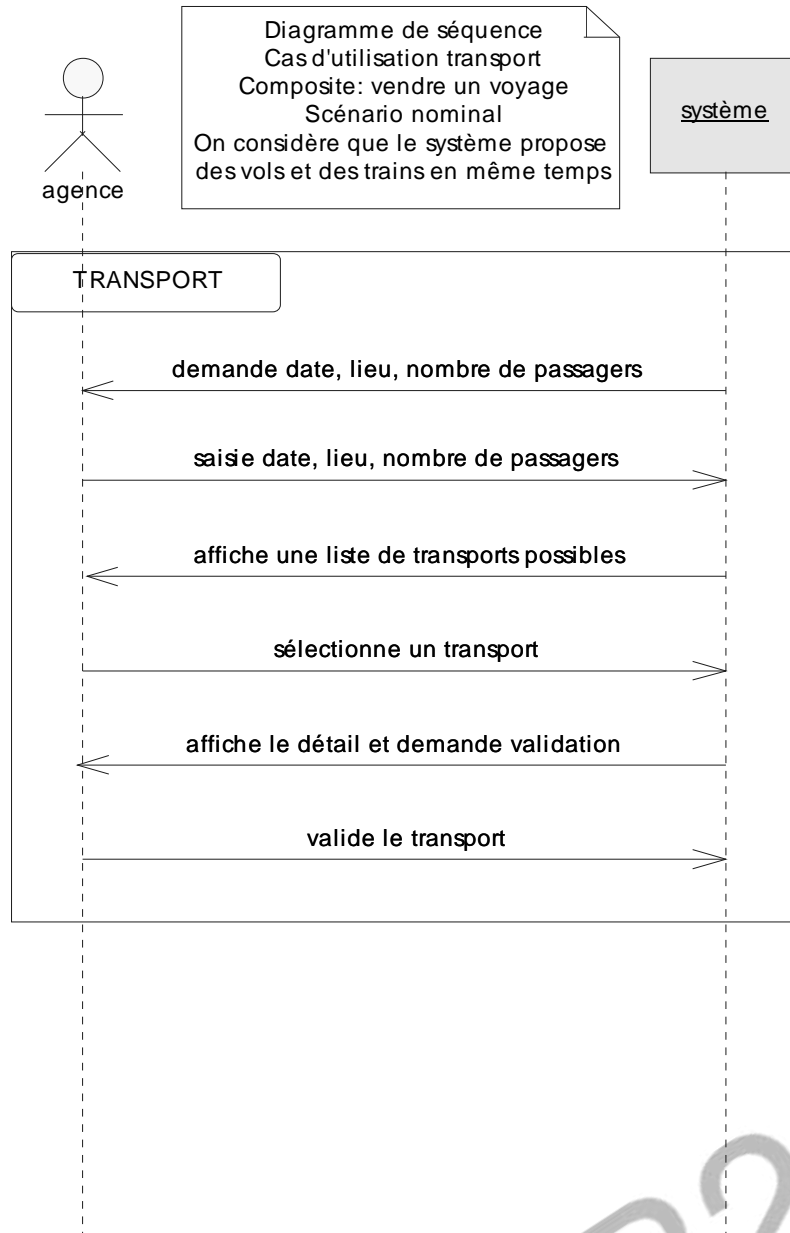
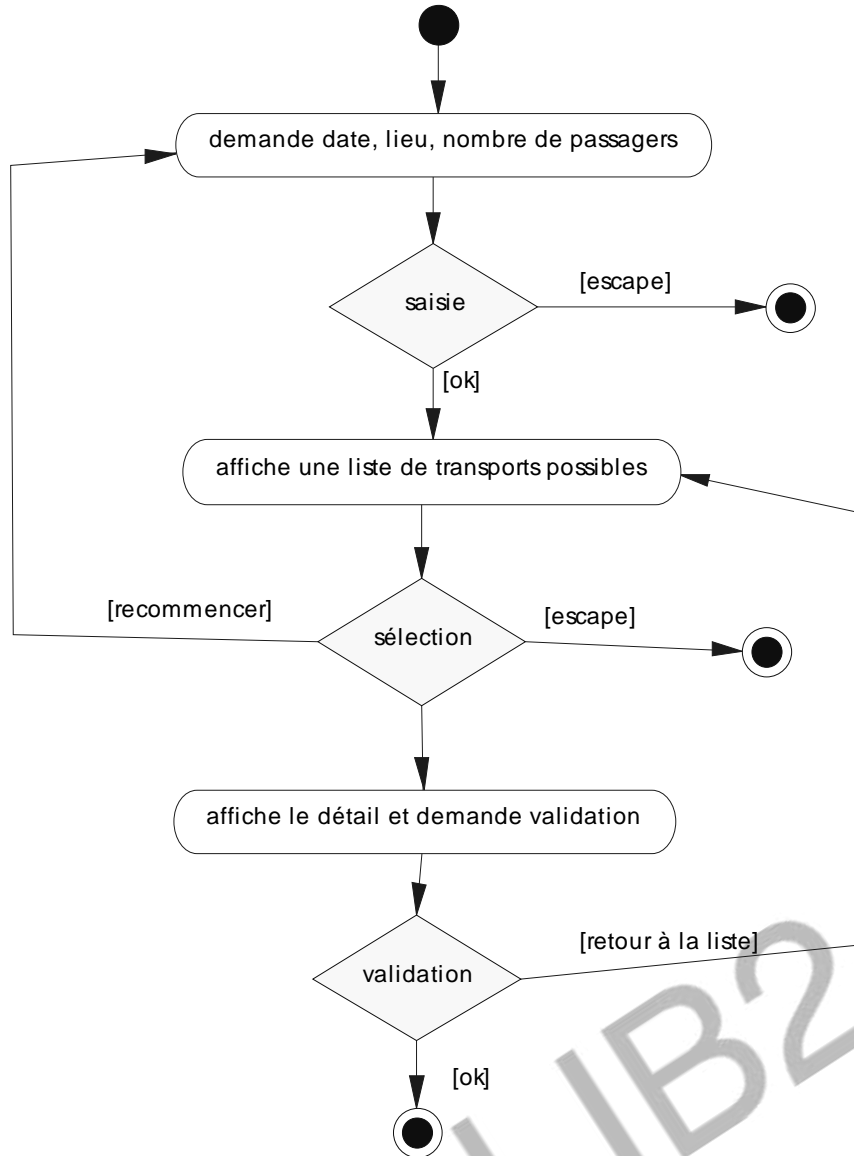


Diagramme d'activités

➤ Cas d'utilisation « transport »

Diagramme d'activités décrivant tous les scénarios du cas d'utilisation « transport »

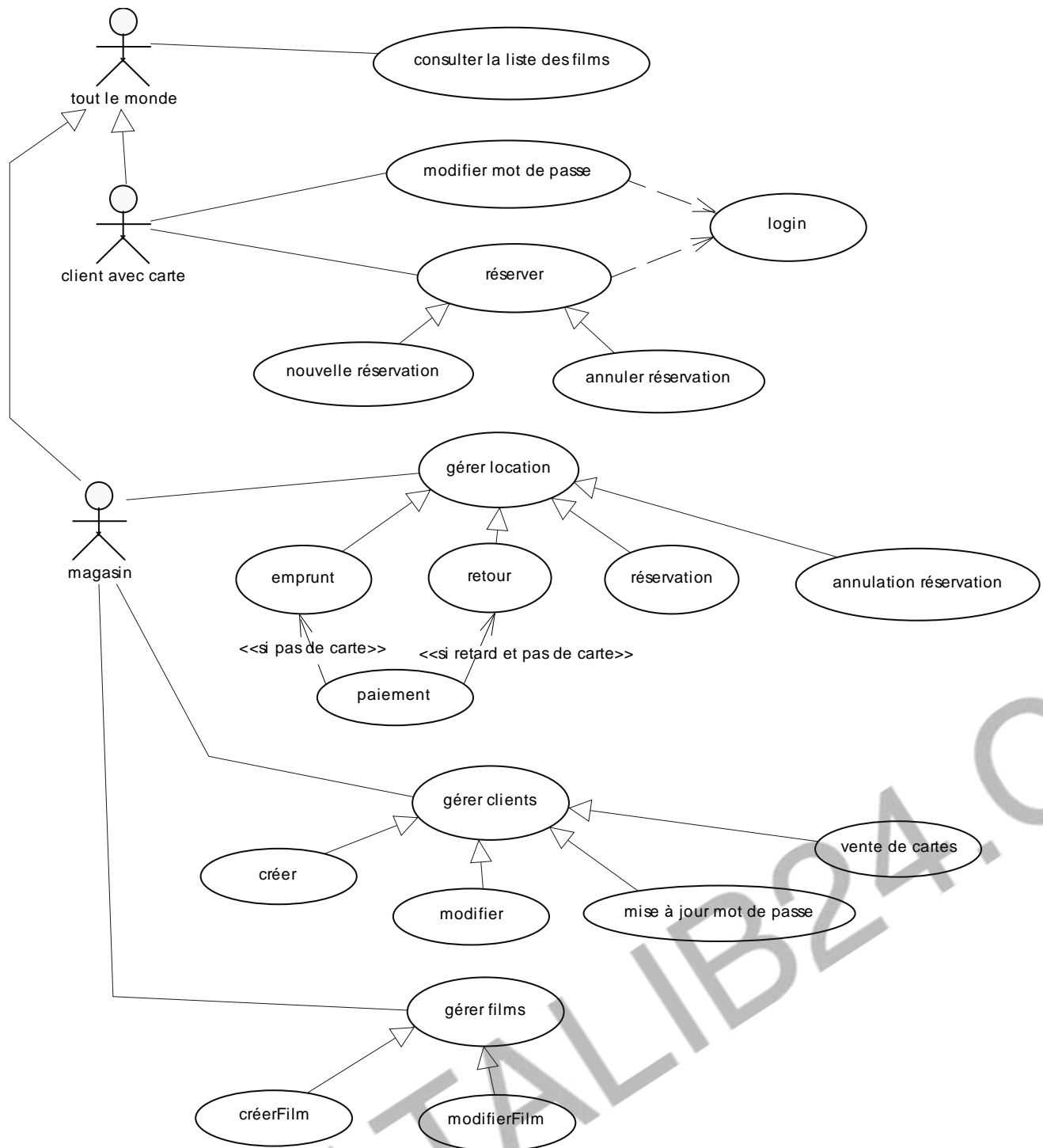


A noter qu'on reprend la séquence système du cas nominal en détaillant les alternatives et les boucles.

La présentation verticale est la plus lisible.

3 : Location de DVD

Voir le sujet dans le poly de TP.



REMARQUE : à ce niveau, la distinction entre la borne et le poste internet n'existe pas. Cette distinction relève de l'architecture : elle distingue entre deux sous-systèmes.

Pour la représenter, il y a plusieurs options selon ce qu'on cherche à montrer. Une solution consiste à faire deux autres diagrammes correspondant au deux sous-système : un pour la borne avec ses cas d'utilisation, un pour l'internet avec ses cas d'utilisation.

TD N°2 Diagramme de classes

On veut implanter une base de données pour la gestion des rencontres des tournois de tennis d'une saison donnée. On pourra ainsi déterminer facilement le classement des joueurs au niveau mondial. On dispose des informations suivantes.

Spécifications

Chaque **joueur** a **un nom** et **un sexe**.

Deux joueurs peuvent former une **équipe**.

Un **tournoi** est identifié par **son nom** et se déroule dans un **pays** donné à **une date prévue**.

La **dotation** des gagnants varie selon les tournois et est exprimée dans la monnaie du pays d'accueil. Afin de plus tard afficher les dotations en utilisant la bonne « unité », on souhaite également stocker le nom de **la monnaie** de chaque pays

À la fin d'un tournoi, un joueur ou une équipe participant à ce tournoi obtient **un score** qui représente le nombre de tours passés dans le tournoi

(1^{er} tour vaut 1 point, 2^{ème} tour vaut 2 points, etc.). On attribue à chaque tournoi **un coefficient** selon son importance.

Le score final d'un joueur (ou d'une équipe) est obtenu de la manière suivante :

$\sum \text{score} \times \text{coefficient}$ pour les n tournois de l'année.

Les joueurs (ou équipes) sont classés par ordre décroissant de leur score final.

Travail demandé

I. En considérant qu'une équipe forme une entité, proposer un modèle conceptuel E/A pour cette base de données.

Recueil des données brutes

Donnée atomique	Entité
NomJ	Joueur
SexeJ	Equipe
NomE	Tournoi
DatePrevue	Pays
Score	
Coeficient	
ScoreFinal	

Données atomiques : Attribut (donnée indivisible)

un attribut appartient à une et une seule entité : classe

NomJ appartient à la classe Joueur

SexeJ appartient à la classe Joueur

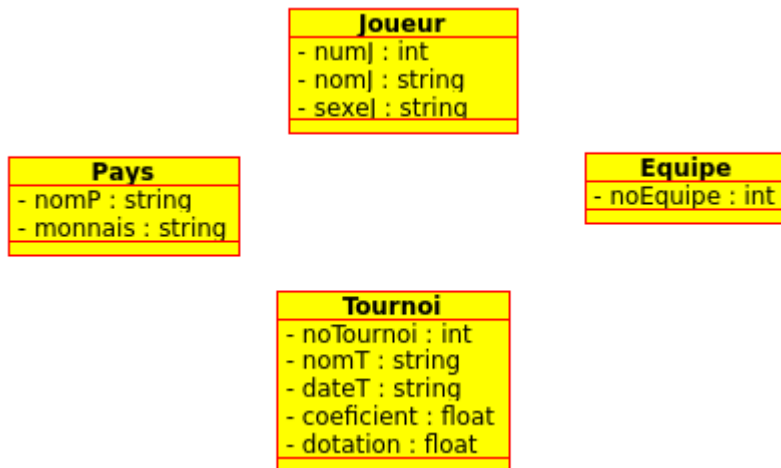
...

L'attribut score ???

ScoreFinal est une donnée atomique qui peut appartenir à la classe joueur ou à la classe Equipe, mais cette donnée est calculée (une classe est composée d'attributs non calculés).

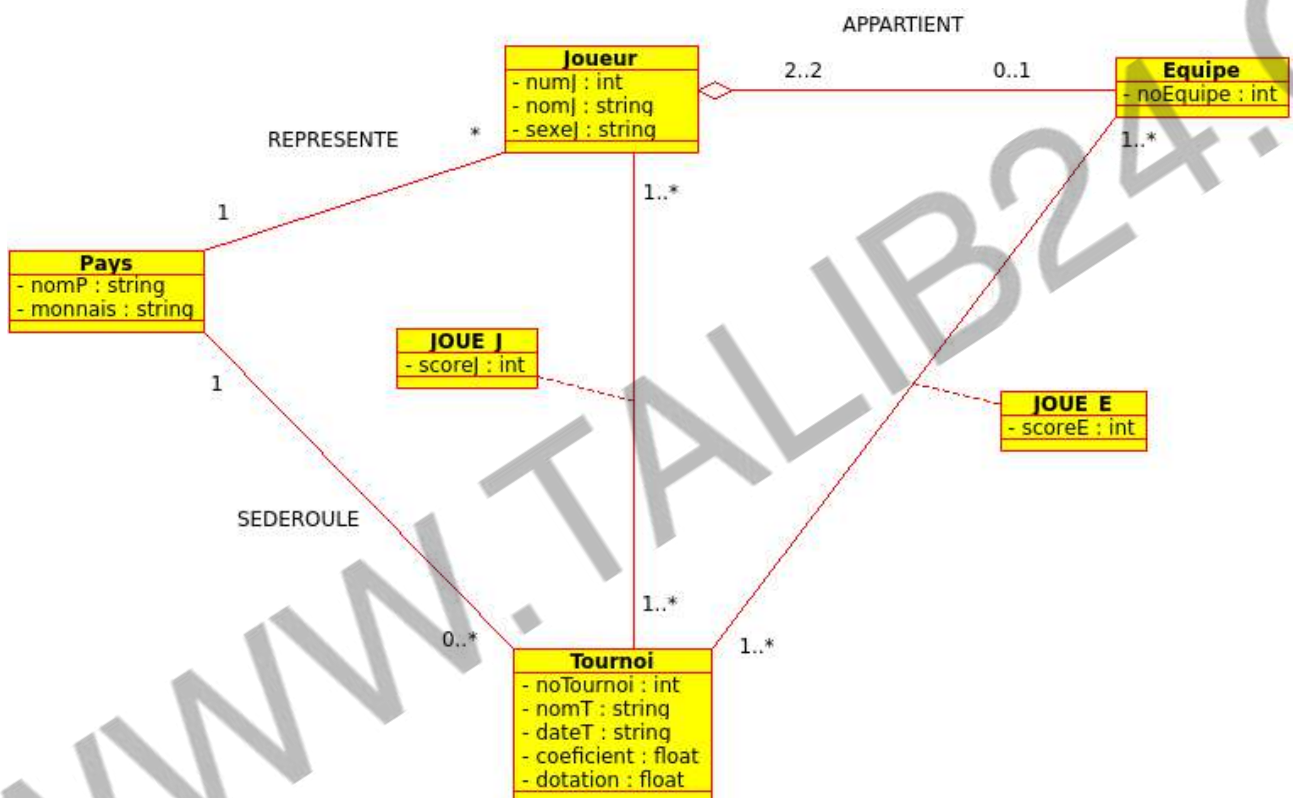
Une classe est une entité est composée de plusieurs attributs .

Une classe est un tout homogène.



Concernant les associations :

Association	Classe n°1	Classe n° 2	Type association
REPRESENTE	Joueur	Pays	Bidirectionnelle
APPARTIENT	Joueur	Equipe	Agrégation : Composition
SEDEROULER	Tournoi	Pays	Bidirectionnelle
JOU_J	Joueur	Tournoi	Classe_Association
JOU_E	Equipe	Tournoi	Classe_Association



Description du processus d'achats de produits et services de Entreprise XYZ

Note : Ce processus d'affaires est dérivé d'un cas réel. Une étude complète, analyse et architecture, a été réalisée par Khalil Mamouny dans le cadre de ses mandats.
Le nom de l'entreprise est fictif.

Description sommaire du besoin d'affaires

Entreprise ^{XYZ} voudrait développer, ou acquérir, une solution logicielle qui lui permettrait de gérer électroniquement l'ensemble de la documentation générée par les différentes activités de son processus d'achat de produits et services.

Description textuelle du processus d'affaires

Le processus d'achat de services et de produits de Entreprise XYZ se décrit comme suit :

- Le processus d'achat commence quand un employé (requérant) complète et transmet une demande d'achat à un des acheteurs de l'entreprise.
- À la réception de la demande d'achat, l'acheteur doit valider la demande. Si la demande d'achat est complète et recevable, l'acheteur, selon le cas :

Scénario A

S'il n'y a pas de fournisseur pré-qualifié, publier un appel d'offres. Dans ce cas, les fournisseurs doivent transmettre leur soumission à l'acheteur. L'acheteur, selon les règles administratives de l'entreprise, sélectionne la soumission la plus avantageuse puis complète un bon de commande qu'il transmet au fournisseur sélectionné.

Scénario B

S'il y a un fournisseur pré-qualifié, avec qui il existe une entente commerciale, compléter un bon de commande et le lui transmettre.

- Si la demande d'achat consiste à réparer un item (par exemple, un meuble ou un équipement), l'acheteur doit, en plus de compléter un bon de commande, expédier l'item au fournisseur et lui transmettre un bon d'expédition. Dans ce cas, c'est le magasinier de l'entreprise qui prépare l'item, complète un bon d'expédition puis envoie le tout au fournisseur.
- Suite à la réception du bon de commande, le fournisseur, selon le cas :

Scénario C

Si l'achat est un service, il réalise le service conformément à ce qui été convenu dans le bon de commande puis transmet sa facture à l'acheteur de l'entreprise.

Scénario D

Si l'achat consiste à réparer un item ou des items, effectue la réparation puis livre les items accompagnés d'un bon de livraison et de la facture.

Scénario E

Si l'achat est un ou des produits, le fournisseur livre le(s) produit(s) accompagné(s) du bon de livraison et de la facture. Si un produit est homologué, il doit également joindre le bon de conformité.

- À la réception de la facture, si le tout est conforme aux spécifications du bon de commande, l'acheteur transmet la facture à l'administration pour paiement.
- Le processus d'achat est complété une fois la facture du fournisseur est payée.

Définir le diagramme de cas d'utilisation et de ses composants conformément aux spécification de la version 2.4.1 de la notation UML

Rappels

Toutes les définitions concernant les concepts et les notations qui seront présentée ici sont basées sur le document original de la version 2.4.1 publié par l'OMG (Object Management Group) que l'on peut télécharger à l'adresse suivante : <http://www.omg.org/spec/UML/2.4.1/superstructure>

Dans la version 2.4.1 de la notation UML, 14 diagrammes ont été standardisés, 7 sont regroupés dans les diagrammes de structure et 7 sont regroupés dans la catégorie des diagrammes de comportement :

Diagrammes de structures :

- Diagramme de classe
- Diagramme d'objets
- Diagramme de composants
- Diagramme de paquetage
- Diagramme de structure composite
- Diagramme de déploiement
- Diagramme de profil

Diagrammes de comportement

- Diagramme de cas d'utilisation
- Diagramme d'états-transitions
- Diagramme d'activités
- Diagramme d'interaction
- Diagramme de communication
- Diagramme d'interaction-Vue d'ensemble
- Diagramme de séquences
- Diagramme de temps

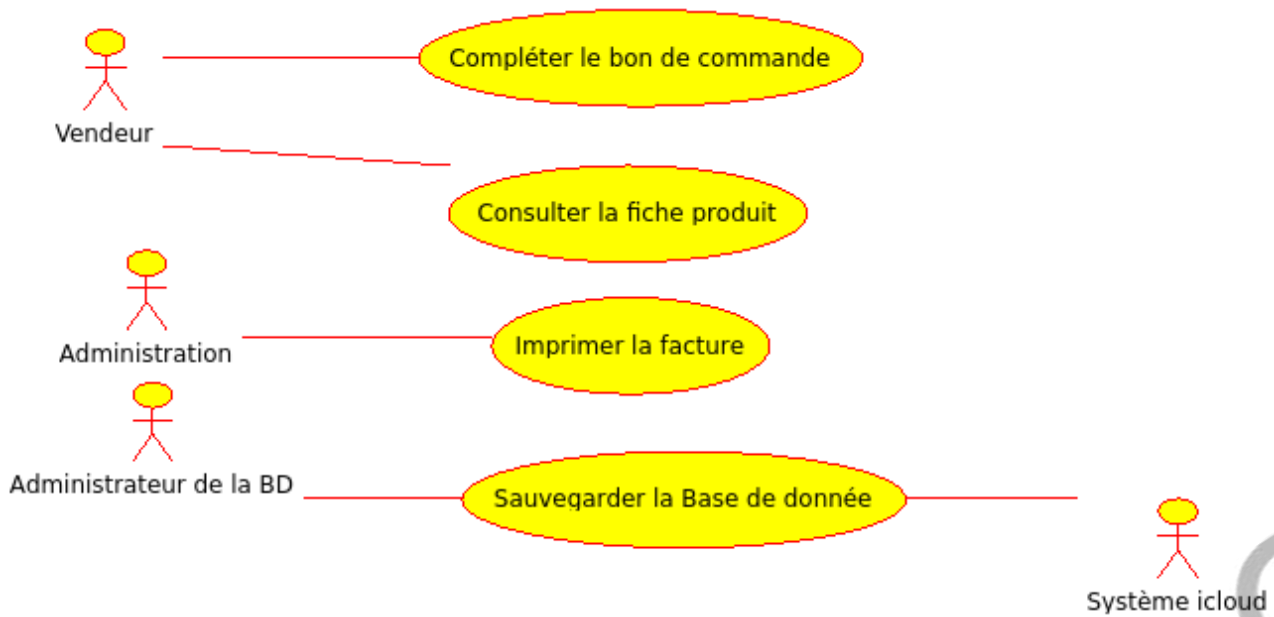
Le diagramme de cas d'utilisation fait partie des diagrammes de comportement.

Les cas d'utilisation servent à spécifier les fonctionnalités d'un système. Ils servent typiquement dans le recueil des besoins d'un système et définissent ce que ce dernier est supposé faire.

Les concepts clés qui sont associés aux cas d'utilisation sont :

- Le sujet,
- L'acteur et
- Le cas d'utilisation.

Systeme de gestion de vente



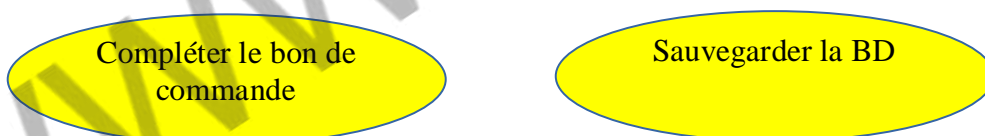
Le sujet est le système qui fait l'objet de l'étude.

Un acteur peut être une personne (vendeur, administrateur de la BD), un groupe de personnes (administration) ou un autre système externe qui interagit avec le sujet :

La représentation symbolique de l'acteur peut être remplacée par une représentation plus significative visuellement :

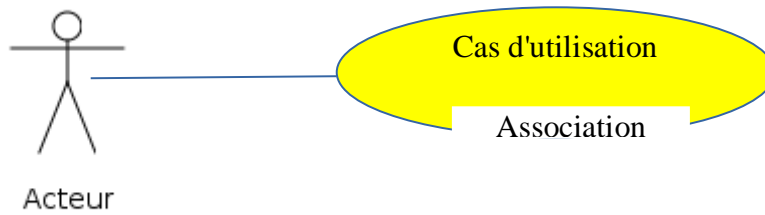


Le cas d'utilisation est représenté par une forme ovale

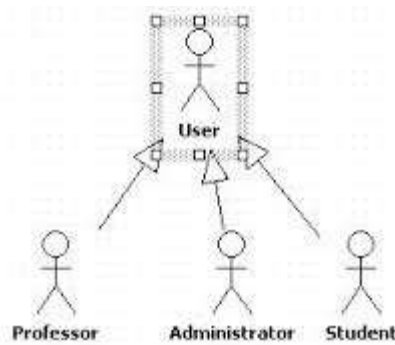


Dans un diagramme de cas d'utilisation, on retrouve différents type de relations entre les éléments.

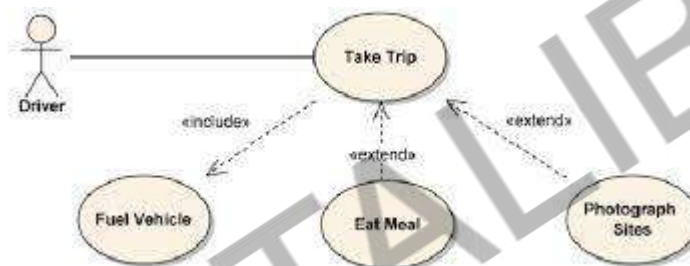
Le type le plus commun entre un acteur et un cas d'utilisation est l' «**Association**»



Le type le plus commun entre deux acteurs est la «**Généralisation**»



Les deux types de relations les plus communs entre 2 cas d'utilisation sont «**Inclusion**» & «**extention**»



Démarche de création du diagramme de cas d'utilisation en 4 étapes:

1. Identifier et ajouter le sujet
2. Identifier et ajouter les acteurs
3. Identifier et ajouter les cas d'utilisation
4. Identifier et ajouter les relations entre les éléments

1. Définir le sujet et l'ajouter au diagrammes

Rappelons que le sujet représente le système qui fait l'objet de l'étude.

Le nom du sujet doit être court et significatif.

Dans notre cas le nom du sujet est identifié dans la description sommaire du besoin d'affaires :

«...une solution logicielle qui lui permettrait de gérer électroniquement l'ensemble de la documentation générée par les différentes activités de son processus d'achat de produits et services»

On peut résumer le sujet par:

«**Système de gestion des dossiers d'achat SGDA**»

2. Identifier les acteurs et les ajouter au diagramme

Rappelons que l'acteur est toutes personne, groupe de personnes, organisation ou système externe qui interagit avec le système.

A partir de la description textuelle du processus d'affaires, nous allons repérer tous les acteurs qui interagissent avec notre sujet c'est-à-dire avec le SGDA .

Dans un premier temps on va identifier les noms qui correspondent à ceux d'un acteur:

Requérant, acheteur, Entreprise, Fournisseur, Magasinier, Administration.

3. Identifier et ajouter les cas d'utilisation au diagramme

Rappelons qu'un cas d'utilisation représente une fonctionnalité du système.

L'identification des cas d'utilisation, nécessite plus d'analyse : il faut interpréter le texte, apporter son jugement et reformuler certaines phrases.

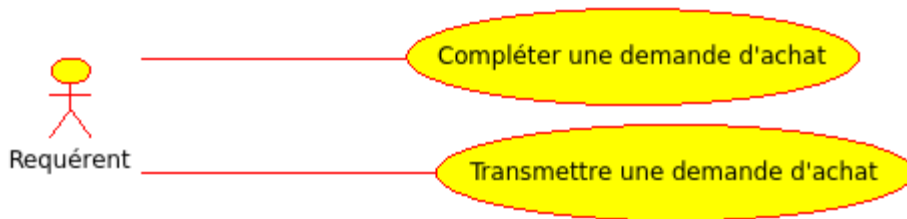
On va procéder en 4 sous étapes:

- 3.1 extraire du texte ce qui s'apparente à des cas d'utilisation et le copier dans un tableau
- 3.2 reformuler le texte pour mettre en évidence les cas d'utilisation
- 3.3 faire des regroupements et simplifier pour éviter de surcharger le diagramme
- 3.4 ajouter les cas d'utilisation au diagramme

3.1 extraire du texte ce qui s'apparente à des cas d'utilisation et le copier dans un tableau

3.2 reformuler le texte pour mettre en évidence les cas d'utilisation

Texte extrait sans modification	Reformulation du texte
un employé (requérant) complète et transmet une demande d'achat à un des acheteurs	Le requérant complète une demande d'achat (crée et enregistre) Le requérant transmet une demande d'achat à l'acheteur (imprime ou envoie par courriel



3.3 faire des regroupements et simplifier pour éviter de surcharger le diagramme



Si on prend un groupe de cas :

- Transmettre une demande d'achat
- Compléter une demande d'achat
- Valider une demande d'achat

On peut puiser dans nos expériences et ajouter d'autres fonctionnalités comme:

- Authentification
- Créer une nouvelle DA
- Mettre à jour une DA

- Imprimer une DA
- Afficher une DA
- Annuler une DA
- Transmettre une DA par courriel
- Supprimer une DA
- Afficher la liste des DA
- Imprimer la liste des DA

On pourrait regrouper tous ces cas d'utilisation dans un seul cas d'utilisation qui serait plus global et qu'on l'appellerait «Gérer les demande d'achat»:

Gérer les demandes
d'achat

Faire la même chose avec les autre regroupements :

Gérer les appels
d'offre

Gérer les bons de
commandes

Gérer les
soumissions

Gérer les ententes
commerciales

Gérer les demandes
d'achat

Gérer les bons
d'expédition

Gérer les factures

Gérer les bons de
livraison

Gérer les certificat de
conformité

Gérer les paiements

3.4 Ajouter les cas d'utilisation au diagramme



Éviter de surcharger le diagramme avec des détails ,

Acteur « Entreprise » est à supprimer car il est synonyme de « Administration »

L'acteur « Fournisseur » est sûrement associé à certains cas d'utilisation mais on va surcharger le diagramme mais on le fera apparaître dans d'autres diagrammes où le détail est pertinent.

4. Les diagrammes de cas d'utilisation détaillé

4,1 Cas « Gérer les demandes achat »

Ce diagramme contient 9 sous cas d'utilisation .
On trouve les relation de type extends et include



4,2 Cas Gérer les bons de commande

Ici le fournisseur réapparaît ici (Il est supprimer du diagramme global pour éviter de surcharger le diagramme)

